INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA UNIDAD CULHUACAN

SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

"CIFRADOR DE BLOQUES, UTILIZANDO LA TRANSFORMACIÓN CAÓTICA UNIDIMENSIONAL DE LA TENT"

T E S I S

PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS DE INGENIERÍA EN MICROELECTRÓNICA

P R E S E N T A :

LIC. CÉSAR ENRIQUE ROJAS LÓPEZ

ASESOR: DR. RUBÉN VÁZQUEZ MEDINA

México, D.F., Mayo de 2011.

SIP-14

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D. F. siendo las 12:00 horas del día 06 del mes de junio del 2011 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de SEPI-ESIME-CULHUACAN para examinar la tesis titulada:

"Cifrador de Bloques, Utilizando la Transformación Caótica Unidimensional de la TENT"

Presentada por el alumno:

Rojas	**López**	**César Enrique**
Apellido paterno	Apellido materno	Nombre(s)

Con registro: B 0 9 1 8 3 9

aspirante de:

MAESTRÍA EN CIENCIAS DE INGENIERÍA EN MICROELECTRÓNICA

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

LA COMISIÓN REVISORA

Director(a) de tesis

Dr. Rubén Vázquez Medina

Dr. Luis Niño de Rivera y Oyarzabal

Dr. Gonzalo Isaac Duchen Sánchez

Dr. Antonio Castañeda Solís

Dr. Rogelio Reyes Reyes

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

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Gonzalo Isaac Duchen Sánchez

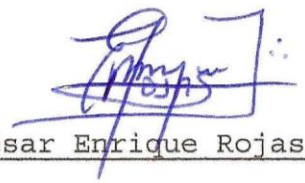
INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México, D.F. el día 06 del mes Junio del año 2011, el (la) que suscribe César Enrique Rojas López alumno (a) del Programa de Maestría en Ciencias de Ingeniería en Microelectrónica con número de registro B091839, adscrito a S.E.P.I. E.S.I.M.E Culhuacan, 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 de Bloques, utilizando la Transformación Caótica Unidimensional de la Tent, 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 crojas@ipn.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.



César Enrique Rojas López

Nombre y firma

DEDICATORIA

A mi hija, Fernanda, ya que es el angelito de mi vida, que me motiva a seguir adelante.

A mi esposa, Grisseld, por su comprensión y apoyo para la culminación de este trabajo. ¡Gracias por creer en mí!

A mis padres, abuelos, hermanos, sobrinos, y toda la familia consanguínea y política, quienes en todo en momento me brindaron su apoyo moral.

AGRADECIMIENTOS

AL GRAN ARQUITECTO DEL UNIVERSO

Porque creo en Él, porque siento dentro de mi mismo la existencia de algo infinitamente superior; porque en todos y cada uno de los seres que pasan ante mis ojos, en cada uno de los fenómenos de la naturaleza, hay algo incomprensible, hay algo infinitamente perfecto que nos pone de manifiesto la existencia de una fuerza creadora, desconocida e indefinible por justa consecuencia, que rige sabiamente todo lo que nos rodea.

A MI ASESOR

Al Dr. Rubén Vázquez Medina, quien a pesar de mi mismo, supo conducirme al término de este trabajo; agradezco infinitamente sus enseñanzas, pero al mismo tiempo, su amistad incondicional que me ha demostrado en el breve tiempo que hemos caminado juntos, ojalá sea largo. Y recuerde doctor, soy un alumno más, no uno menos!

A MIS COMPAÑEROS Y AMIGOS

Gracias por soportarme, sé que hacen un gran esfuerzo, por eso quiero agradecérselos; Al amigo Jorge Alberto M. Ñonthe, por sus enseñanzas; a la amiguita Luz María, por sus explicaciones; al amigo Leo Luengas y Rodrigo Vignettes, por las discusiones sin fin que hemos emprendido; Así mismo a los compañeros del laboratorio de seguridad, que aunque no trabajamos sobre el mismo tema, sí convivimos en el mismo laboratorio, Mario, Azucena, Miguel, Lalo, Memo, Paco, y nuestro compañero de servicio social, Diego Garnica. No quiero dejar de mencionar a mis amigos personales, Rogelio Reyes, Carlos Cortés y Braulio Sánchez, debido a que en algún momento, también me apoyaron de alguna u otra manera para llevar a buen término este trabajo; así como a mi otro amigo, Carlos Aquino Ruiz, quien fue el que me motivó a iniciar esta aventura.

AL IPN

Por permitirme ser parte de su planta docente y darme la oportunidad de ser parte de sus alumnos de posgrado, para posteriormente poder transferir el conocimiento científico y tecnológico a los alumnos de nuestro Instituto, poniendo así, La Técnica al Servicio de la Patria!

A LA COTEPABE

Por otorgarme la licencia con goce de sueldo para estudios de posgrado, eximiéndome temporalmente de mis obligaciones docentes y; sin embargo, percibir los emolumentos correspondientes a mi salario.

ÍNDICE

RESUMEN

Esta tesis muestra el diseño, construcción y evaluación de un algoritmo de cifrado de bloques basado en la función de transformación caótica Tent, como sistema dinámico caótico. Cada bloque y su clave de cifrado son de tamaño de 64 bits de longitud, divididos en 8 subbloques de 8 bits. La función no lineal que se usa en la red desbalanceada de Feistel emplea, como función básica, la función de transformación caótica Tent. Durante el proceso de evaluación de este cifrador, al analizar el diagrama de bifurcación y ver que éste no es totalmente denso una vez que es escalado y discretizado en el universo del ASCII Extendido, se decidió realizar el diseño, construcción y evaluación de otro cifrador de bloques, que mantuviera los 64 bits de longitud en los bloques y en la clave de cifrado, pero ahora divididos en 4 subbloques de 16 bits.

Posteriormente, se calcula la entropía del archivo a la entrada (texto plano) y se compara con la entropía del archivo a la salida (texto cifrado) como medida de difusión en el proceso de cifrado, según se explica en el capítulo IV. Así mismo, se compara el desempeño de los algoritmos propuestos con otros algoritmos de cifrado de bloques que utilizan la transformación caótica logística y senoidal, así como con otros algoritmos que han tenido buena aceptación comercial [1] (AES, DES, TRIPLE DES, BLOWFISH, IDEA Y SKIPJACK), con base en el criterio de seguridad de Claude E. Shannon, calculando la Información Mutua del criptosistema y la Distribución Estadística del mismo empleando las pruebas estadísticas de aleatoriedad del NIST [2].

Finalmente, se dan a conocer los resultados obtenidos de los algoritmos propuestos, donde se puede apreciar que la transformación caótica Tent, ofrece mejores condiciones de comportamiento en un criptosistema caótico de bloques, que cuando se usan otras transformaciones caóticas, como la logística y la senoidal, incluso cuando se utilizan criptosistemas comerciales, según lo reportado en los resultados de la entropía.

ABSTRACT

This thesis shows the design, construction and evaluation of a ciphering block algorithm based on a Tent chaotic transformation function as a dynamic chaotic system. Block and encryption key are 64 bits size, with sub-blocks of 8 bits. The non-linear function used in the non-balanced Feistel network has as its basic function the Tent chaotic transformation. During the evaluation process of this cipher, when analyzing the bifurcation diagram it can be observed that this is not completely dense once it is scaled and discretized in the universe of Extended ASCII, it was decided to design, construct and evaluate another block cipher to keep the 64 bits length in the blocks and the encryption key, but now divided into 4 sub-blocks of 16 bits.

Afterwards, the entropy of the input file (clear text) is calculated and compared with the entropy of the output file (ciphered text) as a diffusion measure during the ciphering process, according to Chapter IV. It also compares the performance of the proposed algorithm with other ciphering block algorithms using chaotic and sine logistic transformation, as well as other algorithms that have good market acceptance [1] (AES, DES, TRIPLE DES, BLOWFISH, IDEA Y SKIPJACK), based on the security criteria proposed by Claude E. Shannon through the estimation of Mutual Information and the Statistical Distribution of the cryptosystem using NIST randomness statistic tests [2].

Finally, the results of the proposed algorithms are shown, in these results it can be seen that the Tent chaotic transformation offers better behavior conditions in a block cryptosystem that when using other chaotic transformations, such as logistics and sine, even when using commercial cryptosystems, as reported on the results of entropy.

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 de transformación caótica Tent y cuya estructura general está definida por una red de Feistel desbalanceada, es decir, se presenta la realización y evaluación de un *cifrador caótico de bloques*.

Este algoritmo de cifrado se ha evaluado empleando conceptos de la Teoría de la Información como son, la entropía del mensaje de entrada, así como la del mensaje 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.

Esta tesis se conforma de 4 capítulos. En el primer capítulo se presenta el estado del arte de la aplicación de las transformaciones caóticas a los procesos de protección de la información. Por ello, se presenta una revisión general de las principales aportaciones de aquellos trabajos científicos más sobresalientes reportados en los últimos años en revistas internacionales especializadas. Esta revisión se hace con la finalidad de determinar la importancia de las transformaciones caóticas unidimensionales en el diseño de cifradores de bloque caóticos. En el segundo capítulo se describen dos grandes temas; en el primer tema se describen los conceptos de la criptografía; y en el segundo tema, los de la teoría de la información. Estos conceptos, que abarcan desde los tipos de cifradores, hasta las herramientas de evaluación como son la Entropía y la Información Mutua, se utilizan para el diseño y desarrollo de un cifrador de bloques caótico. En

[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 pseudo aleatorios.

el tercer capítulo se describen los conceptos de la Transformación Caótica Unidimensional en general, y de la Tent en particular. Además se muestran los Diagramas de Trayectorias y de Bifurcación, así como la Distribución estadística y el Exponente de Lyapunov de Transformación Caótica Unidimensional de la Tent. Al final de este capítulo se realiza el escalamiento y discretización de la función al dominio ASCII Extendido. En el cuarto capítulo se describe el algoritmo de cifrado propuesto por Kocarev, así como también el algoritmo propuesto en esta tesis en sus dos versiones. Al final de este capítulo se dan a conocer los resultados de esta implementación comparándolos con los obtenidos con las transformaciones caóticas logística y senoidal; y otros cifradores comerciales.

DEFINICIÓN DEL PROBLEMA

Las técnicas criptográficas actuales 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. 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 [3]. 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][4].

La Teoría del Caos ofrece múltiples opciones que permiten crear criptosistemas. Una de ellas, como lo demuestra Kocarev, es el uso de la transformación caótica logística [5]. Sin embargo, esta no es una única transformación caótica unidimensional que se puede usar. Existen otras con mejores condiciones de comportamiento, según lo muestran las herramientas de la mecánica estadística, tal como el diagrama de bifurcación y el Exponente de Lyapunov obtenida de cada transformación caótica unidimensional.

Una transformación caótica unidimensional que puede ser usada como función generadora de ruido en un criptosistema de bloques es la transformación de la Tent, la cual no presenta islas de estabilidad, y es lineal en trazos. Con base en lo anterior, y debido a que la Teoría del Caos ofrece diversas transformaciones caóticas unidimensionales, se aborda el problema de determinar si la transformación caótica unidimensional de la Tent ofrece mejores condiciones de comportamiento en un criptosistema caótico de bloques, que cuando se usan otras transformaciones, como la logística, y ofrece condiciones comparables con criptosistemas comerciales actuales, considerando criterios de la Teoría de la Información y usando las herramientas de la mecánica estadística.

[2] Definición del Caos dada por Robert L. Devaney: 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

HIPÓTESIS

Al desarrollar un cifrador de bloques que utilice la transformación caótica unidimensional de la Tent, se espera obtener mejor comportamiento del criptosistema que cuando se usan las transformaciones caóticas logística y senoidal, ya que según lo muestran las herramientas de la mecánica estadística, como el diagrama de bifurcación y el Exponente de Lyapunov de cada transformación caótica unidimensional, la transformación caótica de la Tent no presenta las islas de estabilidad que existen en las transformaciones caóticas logística y senoidal. Se espera que con esta transformación caótica se evite cualquier indicio de periodicidad en el generador de ruido en la estructura de Kocarev del cifrador caótico de bloques, ayudando así a obtener un mejor criptosistema ya que el nivel de incertidumbre en los archivos cifrados se esperaría que fuera máximo, es decir que la entropía sea muy próxima a la entropía máxima.

OBJETIVO GENERAL

Usando la estructura de Kocarev y las redes de Feistel, desarrollar un cifrador de bloques que utilice la transformación caótica unidimensional de la Tent*,* compararlo con el cifrador que usa la transformación caótica logística o la transformación caótica senoidal, y con otros cifradores de bloques de uso comercial (AES, DES, TRIPLE DES, BLOWFISH, IDEA Y SKIPJACK), utilizando conceptos como entropía, información mutua y las pruebas estadísticas de aleatoriedad del NIST.

OBJETIVOS ESPECÍFICOS

- Investigar y definir en qué consiste la transformación caótica unidimensional de la Tent.
- Caracterizar la transformación caótica de la *Tent,* usando el diagrama de trayectorias, el diagrama de bifurcación, la distribución estadística y el exponente de Lyapunov.
- Escalar y discretizar la función de la *Tent,* con la finalidad de usar el cifrador con archivos escritos en el dominio del alfabeto ASCII Extendido, ya que se encuentra originalmente definido en el intervalo (0,1) en los números reales, y se requiere en el intervalo (0, 255) en los números enteros.
- Determinar la manera en que la entropía y la información mutua, que siendo conceptos de la Teoría de la Información, se pueden utilizar como criterios de comparación en el comportamiento de cifradores de bloques.
- Determinar la manera en que se pueden utilizar las pruebas estadísticas de aleatoriedad del NIST como criterio de comparación de cifradores de bloque.
- Comparar el cifrador diseñado con la transformación caótica unidimensional de la *Tent*, con los desarrollados utilizando la transformación caótica logística y la transformación caótica senoidal.
- Comparar el cifrador diseñado con la transformación caótica unidimensional de la Tent, con los cifradores comerciales (AES, DES, TRIPLE DES, BLOWFISH, IDEA Y SKIPJACK).

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 con ello garantizar que la información se mantenga íntegra y confiable.

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.

CAPÍTULO 1

MARCO DE REFERENCIA

RESUMEN

En este capítulo se presenta el Estado del Arte de la aplicación de las transformaciones caóticas a los procesos de protección de la información, brindando una revisión general de las principales aportaciones de aquellos trabajos de investigación más sobresalientes a nivel internacional, reportados en los últimos años. Con esta revisión se conoce el estado actual de las implementaciones de las transformaciones caóticas unidimensionales en el diseño de cifradores de bloque caóticos.

1.1 Estado del Arte

Introducción

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. Esta teoría ha encontrando bastante aceptación en al área de la criptografía; particularmente, en los procesos de cifrado por bloques [6], [7]. Un campo importante de la Teoría del Caos es el que tiene que ver con las transformaciones caóticas unidimensionales como la transformación de Bernoulli [8], la transformación Logística [9] y la transformación Tent [10].

Estas transformaciones se pueden describir como sistemas dinámicos caóticos, los cuales cumplen con las tres condiciones presentadas por Devaney, donde demuestra que

Un sistema dinámico f es caótico si [4]:

- Los puntos periódicos para f son densos.
- *f es transitiva.*
- f tiene dependencia sensitiva de las condiciones iniciales

Se han considerado principalmente 4 revistas internacionales especializadas, que tienen un gran impacto en el ámbito científico y tecnológico, en ediciones que van del año 2000 al año 2009. Entre las revistas más importantes que se consultaron se tienen las siguientes:

- Electronic and Telecommunications Research Institute Journal
- Chaos, Solitons & Fractals, Applications in science and engineering.
- Information Sciences, Informatics and Computer Science Intelligent Systems Applications.
- Journal of Cryptology.

Los 38 temas encontrados en los trabajos que se reportan se encuentran concentrados en las siguientes áreas:

- Cifradores de bloques
- Cifradores caóticos
- Transformaciones caóticas

La relevancia de cada trabajo se considera de acuerdo con la utilidad que tenga en relación con la creación de algoritmos criptográficos para el cifrado de bloques.

A continuación se presentan los trabajos más relevantes encontrados.

Cifradores de bloques

Erez Petrank y Charles Rackoff propusieron en [11], en el año 2000, un cifrador de bloques encadenado usando códigos MAC, el cual ha sido denominado como CBC MAC debido a que utiliza códigos de autentificación de mensajes (MAC: Message Authentication Code). Los autores enfatizan que los códigos MAC son un método de autentificación ampliamente utilizado, pero el uso de CBC MAC no es seguro cuando se emplean mensajes de longitud variable. Ellos mencionan que existen, entre otras, las siguientes reglas de oro para el uso correcto de este cifrador:

- Dividir el texto en bloques
- Realizar un XOR con el bloque cifrado anterior antes de cifrarlo con la clave secreta
- Utilizar solo el último bloque como MAC

En [12] se hace referencia a que la primera demostración rigurosa de la seguridad de CBC MAC, cuando se utiliza en los mensajes de longitud fija, se dio recientemente por Bellare y otros. Ellos también sugirieron variantes de CBC MAC que manejan mensajes de longitud variable pero en estas variantes existe el inconveniente de que la longitud del mensaje debe ser conocida de antemano (es decir, antes de que el mensaje sea procesado). En este trabajo también se presenta un estudio sobre la autenticación CBC para aplicaciones en tiempo real en las que la longitud del mensaje no es conocida hasta que el mensaje termina. Las variantes consideradas de CBC son las siguientes: en primer lugar consideran una variante de CBC MAC, que llaman el encriptado CBC MAC (EMAC), que controla los mensajes de longitudes desconocidas y variables. EMAC en un mensaje es prácticamente tan simple y tan eficiente como lo es el estándar CBC MAC. Los autores proporcionan una prueba rigurosa de que su seguridad está implícita en la seguridad del cifrado de bloques subyacente. A continuación, sostienen que el CBC MAC básico es seguro cuando se aplica a un espacio de mensaje libre de prefijos. Un espacio de mensaje puede hacerse libre de prefijos mediante la autentificación del último carácter (normalmente oculto) que marque el fin del mensaje.

Otro trabajo en esta dirección es el que presentan Thomas Jakobsen y Lars R. Knudsen en el año 2001 [13]. En este artículo los autores presentan un ataque a cifrados de bloque, usando la técnica de ataque de interpolación. Este ataque consiste en: si el texto cifrado puede ser representado como un polinomio o una expresión rotacional (con *n* coeficientes) del texto plano, entonces el polinomio o dicha expresión rotacional puede ser reconstruida usando *n* pares de texto cifrado/plano. Sin embargo, este tipo de análisis es solamente útil contra funciones algebraicas sencillas o contra algoritmos con pocas rondas. Este método es útil para atacar cifradores que usen funciones algebraicas simples, en particular las funciones cuadráticas, que son de pocas rondas y con una expansión de clave más pobre como las cajas de sustitución en cifradores de bloque, comúnmente llamadas S-cajas. Asimismo, se introducen los ataques basados en las diferencias de orden superior. Estos son casos especiales e importantes de ataques de interpolación. Los ataques se aplican a varios cifradores de bloques, el prototipo de cifrado de 6 rondas de Nyberg y Knudsen [14], que es probadamente seguro contra el criptoanálisis diferencial ordinario, una versión modificada del cifrador de bloque SHARK, y un cifrador de bloque sugerido por Kiefer.

Un artículo más, es el publicado en [15] por Lars R. Knudsen, en el año 2002, en el cual se estudia la seguridad de las redes de Feistel, donde las funciones de ronda son escogidas al azar de una familia de 2^k funciones, escogidas aleatoriamente para cualquier *k*. También toma en cuenta que son redes donde las rondas incluyen funciones que son permutaciones. Knudsen ataca las construcciones bajo el supuesto de que un ataque de recuperación de clave en una misma función de ronda requiere una búsqueda exhaustiva sobre todas las 2^k posibles funciones. Los ataques se dan en todas las construcciones de Feistel, tres, cuatro, cinco y seis rondas, obteniendo límites interesantes en sus niveles de seguridad. En un panorama de texto elegido, los ataques de recuperación de claves en las construcciones de cuatro rondas, en analogía a las permutaciones super-pseudo-aleatorias en el modelo Luby y Rackoff, más o menos toma sólo el tiempo de una búsqueda exhaustiva de la clave de una ronda. Un resultado secundario de los ataques que presenta Knudsen, es que algunas construcciones, que han sido demostrados ser super pseudo-aleatorias en el modelo de Luby y Rackoff, no parecen ofrecer más seguridad en el modelo que él propone, que las construcciones que no son super pseudo-aleatorias.

Ju-Sung Kang, et. al., examinan en el año 2003, la pseudo-aleatoriedad del cifrador de bloques KASUMI [16], que será utilizado en la siguiente generación de teléfonos celulares. A principios de

septiembre del año 2009, en la conferencia de LTE ASIA, celebrada en Hong Kong, se demostró el éxito del estándar 3GPP's LTE (3rd Generation Partnership Project Long Term Evolution) por la atención y el optimismo puesta en ella. Más de 200 delegados examinaron estrategias y los plazos para la migración a las redes LTE. En primer lugar, demuestran que la cuarta ronda de la transformación MISTY-type desbalanceada es pseudo-aleatoria a fin de ilustrar la pseudo-aleatoriedad de la ronda interior de la función *FI* de KASUMI bajo un modelo adaptable destacado. En segundo lugar, los autores muestran que la ronda tres de KASUMI como estructura no es pseudo-aleatoria, pero la cuarta ronda de KASUMI como estructura es pseudo-aleatoria bajo un modelo no adaptable destacado. Aquí el término pseudo-aleatorio quiere decir que la distribución estadística de la secuencia cifrante en la ronda en cuestión, posee una distribución estadística cercana a la distribución estadística uniforme (tiene la apariencia de una señal de ruido).

En el año de 2003, Serge Vaudenay propone en [17] herramientas convenientes para estudiar la pseudo-aleatoriedad, que es un modelo clásico para la seguridad de sistemas de cifrado de bloque, en relación con la Teoría de Shannon, el paradigma de funciones universales de hash Carter–Wegman y el enfoque de Luby–Rackoff. Vaudenay dice que esto permite la construcción de nuevos algoritmos de cifrado con pruebas de seguridad bajo modelos específicos. Así mismo, el autor muestra cómo garantizar la seguridad básica del criptoanálisis diferencial y lineal y ataques incluso más generales. También propone planes de construcción práctica, tales como: COCONUT: A Perfect Decorrelation Design; PEANUT: A Partial Decorrelation Design; y WALNUT: An Alternate Design.

En el año de 2005, John Black y Phillip Rogaway sugirieron en [18] algunas variantes simples del CBC MAC que permitan la eficiente autenticación de mensajes de cualquier longitud arbitraria. Sus construcciones utilizan 3 claves, *K1, K2* y *K3*, para evitar relleno innecesario y aplicaciones MAC en cualquier mensaje $M \in \{0, 1\}$ utilizando un máximo $\{1, [|M|/n]\}$ del cifrado de bloques de n-bits subyacente. La construcción favorita de los autores, XCBC, funciona como sigue: si $|M|$ es un múltiplo positivo de n, entonces es posible calcular la función XOR de la clave *K2* de n-bit con el último bloque de M y así se calcula la CBC MAC con la clave *K1.* De lo contrario, se debe ampliar la longitud de M al siguiente múltiplo de *n* anexando un relleno *m*, se realiza la función XOR de la clave K3 de n-bit con el último bloque rellenado del mensaje, para así calcular la CBC MAC con la clave K1.

Los autores demuestran la seguridad de ésta y otras construcciones, dando límites concretos sobre la incapacidad de un adversario en términos de su inhabilidad para distinguir una permutación aleatoria del cifrador de bloques. Concluyen diciendo que su análisis explota nuevas ideas que simplifican pruebas en comparación con trabajos anteriores.

También en el año de 2005, Shiguo Lian, et al., realizaron en [19] un cifrador de bloques basado en una transformación caótica estándar, en este caso la transformación de TENT. Debido a sus características de ergodicidad, sensibilidad a las condiciones iniciales y sensibilidad para controlar los parámetros, etc., las transformaciones caóticas tienen un buen potencial para el cifrado de información. En este documento, los autores proponen un cifrado de bloque basado en la transformación caótica estándar, que se compone de tres partes: un proceso de confusión basada en la transformación caótica estándar, una función de difusión y un generador de claves. Ellos analizan la sensibilidad de parámetro que gobierna el comportamiento de la transformación caótica estándar, y el proceso de confusión basado en esta propuesta. En esta propuesta se hace uso de una función de difusión de alta velocidad, y se deriva un generador de clave basado en la transformación caótica Tent (transformación de la Tent) sesgado. En este trabajo se muestran algunos resultados de criptoanálisis sobre la seguridad en el diseño del cifrador. También se analiza su complejidad computacional. Los resultados experimentales muestran que el nuevo algoritmo de cifrado cuenta con una seguridad satisfactoria, a un bajo costo, lo que le hace un candidato potencial para el cifrado de datos multimedia como imágenes, audios e incluso videos.

En el año de 2008, Muhammad Asim y Varun Jeoti, proponen en [20] un método eficiente para diseñar S-CAJAS basado en transformaciones caóticas, las cuales juegan un rol central en diversos algoritmos criptográficos. El método propuesto se basa en la propiedad de mezclado de las transformaciones caóticas lineales por trazos, como lo es la transformación de la Tent. Las S-cajas se fabrican teniendo unas probabilidades de aproximación diferencial y lineal muy bajas.

También en el año de 2008, Charanjit S. Jutla, define en [21] un nuevo modo de operación para cifrados de bloques que, en adición a la confidencialidad proporcionada, también asegura la integridad del mensaje. En cambio, previamente para la integridad de un mensaje de acceso independiente, se requería calcular un código de autenticación de mensaje (MAC). El nuevo modo de operación, llamado Modo Paralelizable de integridad consciente, (IAPM: Integrity Aware Parallelizable Mode), requiere un total de *m+1* aplicaciones del cifrado de bloques en un texto

plano de longitud de *m* bloques. Para comparación, el bien conocido modo de cifrado CBC (encadenamiento de bloque cifrado) requiere *m* evaluaciones de cifrado de bloques, y el segundo paso de calcular el CBC-MAC requiere adicionalmente *m+1* evaluaciones de cifrado de bloque. Como el nombre lo sugiere, el nuevo modo es también altamente paralelizable.

Cifradores caóticos

Existen dos enfoques principales para el diseño de cifradores caóticos: analógico y digital. El primero de ellos se basa en el concepto de sincronización caótico [22]. En estos sistemas, la información puede transmitirse por la señal caótica en varias formas: chaotic masking [23-27], chaotic switching or chaos shift keying (CSK) [28-29], chaotic modulation [30-33], and chaos control [35-36]. Independientemente del método utilizado para transmitir la señal del mensaje, el receptor se tiene que sincronizar con el generador caótico del transmisor para regenerar la señal caótica *x(t)* y recuperar así el mensaje *m(t)*. El segundo acercamiento al diseño de sistemas criptográficos basados en el caos, consiste en utilizar computadoras digitales para recorrer un mapa caótico y la máscara del mensaje binario en un número de formas [37-42]. Estos cifradores no dependen de la sincronización. En su lugar, por lo general utilizan uno o más mapas caóticos donde el punto inicial x_0 y el valor del parámetro desempeñan el papel de la clave.

Transformaciones caóticas

Los cifradores basados en el caos aparecieron en los años noventa como una aplicación original de la dinámica no lineal en el régimen caótico [45]. [45]

Nuevos algoritmos basados en mapas caóticos se propusieron para protección de diferentes tipos de datos multimedia, especialmente imágenes y vídeos digitales en este período. Sin embargo, muchos de ellos fundamentalmente fueron viciados por la falta de solidez y seguridad [46].

En el año de 2007, Di Xiao, et al., basado en la propiedad de semi-grupo de la transformación caótica de Chebyshev y algunas mejoras eficaces de su propio protocolo original, proponen en [43] un protocolo para el acuerdo de claves basado en transformaciones caóticas, demostrando ser seguros, viables y extensibles.

También en 2007, Bonseok Koo, et al., diseñaron e implementaron en [44] un hardware unificado para cifradores de bloque ARIA y AES de 128 Bit. ARIA y el estándar Avanzado de Encriptación (AES: Advanced Encryption Standard) son la siguiente generación estándar de algoritmos de cifrado de bloque de Korea y de Estados Unidos, respectivamente. En ese artículo, los autores presentan un área eficiente de arquitectura de hardware unificado de ARIA y AES. Ambos algoritmos tienen estructuras de red de substitución y de permutación, (SPN: Substitution Permutation Network) de 128 bits, y sus capas de substitución y permutación podrían combinarse eficientemente. Así, los autores proponen una arquitectura de procesador de 128 bits con intercambio de recursos, que son capaces de procesar ARIA y AES. Esta es la primera arquitectura que soporta ambos algoritmos.

En el año de 2008, S. Behnia, et al., desarrollaron en [45] un algoritmo original para cifrar imágenes basadas en la mezcla de transformaciones caóticas. En ese documento, los autores informaron de una implementación de un esquema de cifrador de imagen digital basado en la mezcla de sistemas caóticos. La técnica de criptografía caótica utilizada por los autores de este artículo, es una criptografía de clave simétrica. En este algoritmo, S. Behnia, et al, mezclaron un mapa acoplado típico con un mapa caótico unidimensional y fue utilizado para cifradores de imagen de alta seguridad, mientras que su velocidad es aceptable. El algoritmo propuesto se describe en detalle, junto con su análisis de la seguridad y las posibles aplicaciones. Los resultados experimentales basados en la mezcla de mapas caóticos, obtenidos por los autores, aprueban la eficacia del método propuesto y la implementación del algoritmo. Esta aplicación de mezcla de transformaciones caóticas muestra ventajas del espacio de claves grandes y alta seguridad. El texto cifrado generado por este método es del mismo tamaño que el texto claro y es adecuado para su uso práctico en la transmisión segura de información confidencial en Internet.

En el año de 2009 S. Behnia, et al., presentan en [46] un nuevo tipo de algoritmo de cifrado de bloque de clave simétrica basado en transformaciones caóticas triples. En este algoritmo, la utilización de dos parámetros de acoplamiento, así como la mayor complejidad del criptosistema, hace una contribución al desarrollo de criptosistemas de mayor seguridad. Con el fin de aumentar la seguridad del algoritmo que ellos proponen, el tamaño del espacio de claves y la complejidad computacional de los parámetros de acoplamiento también deben aumentarse. Los resultados teóricos y experimentales afirman que el algoritmo propuesto tiene ventajas como una velocidad aceptable y su complejidad debido a la existencia de dos parámetros de acoplamiento y alta

seguridad. Se debe tener en cuenta que el texto cifrado tiene una distribución plana y tiene el mismo tamaño que el texto claro. Por lo tanto, es conveniente para uso práctico en comunicaciones seguras.

También en el año de 2009, Ali Kanso y Nejib Smaoui proponen en [47] dos generadores de secuencias binarias pseudo-aleatorias, basados en transformaciones caóticas logísticos destinados a las aplicaciones de cifrado de secuencias. El primero se basa en una única transformación logística unidimensional que exhibe propiedades aleatorias. Como un ruido similar al dado para ciertos valores de los parámetros, y el segundo se basa en una combinación de dos transformaciones logísticas. El paso de cifrado que los autores proponen en ambos algoritmos consiste en una simple operación XOR bit a bit de la secuencia de texto claro binario con la secuencia binaria de la secuencia de claves. Se aplica una función de umbral para convertir la iteración de un punto flotante en una forma binaria. Los resultados experimentales muestran que las secuencias producidas poseen alta complejidad lineal y muy buenas propiedades estadísticas que fueron descritas en [50]. Por último, los sistemas son presentados para la evaluación de seguridad por los comités de cifrado [50] y [51].

También en el año de 2009, Huaqian Yang et al., proponen en [48] un criptosistema basado en una transformación caótica, la transformación logística, y operaciones algebraicas. El algoritmo de cifrado que proponen, cifra texto claro de 128 bits para bloques de texto cifrado de 128 bits, utilizando una clave de 128 bits K y el valor inicial x_0 y el parámetro de control μ de la transformación logística. Consta de una permutación inicial y ocho rondas computacionalmente idénticas seguidas de una transformación de salida. R es una ronda que utiliza una clave de 128 bits $K^{(r)}$ para transformar un $C^{(r-1)}$ de entrada de 128 bits, que se alimenta a la siguiente ronda. El resultado después de 8 rondas entra en la transformación de salida para producir el texto cifrado final. Todas las claves de ronda se derivan de K y una secuencia binaria aleatoria de 128 bits generados por una transformación caótica. El análisis muestra que el cifrado de bloque propuesto no sufre de los fallos de criptosistemas caóticos puros y posee alta seguridad.

Fuyan Sun y Shutang Liu, propusieron en [49] un esquema para generación de secuencias binarias pseudo-aleatorias basado en la transformación caótica espacial. Para hacer frente al reto de utilizar la propuesta PRBS (Pseudo Random Binary Sequence) en criptografía, la propuesta PRBS se somete a pruebas estadísticas que son las bien conocidas, en el ámbito de la criptografía, FIPS-

140-1 [51] y se investigan las propiedades de correlación de las secuencias propuestas. La propuesta PRBS pasa con éxito todas estas pruebas. Los resultados que se encuentran alentadores, ya que las propiedades de correlación de la secuencia propuesta, sugiere una fuerte similitud con las secuencias aleatorias. Los resultados de pruebas estadísticas indican fuerte candidatura para las aplicaciones criptográficas, al solventar con éxito las pruebas propuestas en [51].

Comentarios al capítulo

En este capítulo se realizó una revisión general de las principales aportaciones de aquellos trabajos de investigación más sobresalientes a nivel internacional reportados en los últimos años, y con esto se conoce el estado actual de las implementaciones de las transformaciones caóticas unidimensionales en el diseño de cifradores de bloque caóticos ya que se han encontrado buenas propiedades estadísticas cuando se usan estos tipos de transformaciones caóticas para el cifrado de archivos.

CAPÍTULO 2

MARCO TEÓRICO

RESUMEN

Este capítulo se divide en dos grandes temas; en el primer tema se describen los conceptos de la Criptografía; y en el segundo, conceptos pertenecientes a la Teoría de la Información. Estos conceptos, que abarcan desde los tipos de cifradores, hasta las herramientas de evaluación como son la Entropía y la Información Mutua, son utilizados para el diseño y desarrollo del cifrador de bloques caótico.

2.1 Conceptos de Criptografía

Si bien hasta hace poco la criptografía[3] era un campo de uso reservado casi exclusivamente a diplomáticos y militares, la llegada de la era de la información, con sus avances tanto sociales como tecnológicos, ha traído consigo nuevas necesidades como es la de garantizar la privacidad de las comunicaciones. Además la enorme potencia de cálculo que permiten los procesadores actuales ha hecho de esta ciencia un tema de gran interés, principalmente por el abanico de aplicaciones, cada vez mayor, que puede soportar [52].

La Criptografía es la ciencia que trata de la protección de la información para evitar el acceso a ella por parte de personas no autorizadas. Para ello, deben realizarse un conjunto de transformaciones T_k a un mensaje o texto en claro m (que pertenece a un espacio de mensajes M) que se quiere transmitir de forma segura a través de un canal que se supone interceptado, obteniendo de esta forma un texto cifrado $c = T_k(m)$ también llamado criptograma (y que pertenece a un espacio de claves C). Este criptograma sólo podrá ser descifrado por el receptor legítimo mediante la aplicación de la transformación inversa T_k^{-1}, para obtener de ese modo el texto original m:

$$T_k^{-1}(c) = T_k^{-1}(T_k(m)) = m \qquad \textbf{(1)}$$

El conjunto de reglas de transformación $M \Leftrightarrow C$ se denomina algoritmo criptográfico y depende de un parámetro k que se denomina clave, que selecciona qué transformación del conjunto se va a utilizar. En aplicaciones prácticas, se puede suponer que el algoritmo de cifrado es conocido, por lo que la seguridad del sistema recaerá en el desconocimiento de la clave por parte de un posible atacante, ya que si éste no la conoce deberá probar, de manera exhaustiva, todas las posibles combinaciones. A este tipo de ataques se les suele denominar "por fuerza bruta".

La idea subyacente de la criptografía moderna es que los algoritmos criptográficos sean públicos, de forma que su seguridad no esté condicionada al hecho del desconocimiento de éste por parte de posibles atacantes. Sin embargo, en la práctica, la mayoría de los sistemas actuales mantienen los algoritmos de cifrado y descifrado de forma secreta como refuerzo adicional a la seguridad.

[3] La Criptología (del griego kripto: lo oculto, y logos: estudio) es el estudio de los criptosistemas. Sus áreas principales de estudio son la criptografía y el criptoanálisis.

Un sistema criptográfico debe intentar cubrir una serie de necesidades resultantes de la aplicación de un objetivo de seguridad para la información:

- Evitar que el mensaje sea descubierto (confidencialidad).
- Evitar que el mensaje pueda ser modificado de forma selectiva.
- Evitar la inserción de mensajes falsos.
- Detectar modificaciones del texto cifrado.

Para que un sistema sea de secreto perfecto, el número de claves (K) debe ser, al menos, igual al número de mensajes con probabilidad no nula. Como en la práctica no es posible generar una clave de longitud infinita, se tendrá que conformar con obtener una secuencia pseudo-aleatoria de periodo mucho mayor que la longitud del mensaje que se quiera proteger. Para ello se pueden emplear tanto técnicas de transposición (reordenación de los caracteres del texto original) como de sustitución (basadas en combinar el texto que se quiere transmitir con las secuencias producidas por generadores de secuencias pseudo-aleatorias), que para ser efectivos, estas últimas, deberán cumplir una serie de requisitos:

- Las transformaciones que realice el sistema han de ser reversibles, para todo mensaje producido por la fuente y para toda clave que pertenezca al espacio de claves:

  $$T_k^{-1}\left(T_k\left(m\right)\right) = m \qquad \text{donde } m \in M \text{ y } k \in K \qquad \textbf{(2)}$$

- La fortaleza del sistema debe basarse en la dificultad de obtener la clave a partir del propio sistema y del texto fuente, y no en el desconocimiento de la familia de transformaciones.
- Para dos familias de transformaciones análogas, tendrá mayor seguridad aquella que tenga un espacio de claves (K) mayor.
- El sistema de distribución de claves ha de ser suficientemente seguro, ya que la seguridad introducida por la trasformación está condicionada por el desconocimiento de la clave.

Además, al tratar un sistema de estas características se deben hacer una serie de hipótesis de trabajo previas, como el hecho de que el criptoanalista tiene un conocimiento completo del sistema de cifrado, ha conseguido una cantidad considerable de texto cifrado y conoce el texto en claro correspondiente a una cierta porción del texto cifrado. Si bien estas tres suposiciones pueden parecer pesimistas, son casi siempre realistas, y cualquier sistema de cifrado debe ser seguro bajo las mismas. Por supuesto, para cada situación, los términos considerable y cierta porción deben ser cuantificados. Sus valores precisos dependerán del sistema que se considere y

el nivel de seguridad requerido. Este nivel de seguridad vendrá determinado por el cumplimiento de una serie de condiciones:

- Para usuarios autorizados, las operaciones de cifrado y descifrado deben ser simples y eficaces.
- La operación de descifrado deberá ser muy complicada para los usuarios no autorizados.
- No debe ser una condición imprescindible que el algoritmo de cifrado se mantenga en secreto.
- La eficiencia y seguridad obtenida debe ser independiente de los datos que se vayan a transmitir.

La división más común entre los cifradores es la que los clasifica en cifradores de bloque o cifradores de flujo. Mientras que los cifradores en bloque dividen el texto en bloques de tamaño fijo, y operan de forma independiente sobre cada uno de ellos (son pues cifradores por sustitución simple y con un solo estado interno), los cifradores en flujo dividen el texto en caracteres y cifran cada unidad del mensaje (por ejemplo cada bit) mediante una función, cuya dependencia con el tiempo viene gobernada por el estado interno del cifrador, de forma que tras el cifrado de cada carácter, el dispositivo cambia de estado interno de acuerdo a una cierta regla. Esto hace que en un cifrador en flujo, dos caracteres iguales en el texto a cifrar puedan dar dos caracteres diferentes en el texto cifrado, a diferencia de lo que ocurre en los de bloque. Los cifradores en flujo pueden poseer varios estados internos y requieren de un vector de inicialización que será la clave del criptosistema. Los cifradores de flujo, por su forma de funcionamiento, necesitan disponer de una secuencia pseudo-aleatoria generada por algún método para realizar el cifrado.

Cifradores de flujo

Uno de los campos de interés dentro de la criptografía es el del cifrado en flujo. En este tipo de cifrado, la protección se basa en combinar la información a transmitir con la salida producida por un generador de secuencias pseudo-aleatorias, de forma que la seguridad final ofrecida dependerá de las propiedades de éste. Los cifradores de flujo intentan simular el sistema de cifrado *one-time-pad* que propuso G. Vernam en 1917. Tal como se ve en la figura 2.1, este sistema se basa en el uso de una secuencia de bits aleatoria de clave, que se genera por el procedimiento de tirar monedas de forma sucesiva e independiente (esto es equivalente a usar una fuente binaria simétrica o BSS, Binary Symmetric Source, que genere dicha secuencia).

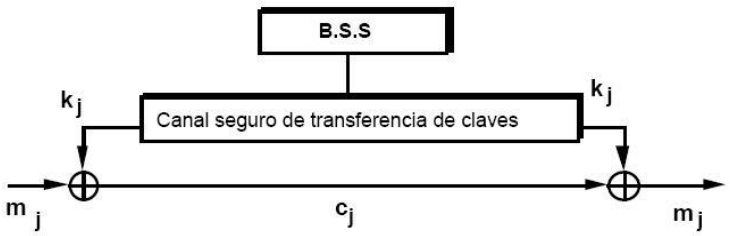


Figura 2.1 Cifrado de Vernam.

Posteriormente, esta secuencia (secuencia cifrante) se suma (módulo 2) bit a bit con el texto en claro para producir el texto cifrado. Un ejemplo de cómo obtener el texto cifrado es:

Texto en claro m:	011000111111101.....
Clave secreta k:	100110010001011.....
Texto cifrado c:	111110101110110.....

En este sistema, la clave debe ser tan larga como el mensaje y el conocimiento de parte de ella no debe dar información sobre ninguna otra parte. Si la clave k, que es precisamente la secuencia aleatoria, está realmente producida por una fuente binaria simétrica, es decir, por una fuente capaz de producir en cada instante un uno o un cero con probabilidad 1/2 independientemente de los bits producidos anteriormente, se puede considerar al sistema *one-time-pad* como un sistema completamente seguro contra cualquier tipo de ataque, que parta tan sólo del conocimiento del texto cifrado y no disponga de información del mensaje que se cifró. Se puede observar que si el atacante dispusiera además de parte del texto en claro (lo que puede darse con mucha probabilidad en la práctica), podría obtener, a partir de ambos, parte de la secuencia aleatoria. Esto no será un problema serio si la secuencia es realmente aleatoria. Sin embargo, si ésta está generada a partir de máquinas de estados finitos, puede representar un problema si la secuencia de dicho generador muestra una elevada predictibilidad, ya que un ataque criptoanalítico permitiría obtener la estructura que la produce y entonces el atacante podría, al disponer ya de la secuencia pseudo-aleatoria completa, descifrar totalmente todo el texto cifrado.

Aunque el sistema *one-time-pad* se considera, al menos teóricamente, incondicionalmente seguro debido a que presupone utilizar secuencias completamente aleatorias, presenta problemas de difícil solución en la práctica a la hora de generar y distribuir las claves. Esto es debido a que,

puesto que el flujo de bits de clave debe ser tan largo como el mensaje a cifrar, y debe ser totalmente aleatorio, la única manera posible de que el receptor use la misma secuencia que el transmisor es enviándosela (¡de forma segura!) antes de enviarle el criptograma o mensaje cifrado, para que pueda descifrarlo.

Los cifradores de flujo intentan solucionar el problema de la gestión (distribución) de claves usando un algoritmo determinista y una clave finita para generar un flujo de bits aleatorios que se sumará, bit a bit, al texto en claro para cifrarlo, tal como se muestra en la figura 2.2. La clave finita (y secreta) k debe tener una longitud L manejable para facilitar su intercambio, y el algoritmo debe ser lo suficientemente eficiente como para generar el flujo de bits aleatorios a la velocidad que requiera la aplicación que lo utilice. Evidentemente, hay un compromiso entre la seguridad que se podrá alcanzar y la longitud del flujo de bits aleatorios obtenidos dada una longitud de clave.

Hay dos puntos cruciales a la hora de establecer los criterios de diseño de cifradores en flujo. El primero es la velocidad a la que se deben generar los bits (es decir, lo eficiente que debe ser el algoritmo de generación o, dicho de otra forma, cómo penaliza a la velocidad de la comunicación el hecho de poner el cifrador de flujo) y, en segundo lugar, lo seguro que se necesite que sea el sistema (cuya mejor medida es el tiempo que se tardaría en romperlo).

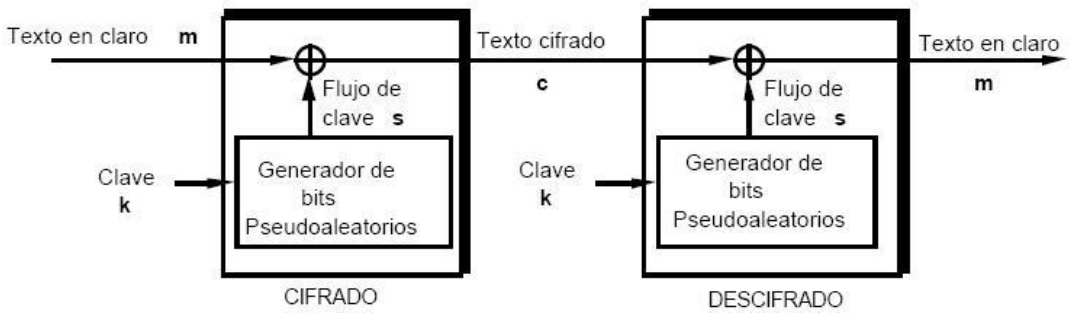


Figura 2.2 Cifrador en flujo.

Un cifrador de flujo rompe el mensaje en claro m ($m \in M$) en un flujo sucesivo de caracteres m_1, m_2, m_3,..., y cifra cada carácter m_i mediante una transformación T variante en el tiempo bajo el control del símbolo s_i de la secuencia pseudo-aleatoria en el instante i, con lo que se obtiene un nuevo carácter de la secuencia de texto cifrado $c_i = T(m_i, k_i)$. Tras cada carácter cifrado, el dispositivo cambia de estado interno de acuerdo con una cierta regla. En esta aplicación, a la secuencia pseudo-aleatoria se la denomina flujo de clave. El flujo de clave (o *keystream* en inglés)

$s=s_1, s_2, s_3,...$, se obtiene a partir de la clave secreta k ($k \in K$). El mecanismo por el cual se expande la clave secreta k, que suele ser de corta longitud, en un gran flujo de bits pseudo-aleatorio s, se le llama generador pseudo-aleatorio. Los cifradores de flujo se usan ampliamente para obtener la confidencialidad de la información.

Se puede concluir que la transformación T consiste en combinar los caracteres que constituyen el mensaje a transmitir con los caracteres de salida de un generador de bits pseudo-aleatorios y éste consistirá en una familia de funciones $\{G_L : L \in N\}$ que extenderá una clave k ($k \in K$) de longitud corta L a una secuencia s de longitud P mucho mayor:

$$G_L : \{0,1\}^L \rightarrow \{0,1\}^P \quad s^P = G_L(k^L) \qquad \textbf{(3)}$$

Modos de operación de un cifrador en flujo

La teoría de autómatas y, en especial, la de máquinas de estado finito, es muy útil para describir los sistemas de cifrado en flujo y sus diferentes modos de operación. Los alfabetos empleados para el texto en claro y para el texto cifrado se determinan normalmente en función de la aplicación que se considere, aunque en la mayoría de los casos se usa el alfabeto binario para ambos, ya que es el utilizado en comunicaciones digitales. Los cifradores en flujo se dividen normalmente en dos tipos: los síncronos y los autosincronizantes[4][52].

Cifradores de bloque

Los cifradores de bloque con clave simétrica son los elementos más destacados e importantes en muchos sistemas criptográficos. Individualmente, ellos proveen confidencialidad. Como un componente fundamental, su versatilidad permite la construcción de generadores de números pseudo-aleatorios, cifradores de flujo, MACs y funciones hash. Ellos, además, pueden servir como un componente central en las técnicas de autenticación de mensajes, mecanismos de integridad de datos, protocolos de autenticación de entrada, y sistemas de firma digital (clave simétrica).

No hay un cifrador de bloques ideal para todas las aplicaciones, incluso uno que ofrezca un alto nivel de seguridad. Los cifradores por bloques operan en grupos de bits de longitud fija, llamados

[4] Estos dos tipos de cifradores de flujo serán descritos ampliamente en el Apéndice B.

bloques. Para el proceso de cifrado el cifrador de bloques toma un bloque del mensaje original como entrada y produce un bloque de igual tamaño del criptograma. La transformación exacta es controlada utilizando una segunda entrada, la clave de cifrado, que puede ser la pública o la privada. El descifrado es similar: se ingresan bloques del criptograma y se producen bloques del mensaje original recuperado.

Red de Feistel balanceada

En criptografía, el Cifrado de Feistel es un método de cifrado de bloque con una estructura particular. Debe su nombre al criptógrafo de IBM Horst Feistel. También es conocida comúnmente como Red de Feistel. Un gran número de algoritmos de cifrado por bloques lo utilizan, siendo el más conocido el algoritmo Data Encryption Standard (DES). Las redes de Feistel presentan la ventaja de ser reversibles por lo que las operaciones de cifrado y descifrado son idénticas, requiriendo únicamente invertir el orden de las subclaves utilizadas. Este algoritmo se denomina simétrico por rondas, es decir, realiza siempre las mismas operaciones un número determinado de veces (denominadas rondas). Los pasos de la red de Feistel son en esencia los siguientes [61]:

1. Se selecciona una cadena, N, normalmente de 64 o 128 bits, y se la divide en dos subcadenas, L*(izquierda)* y R *(derecha)*, de igual longitud ($N/2$)
2. Se toma una función, **F**[5], y una clave K_i
3. Se realizan una serie de operaciones complejas con **F** y K_i y con L o R (solo uno de ellas)
4. La cadena obtenida se cambia por la cadena con la que no se han realizado operaciones, y se siguen haciendo las rondas.

Las operaciones básicas de una red de Feistel son las siguientes: se descompone el mensaje original en dos piezas iguales, (L_0, R_0). Para realizar el cifrado en cada ronda $i=1, 2, \ldots, n$, se calcula

$$L_i = R_{i-1},$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i). \qquad \textbf{(4)}$$

[5] F es la función de transformación de cifrado o descifrado.

donde f es una función y K_i son cada una de las subclaves aplicadas a cada iteración, ver figura 2.3. El texto cifrado viene dado por la concatenación de L_n y R_n.

Para el descifrado las operaciones que hay que realizar son:

$$L_n = L_{n-1} \oplus f(R_{n-1}, K_n),$$
$$R_n = R_{n-1}. \qquad \textbf{(5)}$$

Una ventaja de este modelo es que la función F usada no tiene por qué ser reversible, pudiendo ser todo lo complicada que se desee, esta cualidad permite a los criptógrafos concentrarse en la seguridad de dicha función sabiendo que el proceso de descifrado está garantizado ya que la propia estructura de la red de Feistel es reversible. Para ello únicamente requiere que se invierta el orden de las subclaves utilizadas. Una variación del esquema de Feistel son las redes de Feistel no balanceadas en las que las mitades del texto en plano L_0 y R_0 son de diferente longitud. Un algoritmo de cifrado que utiliza esta variación es el algoritmo Skipjack [53].

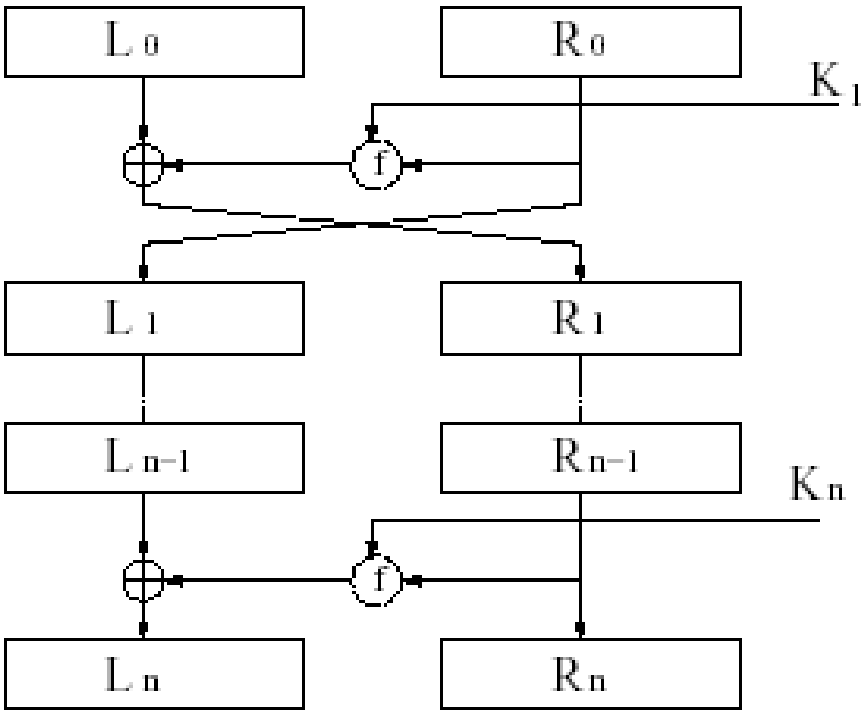


Figura 2.3 Esquema general de las redes de Feistel.

Red de Feistel desbalanceada

Una red de Feistel desbalanceada (UFN; Unbalanced Feistel Networks) [54], es una red donde el bloque del lado izquierdo y el bloque del lado derecho no son del mismo tamaño, tal como se puede observar en la figura 2.4.

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

$$X_{i+1} = \left(F\left(msb_s\left(X_i\right), k_i\right) \oplus lsb_t\left(X_i\right)\right) // \; msb_s\left(X_i\right) \qquad \textbf{(6)}$$

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 se encuentran 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.

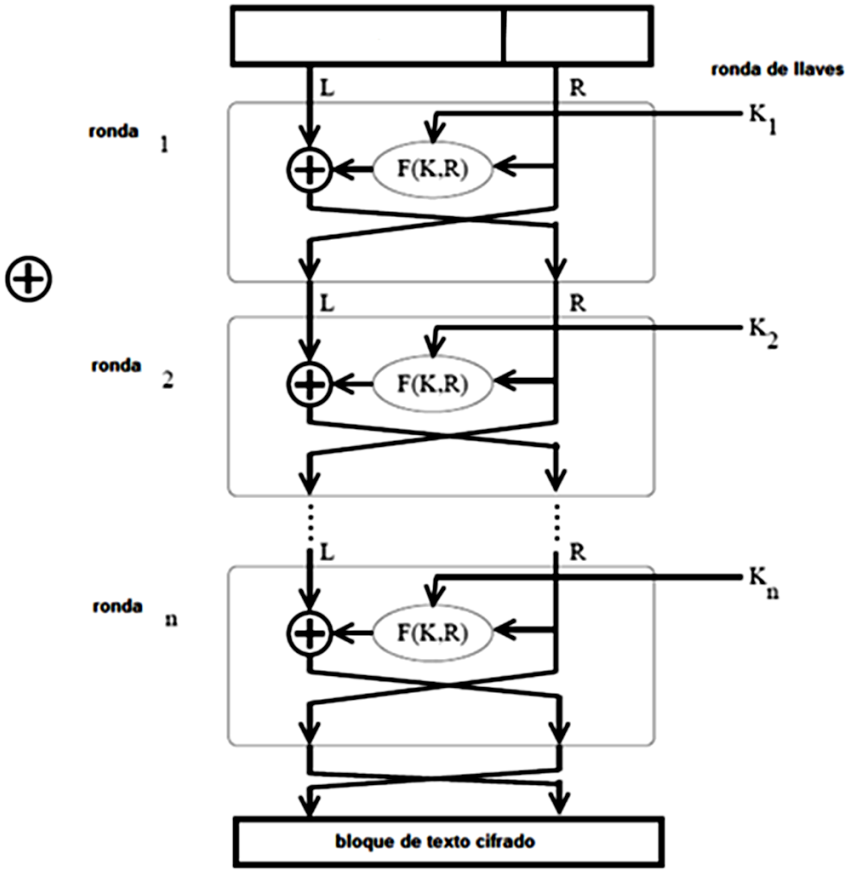


Figura 2.4 Red de Feistel desbalanceada.

2.2 Conceptos de Teoría de la Información

La Teoría de la Información es un área de las matemáticas y de las comunicaciones, cuyos conceptos pueden ser aplicados para la evaluación de secuencias de ruido producidas para sistemas de protección de información; para ello, se 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 Claude 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.

Distribución estadística

La Teoría de la Información, introducida por Claude E. Shannon a finales de los años cuarenta, permite efectuar una aproximación formal al estudio de la seguridad de cualquier algoritmo criptográfico [55].

El enfoque de la Teoría de la Información analiza la estructura matemática y estadística de los mensajes, con independencia de su significado. Los aspectos en los que se interesa la Teoría de la Información son la capacidad de transmisión de los canales, la compresión de datos o la detección y corrección de errores, la criptografía y temas relacionados [56].

Para explicar mejor la Teoría de la Información, a continuación se precisa el concepto de cantidad de información. Para ello, se analiza el siguiente ejemplo:

Se tiene una moneda cargada. Del lado A se tiene el 90% de probabilidad de caer de ese lado, el lado B tiene el 10% de probabilidad. ¿Cuánta información se obtiene al lanzarla cae del lado B? La respuesta es que aporta mucha información, ya que se era muy probable de que caería en el lado A. Si por el contrario, hubiera caído del lado A no sería extraño, es decir, suministra poca información. Se puede entender a la cantidad de información como una medida de la disminución de incertidumbre sobre un evento. La cantidad de información es proporcional a la probabilidad de un evento.

Con objeto de simplificar la notación se empleará:

v Variable aleatoria para representar los posibles eventos.
x_i *i*-ésimo evento.
$p(x_i)$ Probabilidad asociada a dicho evento.
n Número de posibles eventos.

Suponga que el único valor que puede tomar v es x_i. Saber el valor de v 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 proporciona bastante información. Este concepto de información es cuantificable y se puede definir de la siguiente forma:

$$I_i = -\log_2(p(x_i)) \quad \textbf{(7)}$$

siendo $p(x_i)$ la probabilidad del estado x_i. Obsérvese que si la probabilidad de un estado fuera 1 (máxima), la cantidad de información que aporta sería igual a 0, mientras que si su probabilidad se acercara a 0, tendería a +∞ —esto es lógico, un evento que no puede suceder aportaría una cantidad infinita de información si llegara a ocurrir—. Una secuencia aleatoria de bits 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 pseudo-aleatorios.

Entropía

Sumando ponderadamente las cantidades de información de todos los posibles estados de una variable aleatoria x, se obtiene:

$$H(x) = \sum_{i=1}^{n} p_i \log_2\left(\frac{1}{p_i}\right) = -\sum_{i=1}^{n} p_i \log_2(p_i) \quad \textbf{(8)}$$

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 $\log\left(\frac{1}{0}\right)$ 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. Obsérvese que la entropía es proporcional a la longitud media de los mensajes que se necesitaría para codificar una serie de valores de V de manera óptima, dado un alfabeto cualquiera. Esto quiere decir que cuanto más probable sea un valor individual, aportará menos información cuando aparezca, y podrá codificarlo empleando un mensaje más corto. La entropía máxima en un archivo ocurre cuando la huella estadística de dicho archivo es uniforme. Entonces lo que se busca al tratar de tener un sistema equiprobable es llevar el criptograma al estado de mayor incertidumbre. La cantidad de información asociada al suceso más simple, que consta únicamente de dos posibilidades equiprobables será la unidad a la hora de medir esta magnitud, y se denomina bit. Esta es precisamente la razón por la que se emplean logaritmos base 2, para que la cantidad de información del evento más simple sea igual a la unidad. Se puede decir que la entropía de una variable aleatoria es el número medio de bits que se necesita para codificar cada uno de los estados de la variable, suponiendo que se exprese cada evento empleando un mensaje escrito en un alfabeto binario.

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 [55]:

$$I(X,Y) = H(Y) - H(Y/X) \quad \textbf{(9)}$$

La explicación intuitiva de esta magnitud es la siguiente. Inicialmente, se posee un grado determinado de incertidumbre sobre la variable aleatoria Y. Si antes de medir una realización concreta de Y, medimos la de otra variable X, parece lógico que nuestra incertidumbre sobre Y se reduzca o permanezca igual. Por ejemplo, supongamos que Y representa la situación meteorológica (lluvia, sol, viento, nieve, etc.), mientras que X representa el atuendo de una persona que entra en nuestra misma habitación. Inicialmente se tendrá un nivel determinado de entropía sobre Y. Si, acto seguido, la citada persona aparece con un paraguas mojado, seguramente para nosotros aumentará la probabilidad para el valor *lluvia* de Y, modificando su entropía. Esa disminución -o no- de entropía es precisamente lo que mide $I(X,Y)$.

Las propiedades de la cantidad de información entre dos variables son las siguientes:

I. $I(X,Y) = I(Y,X)$
II. $I(X,Y) \geq 0$ **(10)**

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 [55]. Es decir:

$$I(C,M) = 0 \quad \textbf{(11)}$$

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 hasta 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, porque si la clave es tanto o más larga que el mensaje, a la hora de protegerla se encontrará con el mismo problema que se tenía para proteger el mensaje.

Comentarios al capítulo

En este capítulo se estudian los conceptos de la Criptografía, conociendo sobre los 2 grandes tipos de cifradores, los de flujo y los de bloque, así como la implementación de este último en Redes de Feistel para implementar un criptograma; También se estudian los conceptos de la Teoría de la Información, conociendo los conceptos de la Distribución estadística, Entropía e Información Mutua, para poder aplicarlos más tarde en la evaluación de los archivos cifrados.

CAPÍTULO 3

TRANSFORMACIÓN CAÓTICA TENT

RESUMEN

En este capítulo se describen los conceptos de la Transformación Caótica Unidimensional en general, y de la Tent en particular. Además se muestran los Diagramas de Trayectorias y de Bifurcación, así como la Distribución estadística y el Exponente de Lyapunov de Transformación Caótica Unidimensional de la Tent. Al final del capítulo se realiza el escalamiento y discretización de la función al dominio ASCII Extendido.

3.1 Transformaciones caóticas unidimensionales

Una Transformación caótica unidimensional (1-D), es un sistema dinámico caótico (SDC) que está dado por la siguiente transformación,

$$X_{n+1} = \tau_\mu(X_n), \qquad n = 0,1,2\cdots \tag{12}$$

con

$$X_n = \tau_\mu^n(X_0) \in I, \qquad n = 0,1,2\cdots \tag{13}$$

La transformación $\tau_\mu(\cdot): I \to I$ es un mapeo no lineal, I denota el intervalo dentro del cual queda definido y n indica el índice de la iteración actual de la función $\tau_\mu(\cdot)$, comenzando con la condición inicial x_0 y usando el parámetro μ. Estos mapeos producen órbitas, las cuales quedan definidas por,

$$\phi(x_0) = \{x_0, x_1, x_2, \cdots, x_n, \cdots\} = \{x_n\}_{n=0}^{\infty}, \tag{14}$$

Cada una de estas órbitas depende de la condición inicial x_0 impuesta para la transformación. Es posible que estas órbitas tiendan a uno o más valores, en tal caso, estos valores constituyen los puntos fijos atractores u órbitas estables del mapeo. Pero también es posible que existan otros valores de los que se alejan las órbitas, siendo estos los puntos fijos repulsores u órbitas inestables del mapeo. Ahora bien, la condición que hace que un punto fijo sea atractor o repulsor depende del valor que tenga el parámetro μ. Un punto fijo de una función se define por,

$$\tau_\mu(x_1^*) = x_1^* \tag{15}$$

y será estable si

$$\lim_{n\to\infty} \quad \tau_\mu^n(x_0) = x_1^* \tag{16}$$

Por lo que se pueden resaltar tres aspectos importantes,

i. La órbita depende del valor inicial,

ii. La órbita también depende del parámetro μ,

iii. Pueden existir órbitas estable de periodo $p = 2^k, k \in Z^+$

La decisión de estudiar transformaciones caóticas unidimensionales se debe a que permiten, de manera sencilla y bajo condiciones específicas, generar señales o secuencias con distribuciones cercanas a la distribución uniforme, dependiendo de la transformación que se utilice, y del valor del parámetro que gobierne su comportamiento [57].

Transformación Caótica Tent

Una clase amplia de transformaciones utilizada en la literatura son las transformaciones unimodales: funciones con una única moda (máximo). Su definición formal se proporciona a continuación. Sea una aplicación $f : I \to I$ con $I = [e_0, e_2]$. Se dice que la transformación $f(x)$ es unimodal si existe un punto e_1, con $e_0 < e_1 < e_2$, tal que $f(x)$ es monótona estrictamente creciente en el intervalo $[e_0, e_1)$, y monótona estrictamente decreciente en el intervalo $[e_1, e_2]$.

Las transformaciones unimodales son transformaciones caóticas simples y extendidas. Las dos clases de transformaciones caóticas unimodales más importantes utilizadas son la familia de la transformación Tent, y las transformaciones polinómicas, dentro de los cuales la más conocida es la transformación logística.

La familia de la transformación Tent es una subclase de mapas lineales a tramos, PWL, que constan únicamente de dos intervalos. Dentro de esta familia a su vez se pueden distinguir cuatro subfamilias: la transformación Tent (TM), la transformación Tent sesgado o "skew Tent-map" (SK-TM), la transformación Tent simétrico o "symmetric Tent-map" (S-TM), y la transformación Tent sesgado bipolar o "bipolar skew Tent-map" (BSK-TM) [58]. La transformación Tent es la que da nombre a la familia, y el más utilizado en la literatura. Se trata de una aplicación $f : [0,1] \to [(1-\mu)/2, (1+\mu)/2]$ con $0 < \mu \leq 1$ en general, de la forma [59]

$$f(x) = \begin{cases} \mu x, & 0 \leq x < 0.5; \\ \mu(1-x), & 0.5 \leq x \leq 1. \end{cases} \qquad \textbf{(17)}$$

La forma del TM para $\mu = 1$ se muestra en la siguiente figura 3.1.

Como se puede apreciar, la transformación Tent tiene un único máximo en x = 0,5. Obviamente se trata de un mapa PWL con dos intervalos, E1 = [0, 0.5) y E2 = [0.5, 1].

La expresión habitualmente utilizada para el TM resulta

$$f(x) = \mu\left(1 - 2\left|x - 0.5\right|\right) \qquad \textbf{(18)}$$

El comportamiento del TM viene definido por el parámetro μ. Este parámetro de control se suele denominar parámetro de bifurcación en el caso de las transformaciones unidimensionales, ya que determina su comportamiento asintótico.

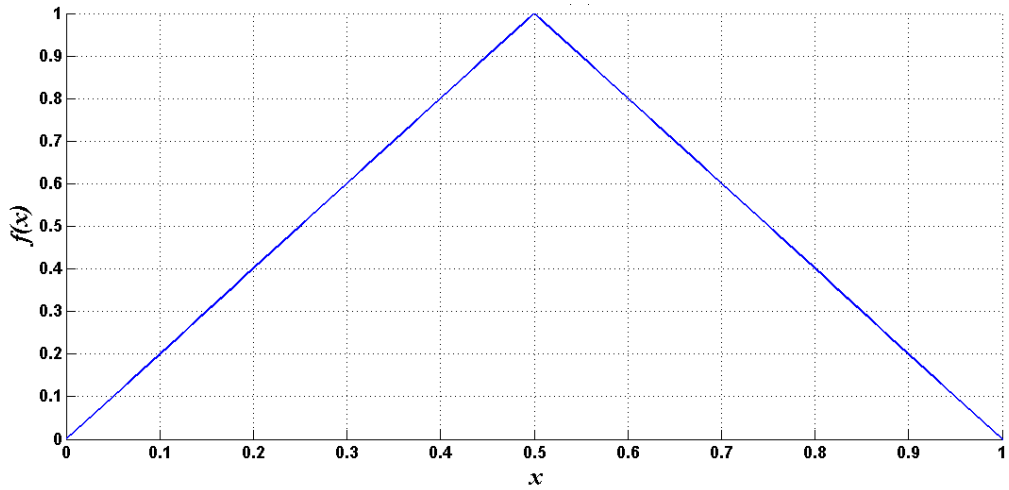


Figura 3.1 Transformación Caótica unidimensional de la Tent con μ =1.

Para la implementación de esta transformación en el cifrador, se requiere de una forma equivalente que permita representar a la transformación Tent para los fines de este trabajo y para poder expresarla se usa la ecuación:

$$f(x) = \begin{Bmatrix} 2\mu x_n + \dfrac{(1-\mu)}{2}; \ldots\ldots\ldots 0 \le x_n \le \dfrac{1}{2} \\ 2\mu(1-x_n) + \dfrac{(1-\mu)}{2}; \ldots \dfrac{1}{2} < x_n \le 1 \end{Bmatrix} \qquad \textbf{(19)}$$

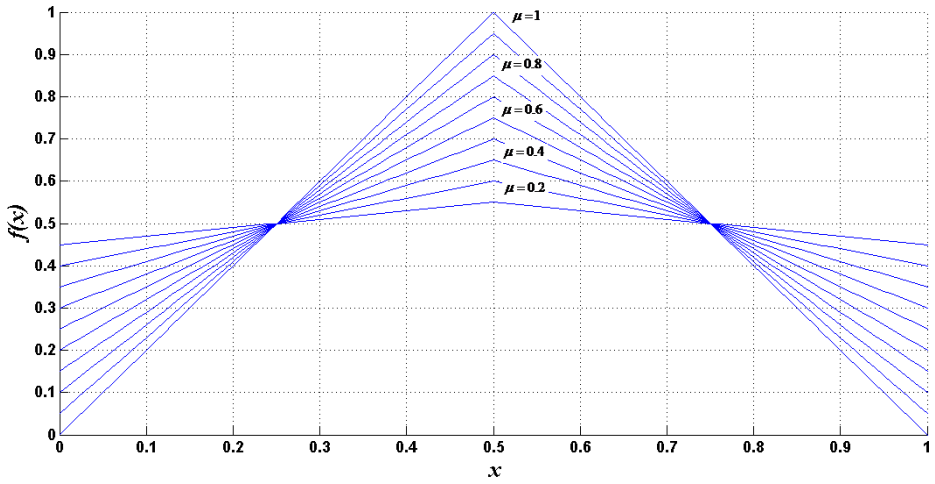


Figura 3.2 Familia de la Transformación Caótica Tent, para parámetros desde $\mu = 0.1$ ***hasta*** $\mu = 1$.

3.2 Caracterización de la Transformación caótica

Diagrama de Trayectorias

Para iterar la función se necesita un valor inicial y el resultado será la siguiente entrada de la función, como se puede observar,

$$\begin{aligned} x_1 &= f(x_0), \\ x_2 &= f(x_1) = f(f(x_0)) = f^2(x_0), \\ &\vdots \\ x_n &= f(x_{n-1}) = f(f^{n-1}(x_0)) = f^n(x_0). \end{aligned} \qquad \textbf{(20)}$$

donde x_n es la *n*-ésima iteración de x_0. El conjunto de todas las iteraciones de una función es llamado transformación de la función. Una manera 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 se comienza a dibujar líneas siguiendo los puntos $(x_0, x_0), (x_0, f(x_0) = x_1)$, $(x_1, x_1), (x_1, f(x_1) = x_2)$, $(x_2, x_2), (x_2, f(x_2) = x_3), \ldots$ se observa que la iteración se puede hacer geométricamente trazando líneas desde $f(x)$ hasta la línea $y = x$ y de regreso a $f(x)$. En la figura 3.3 se observa este comportamiento. A esto se le llama diagrama de trayectorias.

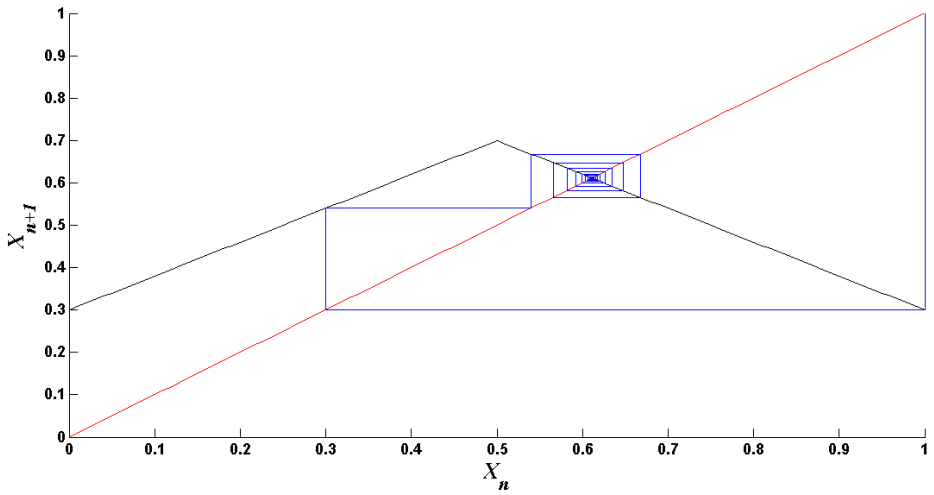


Figura 3.3 Función iterada con $\mu = 0.4$, $x_0 = 1$ ***y 1000 iteraciones.***

Sin embargo, como se puede observar, el sistema tiende a comportarse de una manera estable, lo que no es de interés, ya que se busca el caos total.

En la figura 3.4, se observa que se acerca al caos al variar el parámetro μ que rige la transformación. Finalmente, utilizando la μ máxima, se llega al caos, como se puede observar en la figura 3.5. Cabe hacer notar que esta función llega al caos con una $\mu = 0.9999$, más sin embargo, y comprobando la Teoría del Caos, que dice que es altamente sensible a las condiciones iniciales, si la $\mu = 1$, el sistema se comporta de una manera estable.

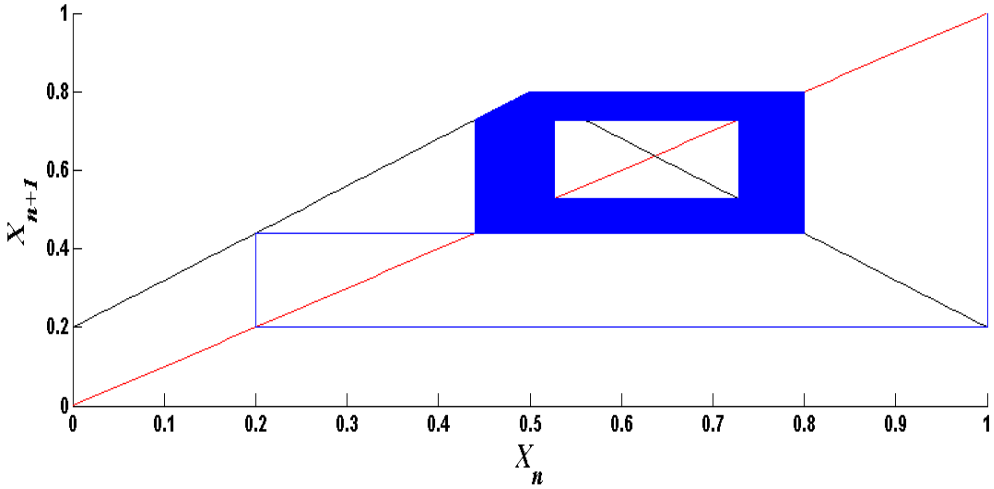


Figura 3.4 Función iterada con $\mu = 0.6$, $x_0 = 1$ ***y 1000 iteraciones.***

Con este método es difícil determinar a partir de qué valor el valor μ genera caos, por lo cual se utiliza otro método más efectivo para determinarlo, el cual es, el diagrama de bifurcación.

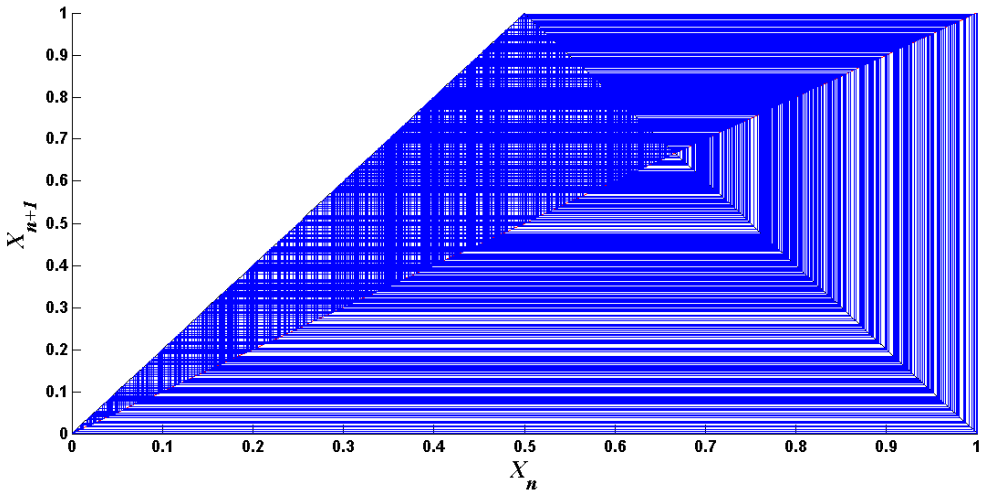


Figura 3.5 Función iterada con $\mu = 0.9999$, $x_0 = 1$ ***y 1000 iteraciones.***

Diagrama de Bifurcación

En los sistemas dinámicos un diagrama de bifurcación es una herramienta que muestra las posibles órbitas a las que tiende la transformación, como una función del parámetro que lo gobierna, y permite visualizar donde existe el comportamiento periódico y donde existe el comportamiento caótico. Comúnmente se grafica en el eje horizontal el parámetro que lo gobierna (μ) y en el eje vertical las posibles órbitas del sistema (transformación caótica) [60].

Para construir el diagrama de bifurcación se toma en cuenta el siguiente procedimiento:

a) Se selecciona una condición inicial aleatoria de $x_0 \in (0,1)$ y se itera la transformación $n > 500$ veces, calculando $x_1, x_2, x_3, \cdots x_n$.
b) Se desprecia las primeras 100 iteraciones para garantizar que se ha superado la etapa transitoria.
c) Se grafica las iteraciones restantes.

Se puede ver el resultado de este procedimiento para valores del parámetro de $\mu \in (0,1)$ en la figura 3.6. Como se observa claramente, a partir de la $\mu > 0.72$ la transformación entrará el fenómeno de doblamiento del periodo.

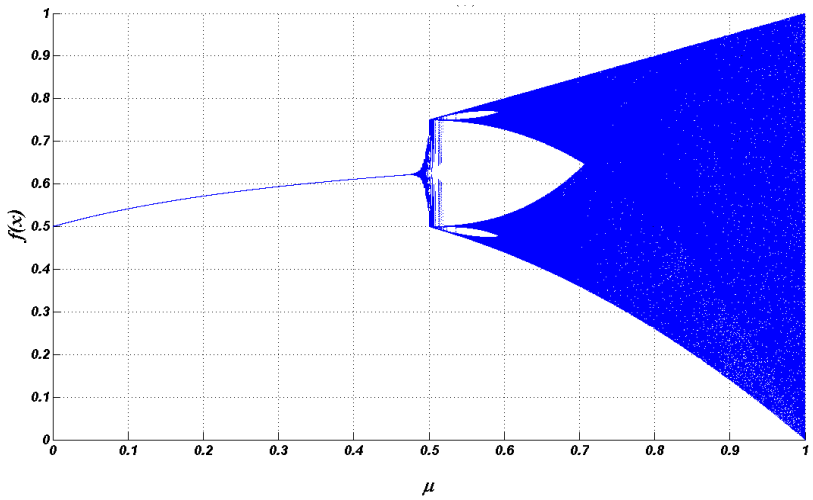
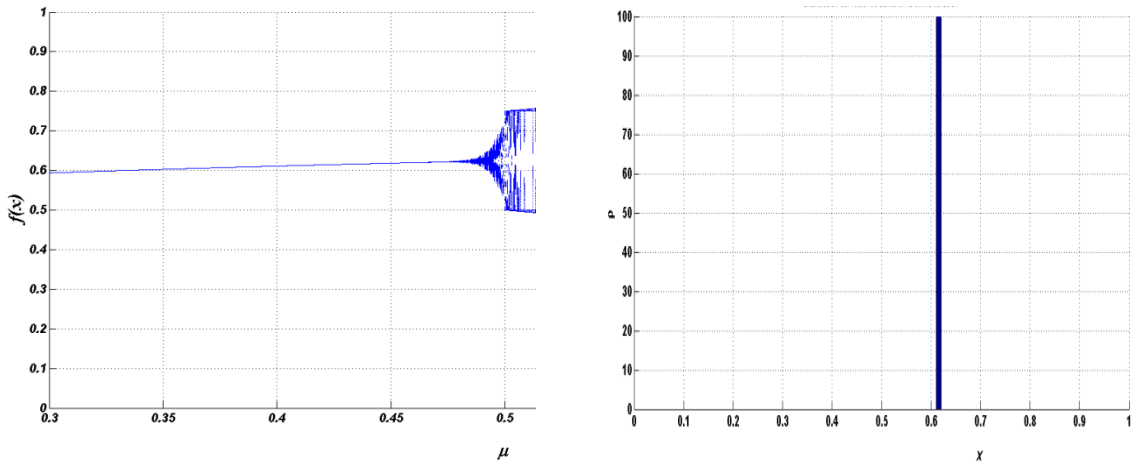


Figura 3.6 Diagrama de bifurcación de la Transformación Caótica unidimensional de la Tent.

Distribución Estadística

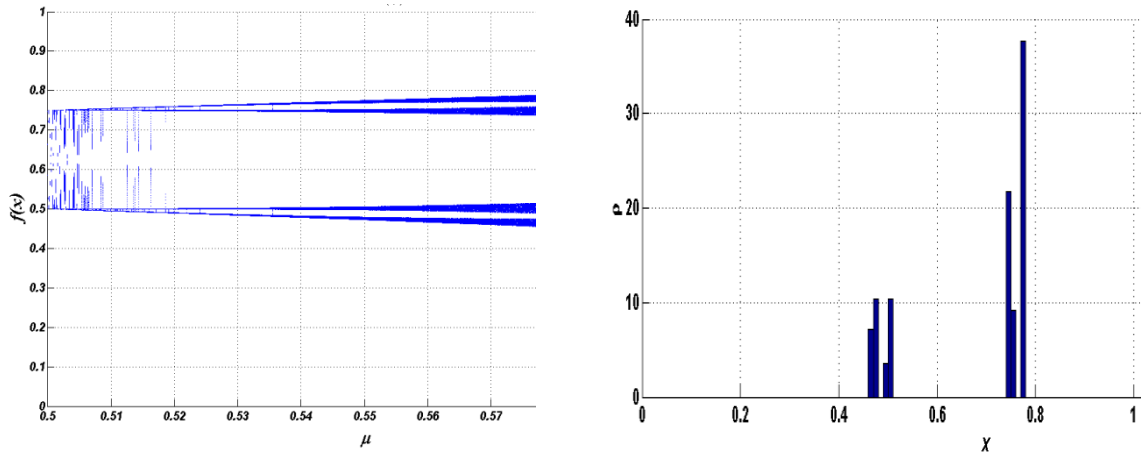
Cuando el sistema es caótico, no se puede determinar el comportamiento de la órbita a largo plazo. Entonces se requiere un análisis estadístico de la órbita, el cual consiste en averiguar qué tan frecuentemente la órbita visita diferentes regiones, dando lugar a un histograma asociado a esta órbita. 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 $\mu < 0.4$, la distribución invariante estará compuesta por una delta de Dirac; en tanto que, para valores de $\mu \in (0.52, 0.58)$ se presentarán tres deltas de Dirac. Finalmente, para valores de $\mu > 0.999$ la distribución cubre un solo intervalo. Lo anterior se puede observar en las figuras 3.7 a 3.9 donde se muestra que 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.



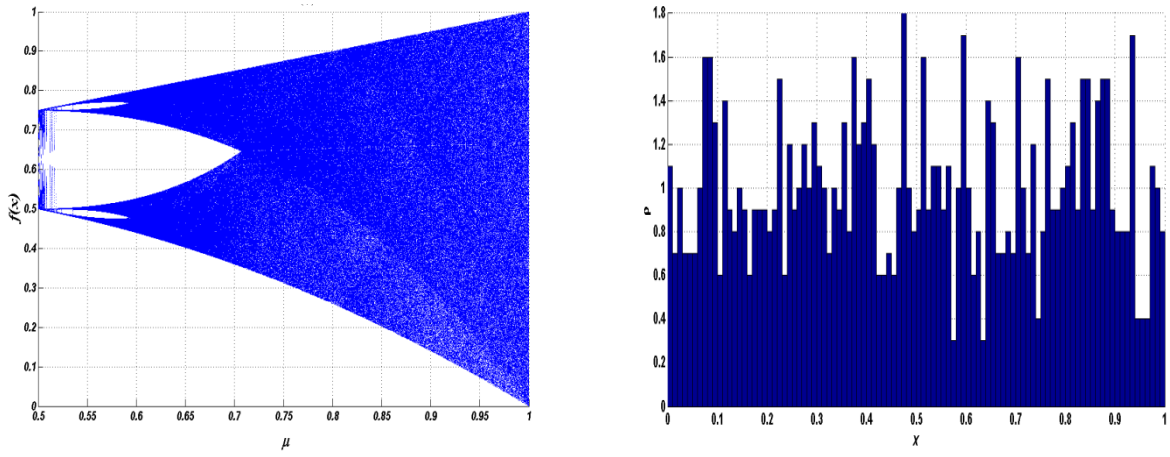
a) **b)**

Figura 3.7 Transformación Tent para $\mu = 0.4$. a) Diagrama de bifurcación; b) Densidad de Probabilidad



a) ***b)***

Figura 3.8 Transformación Tent para $\mu = 0.56$. a) Diagrama de bifurcación; b) Densidad de Probabilidad



a) ***b)***

Figura 3.9 Transformación Tent para $\mu = 0.999$. a) Diagrama de bifurcación b) Densidad de Probabilidad

Exponente de Lyapunov

Una transformación caótica puede exhibir órbitas aperiódicas para ciertos valores del parámetro, pero surge la pregunta, ¿Cómo se puede estar seguros de que esta condición es realmente caos? Para que un sistema se llame caótico debería mostrar dependencia sensitiva a condiciones iniciales, en el sentido de que dos órbitas muy cercanas se separen rápidamente, en promedio, en forma exponencial. Una forma de cuantificar esta dependencia es usando el exponente de Lyapunov.

Para mostrar esto considere alguna condición inicial x_0, ahora considere un punto cercano, $x_0 + \delta_0$, donde la separación inicial δ_0 es extremadamente pequeña. Sea δ_n la separación después de n iteraciones. Si $|\delta_n| \approx |\delta_0| e^{n\lambda}$, entonces λ es el exponente de Lyapunov. Un exponente positivo es signo de caos.

El exponente de Lyapunov puede aproximarse por la siguiente expresión:

$$\lambda \approx \frac{1}{n}\sum_{i-0}^{n-1} \ln\left| f'(x_i)\right| \qquad \textbf{(21)}$$

Si esta expresión tiene límite cuando $n \to \infty$, entonces el exponente de Lyapunov para la órbita que inicia en x_0 queda determinado por:

$$\lambda = \lim_{n\to\infty}\left\{\frac{1}{n}\sum_{i-0}^{n-1} \ln\left| f'(x_i)\right|\right\} \qquad \textbf{(22)}$$

Note que λ depende de x_o. Sin embargo, es la misma para todas las x_o en la cuenca de atracción de un atractor dado. Para puntos y ciclos estables, λ es negativa. Para atractores caóticos es positivo.

En la figura 3.10 se presume la forma de interpretar el exponente de Lyapunov sobre la transformación Tent, éste 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. Para la transformación Tent el valor de $\mu=0.499999$ se observa que a partir de este valor el comportamiento se vuele positivo.

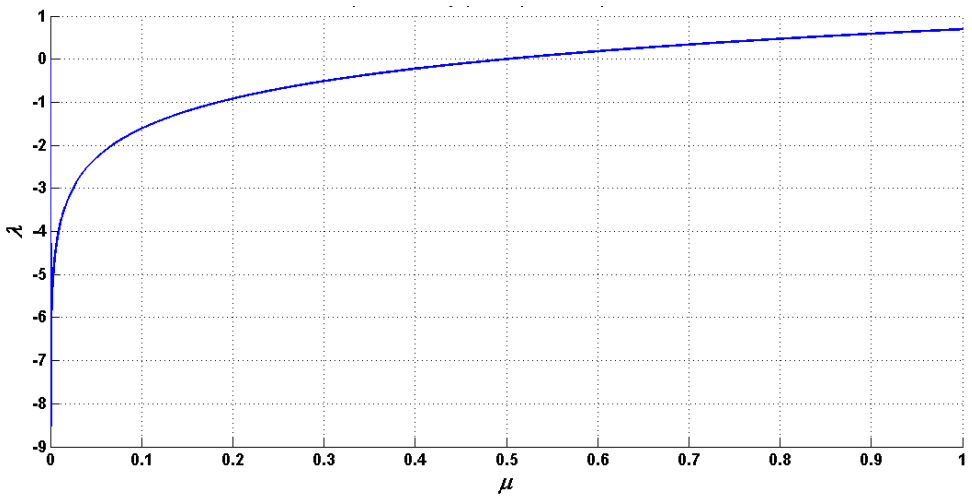


Figura 3.10 Exponente de Lyapunov para la Transformación Tent.

3.3 Escalamiento y Discretización al dominio ASCII EXT

La función de transformación corresponde a la transformación Tent, con $\mu \in (0,1)$ y $x \in (0,1)$. Sin embargo, en este dominio el cifrador no puede trabajar, ya que para hacerlo, el dominio utilizado es el del alfabeto ASCII Extendido, el cual se encuentra definido en el intervalo [0, 255]. Por lo tanto, la transformación Tent se tiene que escalar y discretizar del intervalo (0, 1) en los números reales, al intervalo (0, 2^n) en los números enteros, donde 2^n representa el número de bits a los cuales se requiere hacer el escalamiento; en esta tesis, se utilizaron 2 diferentes tipos, 2^8 y 2^{16}, por lo que la transformación Tent discretizada pasa de la forma como la vemos en la ecuación 17, a la forma:

$$f(x)=\begin{cases} floor\left(2x\mu+\left(2^n\left(\frac{1-\mu}{2}\right)\right)\right), & 0\le x_n<2^{n-1} \\ floor\left(-\mu\left(2\left(x-2^n\right)\right)+\left(2^n\left(\frac{1-\mu}{2}\right)\right)\right) & 2^{n-1}\le x_n\le 2^n. \end{cases} \qquad \textbf{(23)}$$

donde *n* es el número de bits usado para representar cada símbolo del alfabeto considerado. En la figura 3.11, se puede observar el comportamiento de la familia de la transformación caótica Tent, cuando se escala y discretiza a un valor de 2^8.

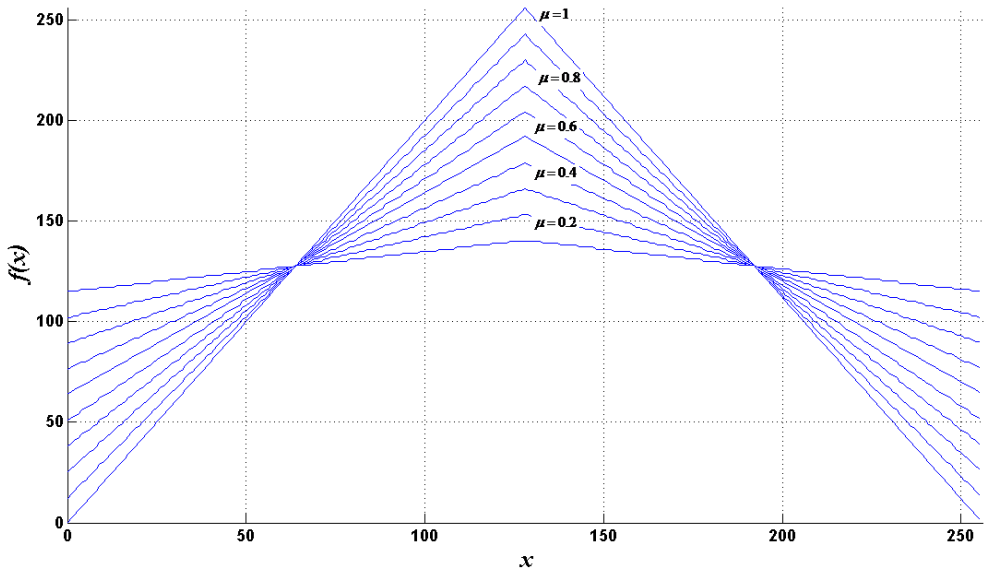


Figura 3.11 Familia de la Transformación Caótica Tent, escalada y discretizada a 2^8. Para parámetros desde $\mu = 0.1$ hasta $\mu = 1$.

3.4 Caracterización de la Transformación caótica Tent, escalada y discretizada

Diagrama de Trayectorias

Como ya se explicó en la sección 3.2, en la figura 3.12 se puede observar que para iterar la función se necesita un valor inicial y el resultado será la siguiente entrada de la función. Más sin embargo, como se puede observar, el sistema tiende a comportarse de una manera estable, por lo que no es de interés, ya que se busca el caos total. En la figura 3.13, se observa que se acerca al caos al variar el parámetro μ que nos rige la transformación. Finalmente, se utiliza la μ máxima, para tratar de llegar al caos, más sin embargo, como se puede observar en la figura 3.14, el caos no es total. Con la $\mu = 0.9852$, es donde se obtiene el mayor caos cuando se escala y discretiza la función Tent al campo de los enteros [0,255], y sin embargo, no es suficiente, incluso no sirve para utilizarlo con fines criptográficos.

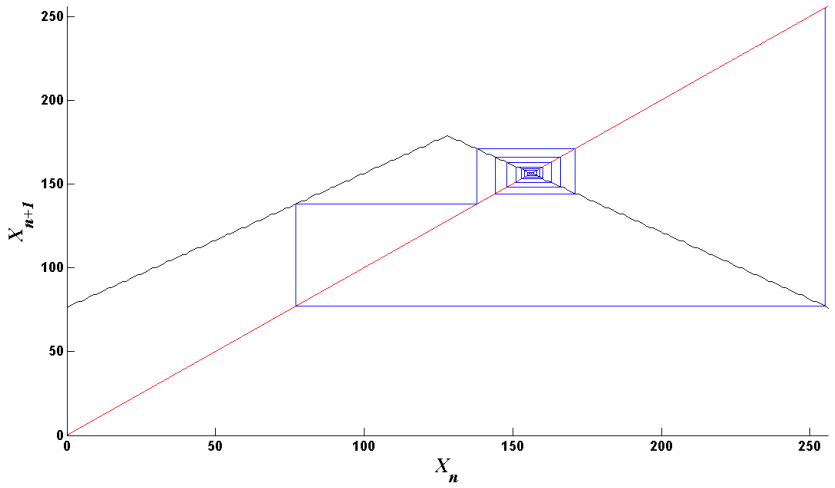


Figura 3.12 Función escalada y discretizada, iterada con $\mu = 0.4$, $x_0 = 255$ ***y 1000 iteraciones.***

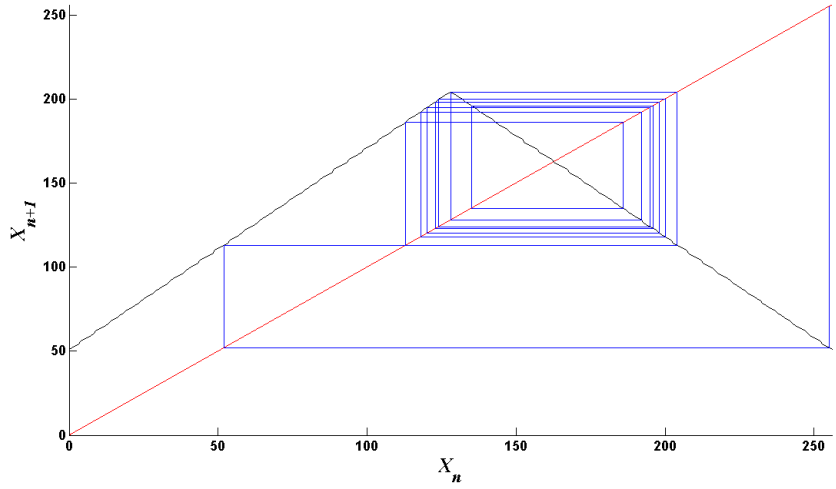


Figura 3.13 Función escalada y discretizada, iterada con $\mu = 0.6$, $x_0 = 255$ ***y 1000 iteraciones.***

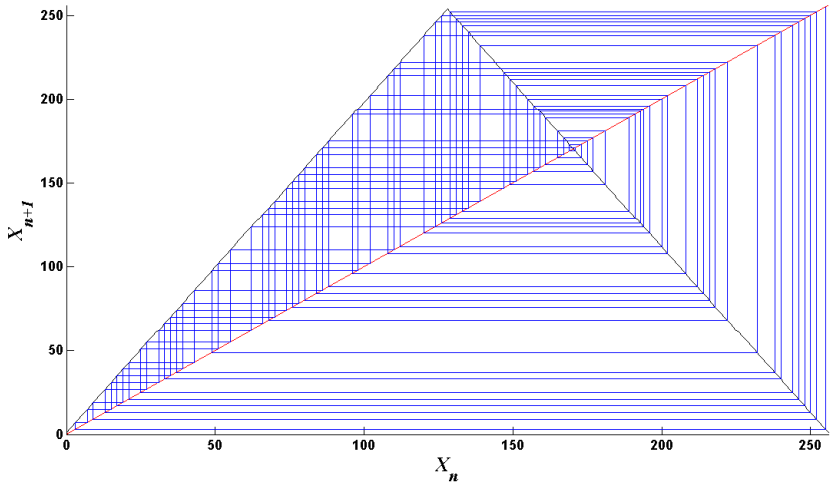


Figura 3.14 Función escalada y discretizada, iterada con $\mu = 0.9852$, $x_0 = 255$ ***y 1000 iteraciones.***

Diagrama de Bifurcación

Como se vio en la sección 3.2, el diagrama de bifurcación es una herramienta que muestra las posibles órbitas a las que tiende la transformación, como una función del parámetro que lo gobierna, y permite visualizar dónde existe el comportamiento periódico y dónde existe el comportamiento caótico. Se puede ver el resultado de este procedimiento para valores del parámetro de $\mu \in (0,1)$ en la figura 3.15. Como se observa claramente, a partir de la $\mu > 0.72$ la transformación entrará el fenómeno de doblamiento del periodo, y comprobando el diagrama de trayectorias que se ve en la figura 3.14, se puede notar que el diagrama no es denso, como es requerido.

Distribución Estadística

Como puede observarse del diagrama de bifurcación de la transformación Tent escalada y discretizada, la distribución invariante depende del valor del parámetro. Así, para $\mu < 0.4$, la distribución invariante estará compuesta por una delta de Dirac; en tanto que, para valores de $\mu \in (0.52,0.58)$ se presentarán tres deltas de Dirac. Finalmente, para valores de $\mu > 0.999$ la distribución cubre un solo intervalo. Lo anterior se puede observar en las figuras 3.16 a 3.18 donde se muestra que 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.

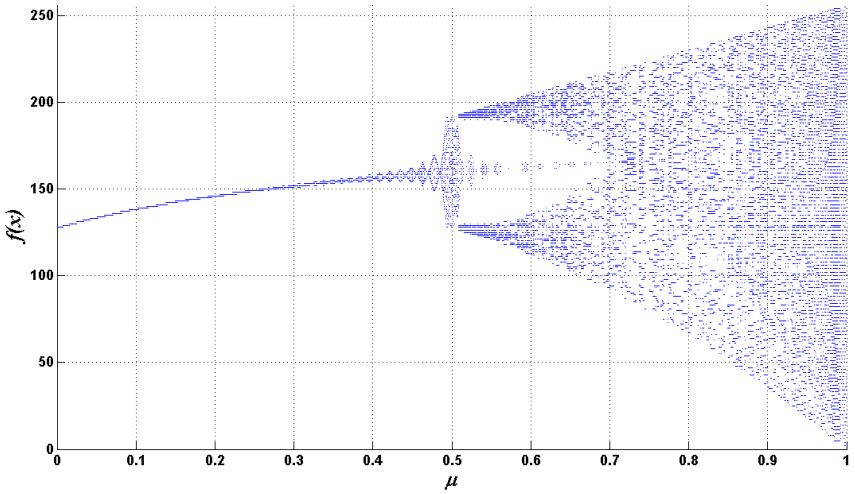
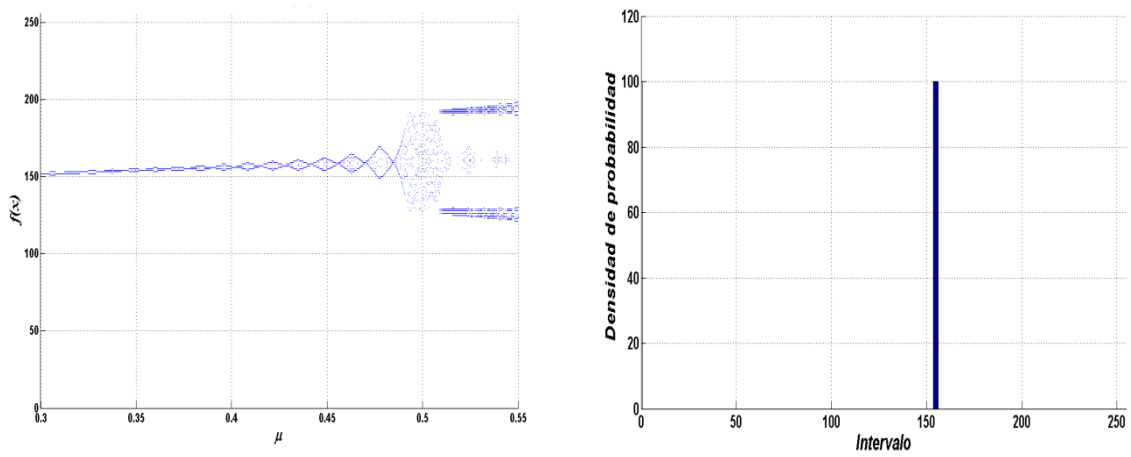
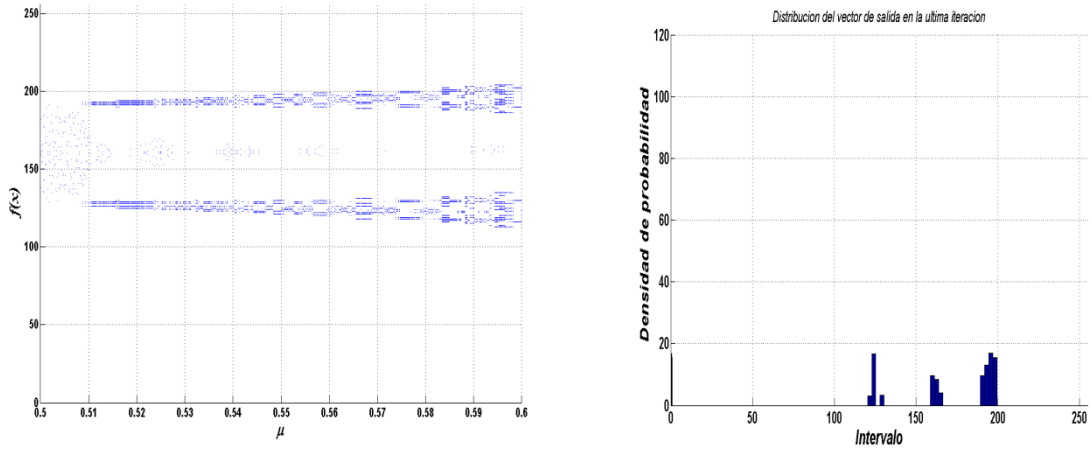


Figura 3.15 Diagrama de bifurcación, escalado y discretizado, de la Transformación Caótica unidimensional de la Tent.



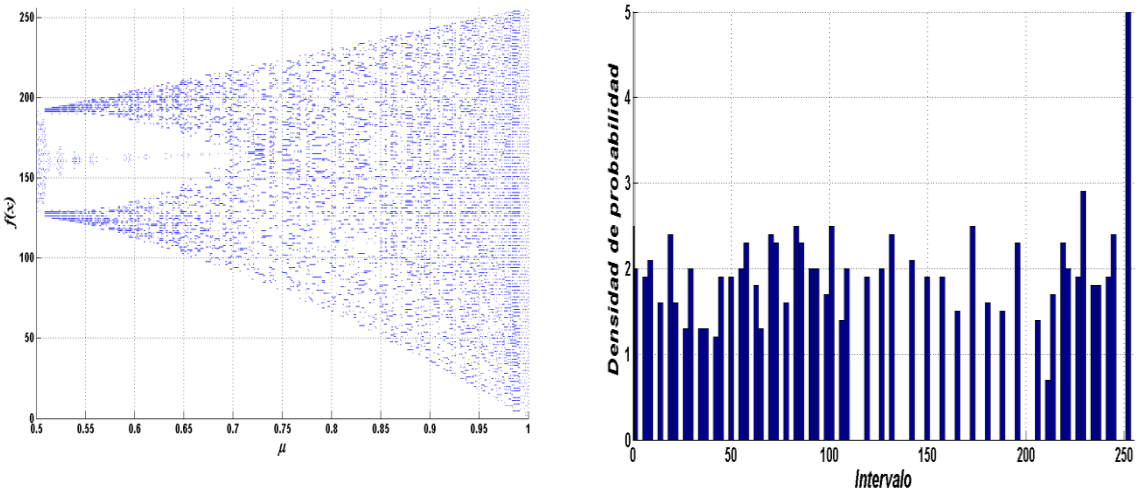
a) **b)**

Figura 3.16 Transformación Tent para $\mu = 0.4$***. a) Diagrama de bifurcación; b)Densidad de Probabilidad.***



a) ***b)***

Figura 3.17 Transformación Tent para $\mu = 0.56$***. a) Diagrama de bifurcación; b) Densidad de Probabilidad.***



a) ***b)***

Figura 3.18 Transformación Tent para $\mu = 0.999$***. a) Diagrama de bifurcación b) Densidad de Probabilidad.***

Exponente de Lyapunov

En la figura 3.19 se presume la forma de interpretar el exponente de Lyapunov sobre la transformación Tent, éste 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 de la transformación, esto es el caos. Para la transformación Tent se observa que a partir del valor de $\mu = 0.499999$, el comportamiento se vuelve positivo.

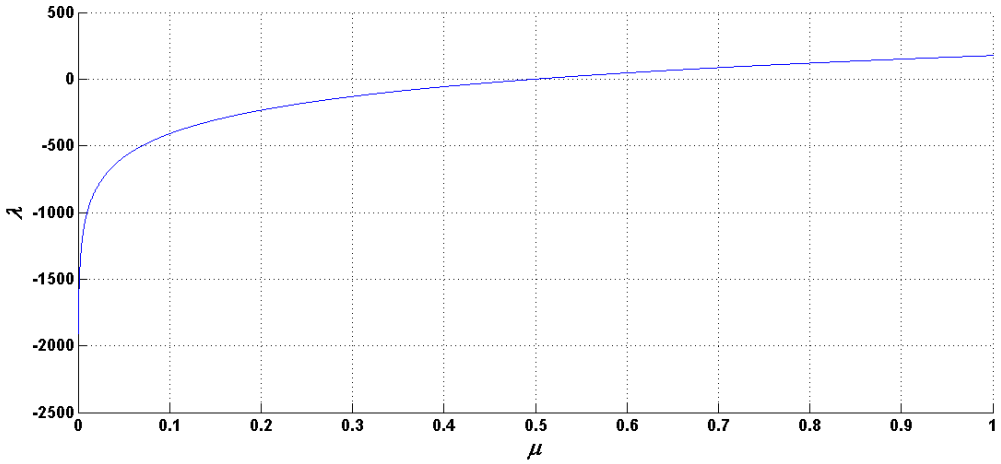


Figura 3.19 Exponente de Lyapunov para la Transformación Tent escalada y discretizada.

Comentarios al capítulo

En este capítulo se describen los conceptos de la Transformación Caótica Unidimensional en general, y de la Tent en particular. Aplicando estos conceptos, se muestran los Diagramas de Trayectorias y de Bifurcación, así como la Distribución estadística y el Exponente de Lyapunov de la Transformación Caótica Unidimensional de la Tent; esto es en los campos de los reales, y en la versión escalada y discretizada en el dominio ASCII Extendido.

CAPÍTULO 4

CIFRADO DE BLOQUES CAÓTICO

RESUMEN

En este capítulo se describe el algoritmo de cifrado propuesto por Kocarev, así como también el algoritmo propuesto en esta tesis en sus dos versiones. Al final de este capítulo se dan a conocer los resultados de esta implementación comparándolos con los obtenidos con las transformaciones caóticas logística y senoidal; y otros cifradores comerciales.

4.1 Descripción del algoritmo propuesto por Kocarev

En la literatura, se han estudiado tres tipos de funciones de transformación: Redes de Feistel [61],[61] redes de Feistel desbalanceadas, siendo los ejemplos más comunes los algoritmos MacGuffin [62], BEAR/LION [63] y Redes de sustitución-permutación (SPNetworks), también llamadas estructuras de transformación uniformes, como por ejemplo, IDEA [64] y SAFER [65]. Ljupco Kocarev es un científico asociado a la investigación en el Institute of Nonlinear Science de la Universidad de California, San Diego y un profesor en la Graduate School of Engineering en Skopje, Macedonia. Su investigación e intereses profesionales incluyen la ciencia no lineal y su aplicación, la comunicación de caos, la teoría de la información y la criptografía. Ha publicado en revistas más de 17 trabajos diferentes. Según la SCI (*Science Citation Index*), su trabajo ha sido citado más de 1,500 veces. Para esta tesis, se trabaja con los algoritmos de cifrado por bloque. Primero se describe brevemente la propuesta de Ljupco Kocarev y Goce Jakimoski en [5] que fue publicado en el año 2001. La mayoría de los algoritmos de cifrado tienen la forma

$$\begin{aligned} &x_0 = B_0. \\ &x_1 = E_z\left[x_{i-1}\right], \qquad i = 1,\ldots,r \\ &B_r = x_r. \end{aligned} \tag{24}$$

Donde B_0, B_r representan el bloque del mensaje original y el bloque del criptograma, 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. Sea un bloque del mensaje original 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 del mensaje original. La función de cifrado está dada por

$$x_{i,k+1} = x_{i-1,k} \oplus f_{k-1}\left(x_{i-1,1},\ldots,x_{i-1,k-1},z_{i-1,k-1}\right) \tag{25}$$

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 subclave , la cual controla la i-enésima ronda, véase la figura 4.1. Las funciones $f_0, \ldots, f_7$ tienen la siguiente forma:

$$f_j = \left(x_{1,\ldots,} x_j, z_j\right) \quad \textbf{(26)}$$

donde $j = 1,\ldots,7$ y $f : M \to M, M = \{0,\ldots,255\}$ es una transformación derivada de una transformación caótica. 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 del criptograma. La longitud del criptograma es de 64 bits (8 bytes) y es igual a la longitud del bloque del mensaje original. Cada ronda i es controlada por una subclave de 8 bytes. Hay r subclaves en total y se derivan de la clave en un procedimiento para la generación de rondas.

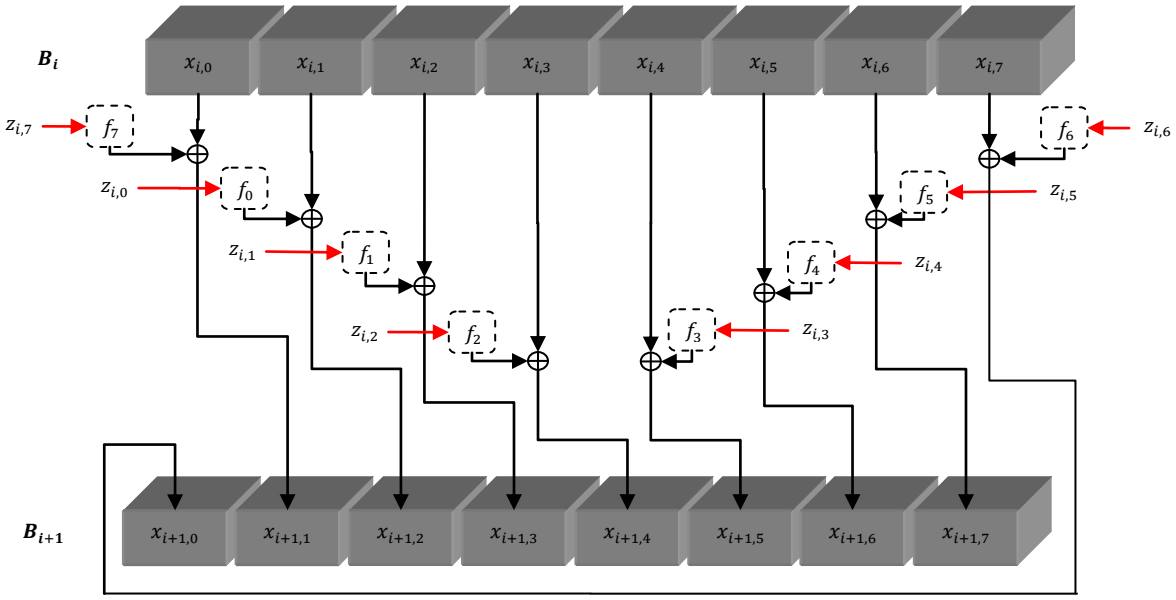


Figura 4.1 Esquema general de cifrado propuesto por Kocarev.

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

$$x_{i-1,k} = x_{i,k+1} \oplus f_{k-1}\left(x_{i-1,1},\ldots,x_{i-1,k-1}, z_{i-1,k-1}\right) \quad \textbf{(27)}$$

donde $i=1, \ldots, r, k=1,\ldots, 8, f_0=zi_{,0}, x_8 \equiv x_0, x_9 \equiv x_1$.

4.2 Propuesta de modificación al algoritmo de Kocarev

La propuesta de trabajo de tesis está basada en la propuesta de Kocarev, con algunas modificaciones significativas, ya que como se puede ver en la figura 4.2, por cada bloque cifrado se utilizan 8 rondas, y cada una de éstas se cifra con una clave diferente, derivada de la clave principal, para obtener el bloque cifrado. Al finalizar, se cifra la propia clave principal y el número de rondas con que se realizó el cifrado y estos resultados se concatenan al resultado obtenido del archivo cifrado; lo anterior con la finalidad de que en el proceso de descifrado, se realice una validación de la clave y del número de rondas con que se cifró el archivo, antes de proceder a realizar el descifrado, y en caso de que alguno de los dos no coincida, se aborta el proceso de descifrado, evitando con esto que se realice el proceso de descifrado que aunque no arroja la información que se cifró, si se realizara un análisis se podría llegar a obtener el archivo en claro.

Proceso de cifrado

Antes de iniciar la descripción del proceso de cifrado, cabe mencionar que al comienzo de la investigación que dio paso a este trabajo de tesis, se empezó a trabajar con 8 bloques de 8 bits cada uno, para hacer un total de 64 bits por bloque de cifrado; sin embargo, cuando ya se tenían los resultados correspondientes y se estaban realizando las pruebas de evaluación del cifrador, se pudo constatar que el diagrama de bifurcación no era completamente denso, ver figura 3.15, por lo que se decidió realizar un cifrador alterno, que mantuviera los 64 bits por bloque de cifrado, solo que con 4 bloques de 16 bits cada uno. En adelante se hará referencia a ellos como Cifrador de 8 bloques y Cifrador de 4 bloques, respectivamente.

Cifrador de 8 bloques

Este proceso se basa en la trabajo de Goce Jakimoski y Ljupco Kocarev [5] la propuesta para esta tesis es la siguiente. Se toma un bloque de B_0 del mensaje original, dicho bloque consta de 64 bits de longitud dada por:

$$B_i = \left\langle x_{i,0}, x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}, x_{i,5}, x_{i,6}, x_{i,7} \right\rangle \quad \textbf{(28)}$$

De modo que las $x_{i,j}$ con $j = 0,\ldots,7,$ constituyen los 64 bits que conforman el bloque B_0. Cada $x_{i,j}$ contiene 8 bits.

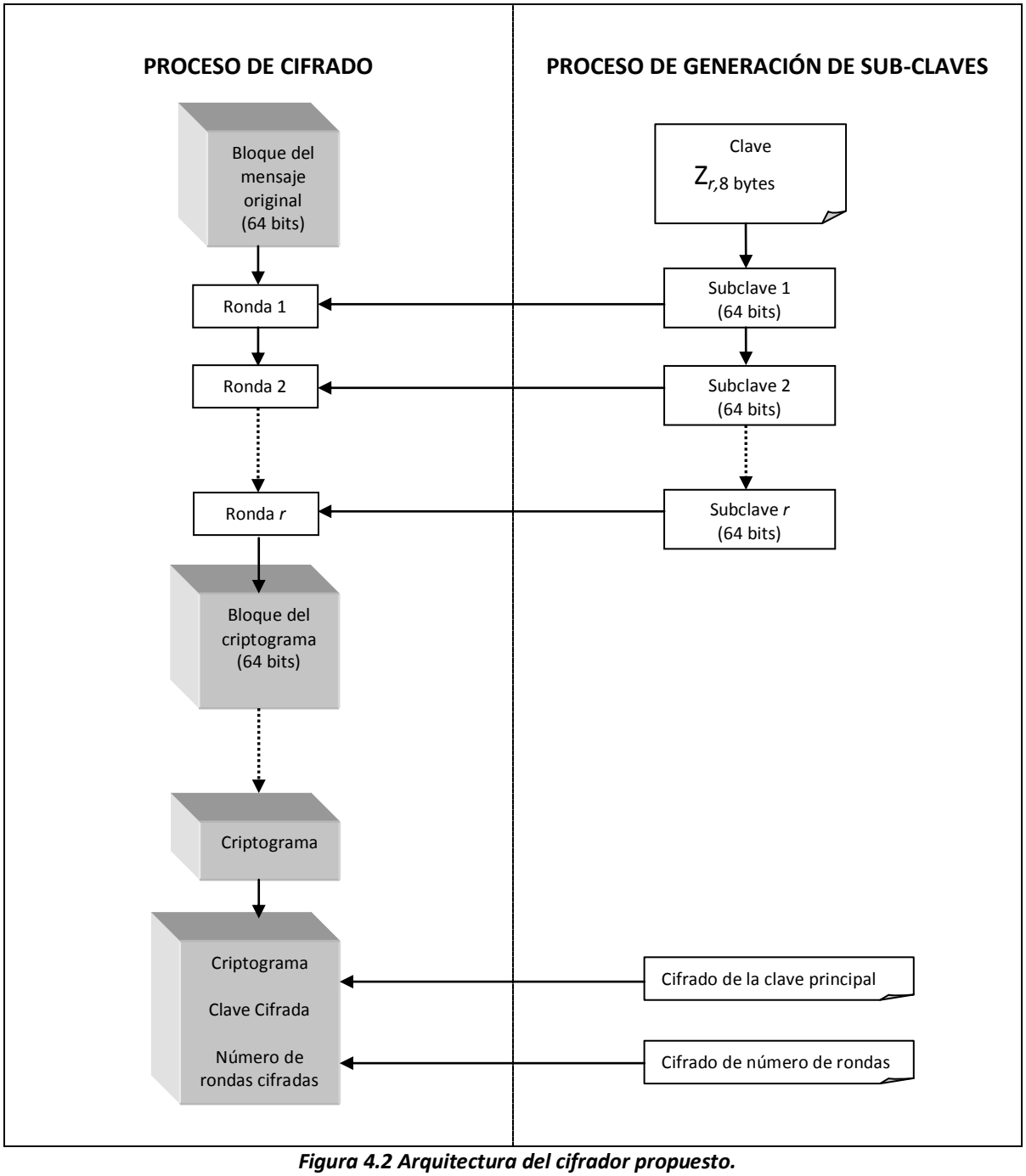


Figura 4.2 Arquitectura del cifrador propuesto.

Para este trabajo el proceso de cifrado se compone de 8 rondas, *r*=8. Cada ronda se compone de una red de Feistel desbalanceada que utiliza una subclave de 64 bits. La ronda se puede definir como un conjunto de funciones de transformación como sigue:

$$
\begin{aligned}
x_{i+1,2} &= x_{i,1} \oplus f_0\left(z_{i-1,0}\right),\\
x_{i+1,3} &= x_{i,2} \oplus f_1\left(x_{i,1} \oplus z_{i-1,1}\right),\\
x_{i+1,4} &= x_{i,3} \oplus f_2\left(x_{i,1} \oplus x_{i,2} \oplus z_{i-1,2}\right),\\
x_{i+1,5} &= x_{i,4} \oplus f_3\left(x_{i,1} \oplus x_{i,2} \oplus x_{i,3} \oplus z_{i-1,3}\right),\\
x_{i+1,6} &= x_{i,5} \oplus f_4\left(x_{i,1} \oplus x_{i,2} \oplus x_{i,3} \oplus x_{i,4} \oplus z_{i-1,4}\right),\\
x_{i+1,7} &= x_{i,6} \oplus f_5\left(x_{i,1} \oplus x_{i,2} \oplus x_{i,3} \oplus x_{i,4} \oplus x_{i,5} \oplus z_{i-1,5}\right),\\
x_{i+1,0} &= x_{i,7} \oplus f_6\left(x_{i,1} \oplus x_{i,2} \oplus x_{i,3} \oplus x_{i,4} \oplus x_{i,5} \oplus x_{i,6} \oplus z_{i-1,6}\right),\\
x_{i+1,1} &= x_{i,8} \oplus f_7\left(x_{i,1} \oplus x_{i,2} \oplus x_{i,3} \oplus x_{i,4} \oplus x_{i,5} \oplus x_{i,6} \oplus x_{i,7} \oplus z_{i-1,7}\right).
\end{aligned}
\qquad \textbf{(29)}
$$

Los valores $x_{i+1,1}, x_{i+1,2}, \ldots, x_{i+1,j}$, representan los bits del bloque del criptograma. Las f_j suman (suma lógica) el bloque o los bloques anteriores con la subclave correspondiente y el resultado se le aplica la transformación caótica Tent, excepto la f_0, ver la figura 4.3. Esto se puede apreciar mejor en cada uno de los esquemas de cifrado representados por las figuras 4.4 a 4.10.

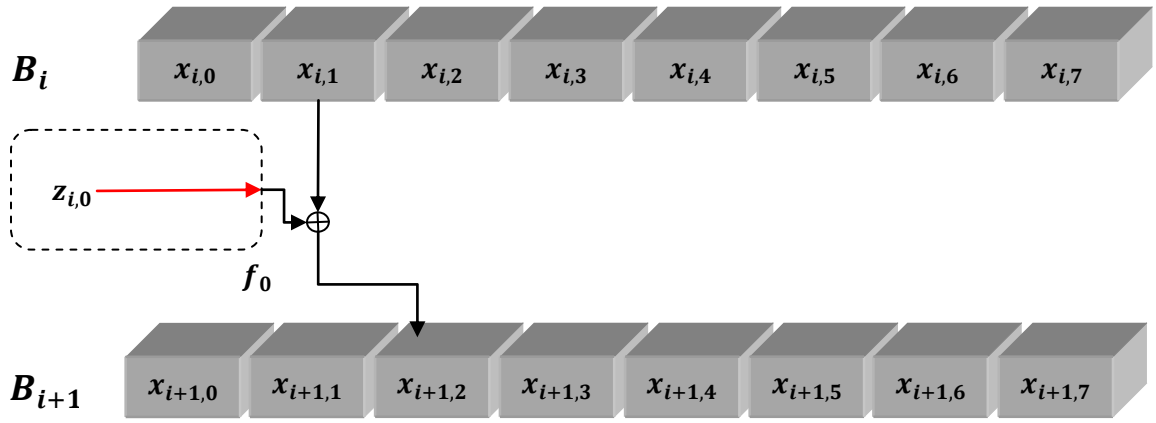


Figura 4.3 Esquema de cifrado para f_0.

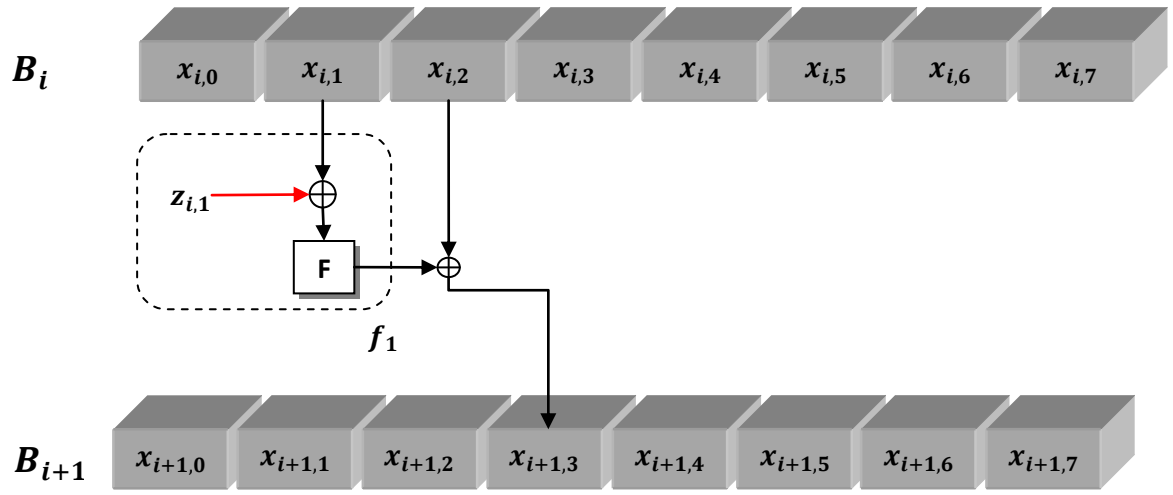


Figura 4.4 Esquema de cifrado para f_1.

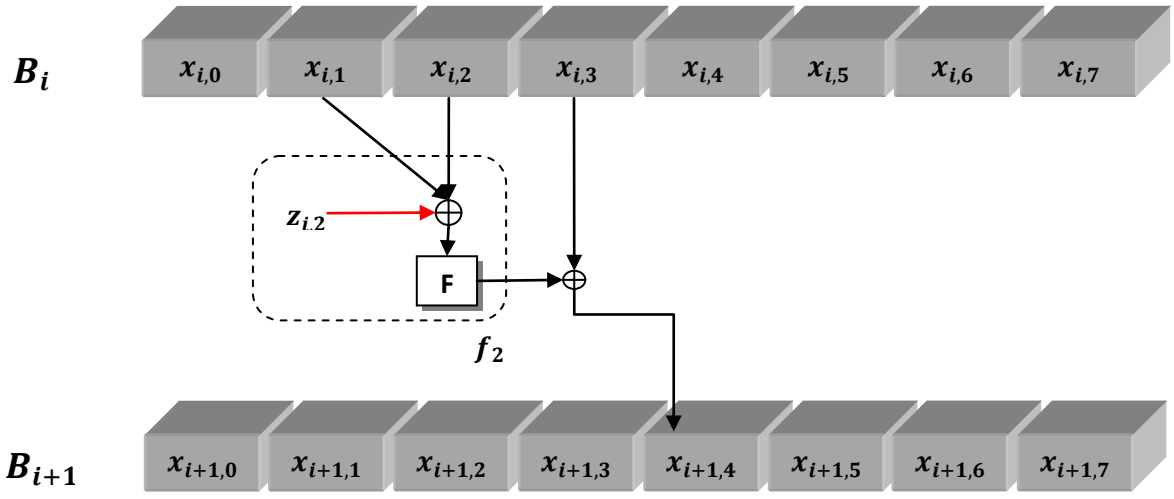


Figura 4.5 Esquema de cifrado para f_2.

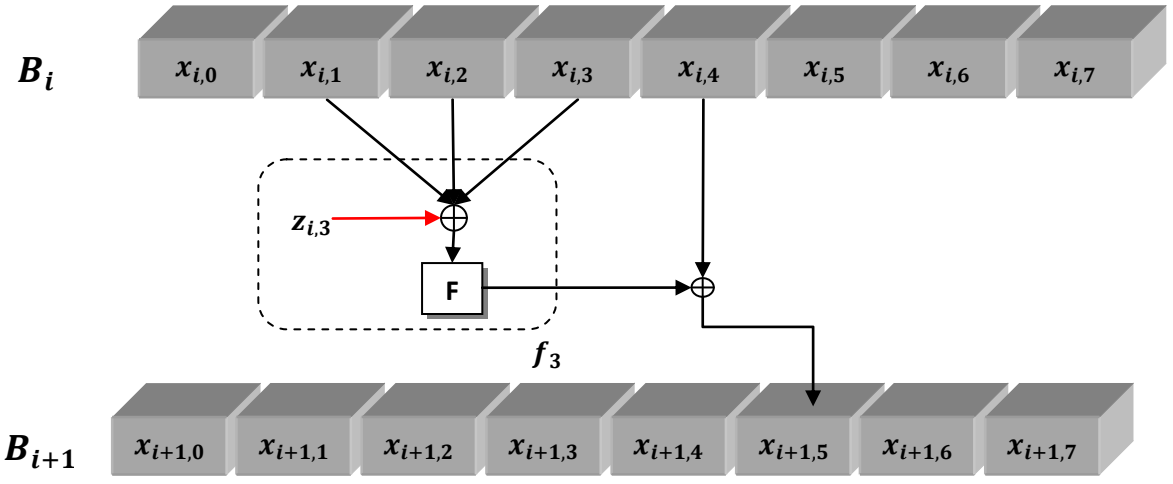


Figura 4.6 Esquema de cifrado para f_3.

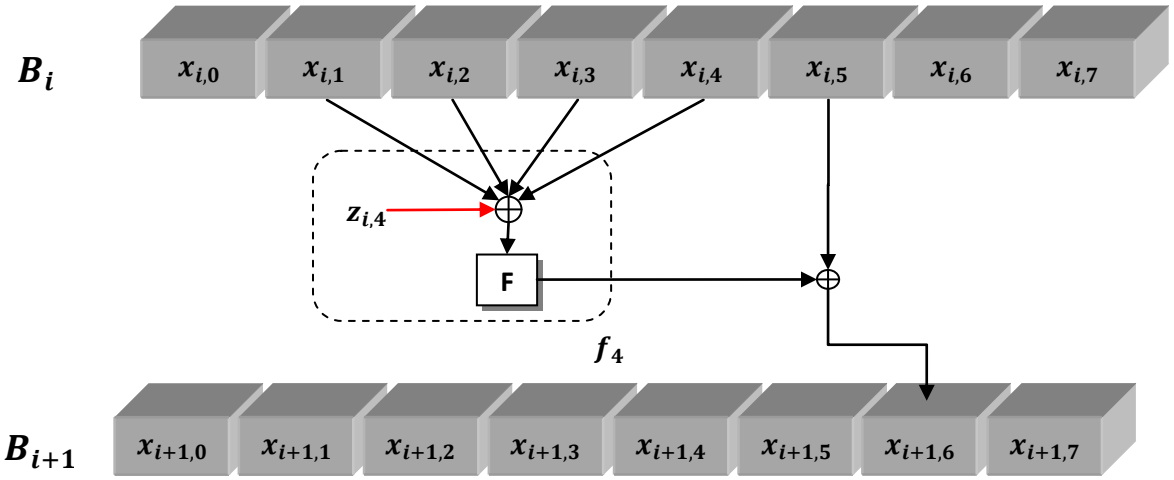


Figura 4.7 Esquema de cifrado para f_4.

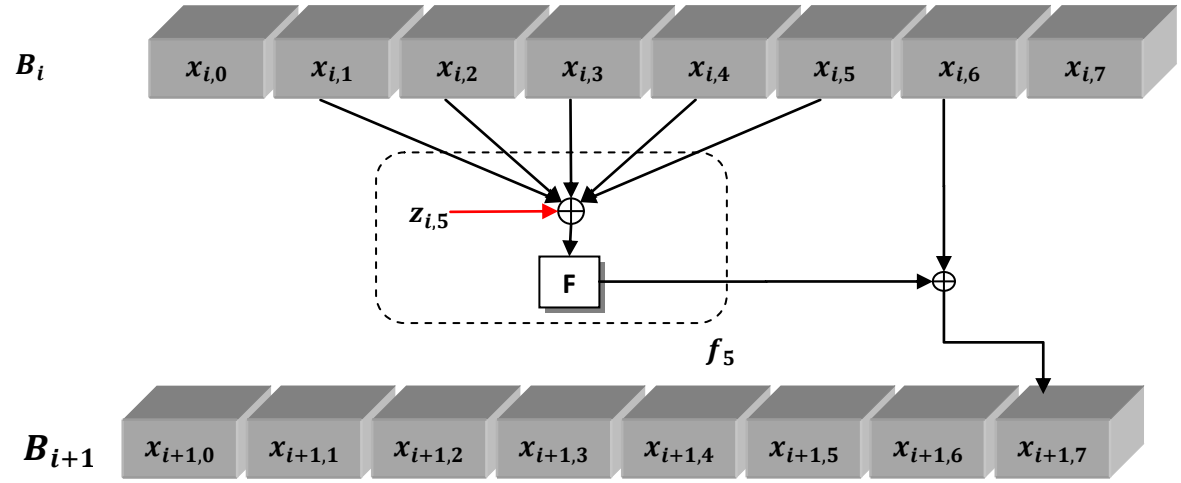


Figura 4.8 Esquema de cifrado para f_5.

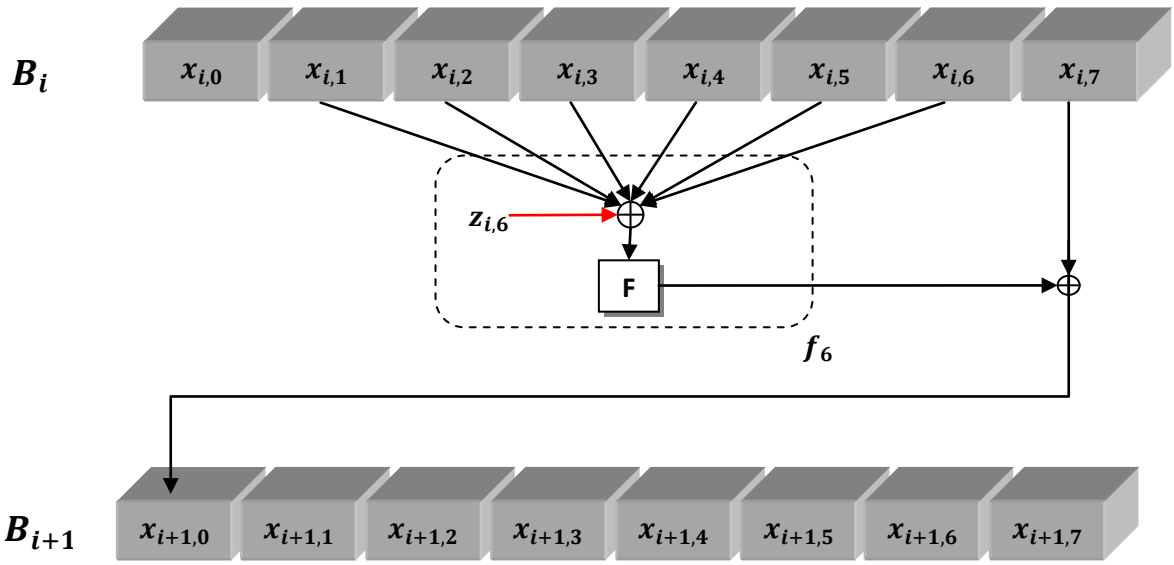


Figura 4.9 Esquema de cifrado para f_6.

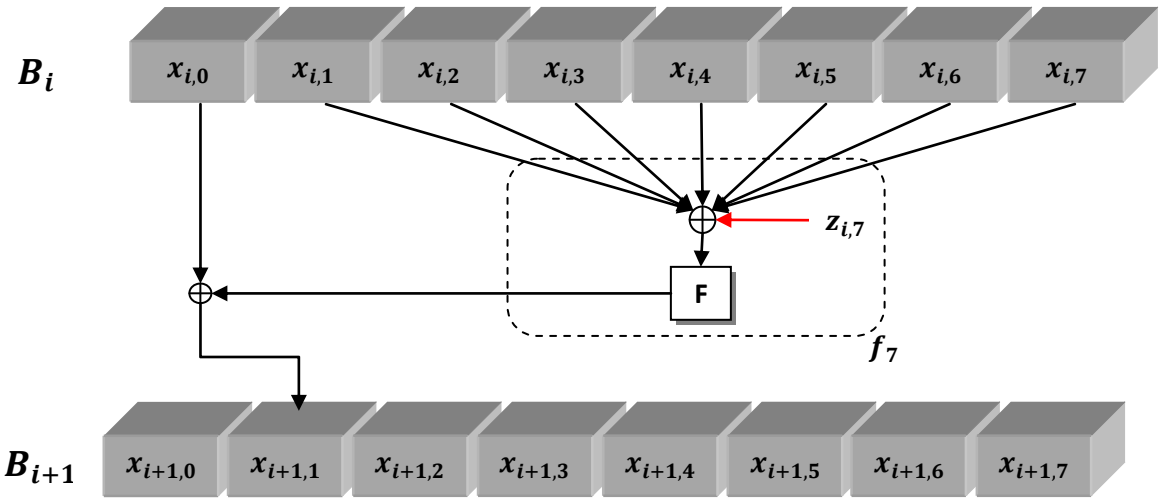
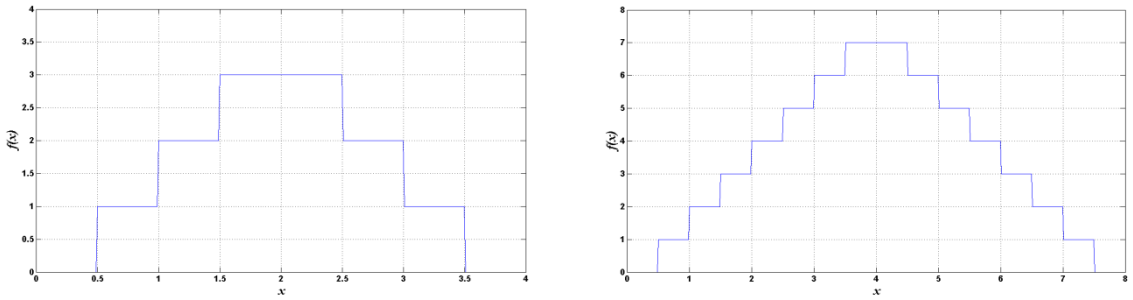


Figura 4.10 Esquema de cifrado para f_7 **.**

La función caótica utilizada para la construcción del cifrador está contenida en **F**, siendo **F** misma, la función de transformación Tent, ver ecuación (19), con $\mu \in (0,1)$ y $x \in (0,1)$. Sin embargo en este dominio el cifrador no puede trabajar, ya que para cifrar, el dominio utilizado es el del alfabeto ASCII extendido, el cual se encuentra definido en el intervalo [0, 255]. Por lo tanto, la transformación Tent se tiene que escalar y discretizar del intervalo (0, 1) en los números reales, al intervalo (0, 255) en los números enteros, de acuerdo con lo descrito en la sección 3.3, por lo que la transformación Tent escalada y discretizada queda como la ecuación 23, con *n=8*.

La curva de la transformación queda escalada y discretizada en el intervalo [0,255], que es el dominio de definición del alfabeto en el que se representan los contenidos de cualquier archivo; esto es, el universo de los caracteres ASCII. Es importante mencionar que el escalamiento y la discretización en el intervalo [0,255] ofrece un adecuado comportamiento, como se ve en las distintas gráficas de la figura 4.11, ya que cuando se usan 8 bits la Tent resultante es una buena aproximación a la Tent no discretizada, los rizos de la Tent son despreciables.



a) ***b)***

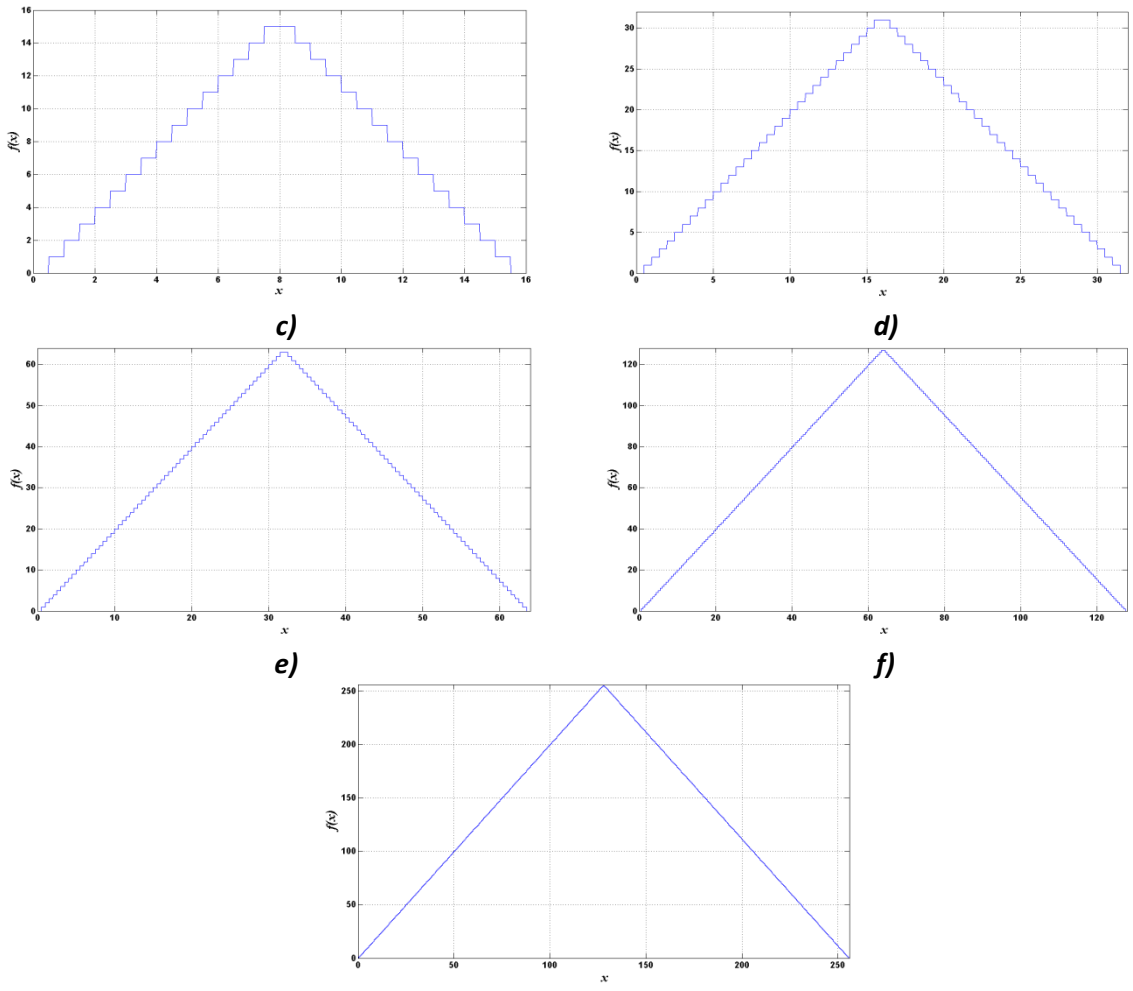


Figura 4.11 Curva de la transformación Tent escalada y discretizada, con valores de μ=1.0 y 100 puntos usando distintos valores de n. a) 2^2 bits, b) 2^3 bits, c) 2^4 bits, d) 2^5 bits, e) 2^6 bits, f) 2^7 bits, g) 2^8 bits.

Sin embargo, desde el punto de vista del universo de 256 elementos que constituyen el alfabeto en el que están escritos los archivos, se aprecia una desventaja notoria. El diagrama de bifurcación, para cuando se usan 8 bits, no resulta denso, lo cual repercute en la distribución estadística de la señal generada usando la transformación Tent. Nótese que en cada caso la altura máxima de la parábola está definida por el parámetro μ. El procedimiento de descifrado es similar: se aplican 8 rondas de descifrado sobre el bloque del criptograma B_j para así producir el bloque del mensaje original B_{j-1}. Las rondas de las subclaves son aplicadas ahora en forma inversa. El proceso de descifrado se muestra en la figura 4.12.

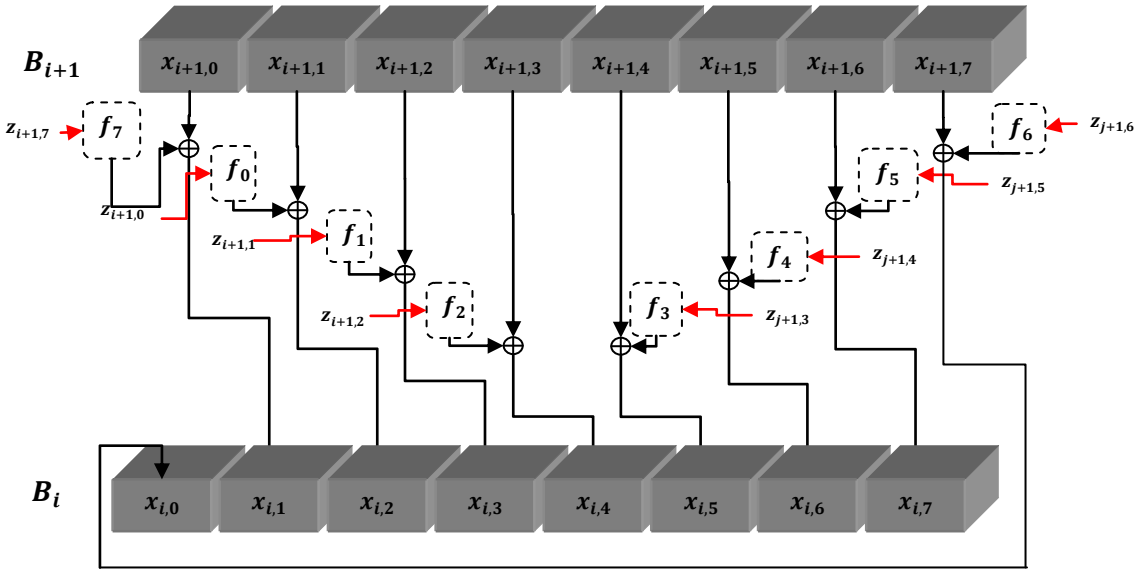


Figura 4.12 Esquema de descifrado

Proceso de generación de subclaves

El proceso de generación de subclaves nace de la necesidad de quitar un problema al criptosistema, la **redundancia**. Imagine un bloque de mensaje original *B* el cual se cifra usando una clave *k*, hay una gran posibilidad de que se vuelva a cifrar un bloque del mensaje original igual a *B*, como se usa la misma clave k, genera el mismo bloque del criptograma. Por lo tanto, esto crea redundancia al criptograma el cual puede dar una herramienta eficaz para su análisis. Pensando en este problema se estructuró un proceso para generar subclaves para cada bloque del mensaje original a cifrar. Primero se necesita un proceso que genere subclaves diferentes para cada bloque a cifrar, pero que dependan de una clave principal. El proceso de generación de subclaves provee a cada ronda una subclave de una longitud de 64 bits. Estas subclaves se derivan de una clave principal que el usuario provee antes de comenzar a cifrar el archivo. Para cada bloque del mensaje original se necesitan 8 subclaves de 64 bits, en total para poder cifrar un bloque del mensaje original se necesita 512 bits, los cuales tienen que ser distintos para cada bloque[6].

Pruebas de funcionalidad del cifrador de 8 bloques

El proceso de cifrado tiene como objetivo afectar tres de las propiedades fundamentales del lenguaje natural. Estas propiedades son las sintáctica, la semántica y la estadística del lenguaje. Es

[6] Este proceso de generación de subclaves puede verse completamente en [60]

importante mencionar que en el lenguaje natural está implícito un grado de redundancia o repetición de caracteres. La frecuencia con la que cada carácter se repite en un lenguaje crea una cierta huella estadística. Los archivos digitales también tienen una huella estadística, la cual cambia dependiendo del tipo de archivo. Cuando un mensaje es cifrado se espera que su huella estadística sea lo más parecido a una señal de ruido, o sea, que la frecuencia de cada uno de los caracteres del alfabeto usado, sea igual. Para demostrar lo anterior se cifraron diferentes archivos; texto, audio, imágenes, etc., en diferentes formatos. Se seleccionan diferentes tipos de archivos para demostrar que sin importar como sea su huella estadística al ser cifrados tendrán una huella estadística parecida a una señal de ruido. Se calculan el histograma de los archivos originales y de los archivos cifrados y se comparan a fin de poder apreciar el grado de dispersión efectuado sobre los archivos cifrados con el proceso de transformación. En la Tabla 1 se observan los tipos de archivos usados para probar el criptosistema. La primera prueba de funcionalidad es probar que en el proceso de cifrado y descifrado el criptograma descifrado no pierda la información del mensaje original. Esta propiedad la debe de cumplir cada criptosistema y para verificarla, se trabajan con 19 archivos de referencia de diferentes tipos, los cuales se cifran y descifran y se verifica su tamaño.

Este análisis, aunque indica que el archivo que se descifró conserva el mismo tamaño que el archivo original, no dice qué tan íntegra queda la información después de haber pasado por el criptosistema. Para probar esto, se utiliza una función HASH, en este caso la MD5. Se compara el resumen de los archivos originales y los descifrados. En la Tabla 2, se observan los resúmenes de cada archivo original y se comprueba que todos los archivos después de descifrarlos mantienen su integridad. En las gráficas de la figura 4.13 se puede apreciar, desde un punto de vista estadístico, el comportamiento de los archivos de originales. Las figuras del lado izquierdo corresponden a los archivos originales. Las figuras del lado derecho corresponden a los archivos cifrados con el cifrador de 8 bloques propuesto. Las gráficas de los archivos originales muestran huellas estadísticas particulares para cada tipo de archivo. Obsérvese que para archivos que tienen un proceso de compresión, como el MP3, sus histogramas de origen se parecen a una señal de ruido. Los procesos de compresión quitan redundancia, por eso los histogramas de estos archivos son casi uniformes.

Tabla I Tamaño de los archivos de referencia.

Tipos de archivos	Tamaño del archivo original	Tamaño del archivo cifrado	Tamaño del archivo descifrado
TXT	213 KB	213 KB	213 KB
DOC	356 KB	356 KB	356 KB
RTF	3,124 KB	3,124 KB	3,124 KB
XML	997 KB	997 KB	997 KB
PPT	477 KB	477 KB	477 KB
RTFPDF	232 KB	232 KB	232 KB
XMLPDF	232 KB	232 KB	232 KB
DOCPDF	233 KB	233 KB	233 KB
PPTPDF	372 KB	372 KB	372 KB
BMP	2,236 KB	2,236 KB	2,236 KB
JPG	65 KB	65 KB	65 KB
PNG	1,038 KB	1,038 KB	1,038 KB
TIF	1,792 KB	1,792 KB	1,792 KB
MP3	3,346 KB	3,346 KB	3,346 KB
MP3-64	409 KB	409 KB	409 KB
MP3-128	816 KB	816 KB	816 KB
MP3-256	1,630 KB	1,630 KB	1,630 KB
WAV	820 KB	820 KB	820 KB
WMV	3,937 KB	3,937 KB	3,937 KB

Todas las gráficas de los archivos cifrados presentan señales semejantes, a tal grado que son casi uniformes. Esto quiere decir que el cifrador de 8 bloques propuesto cambia exitosamente las propiedades de la semántica, la sintáctica y la estadística de los archivos.

Cifrador de 4 bloques

Este proceso también se basa en la trabajo de Goce Jakimoski y Ljupco Kocarev [5], solo que en esta ocasión se utilizaron 4 bloques para el proceso de cifrado, con 16 bits cada uno, para mantener el bloque total de 64 bits de longitud, como se puede observar en la figura 4.14. Se toma un bloque de B_0 del mensaje original, dicho bloque consta de 64 bits de longitud dada por

$$B_i = \langle x_{i,0}, x_{i,1}, x_{i,2}, x_{i,3} \rangle \quad \textbf{(30)}$$

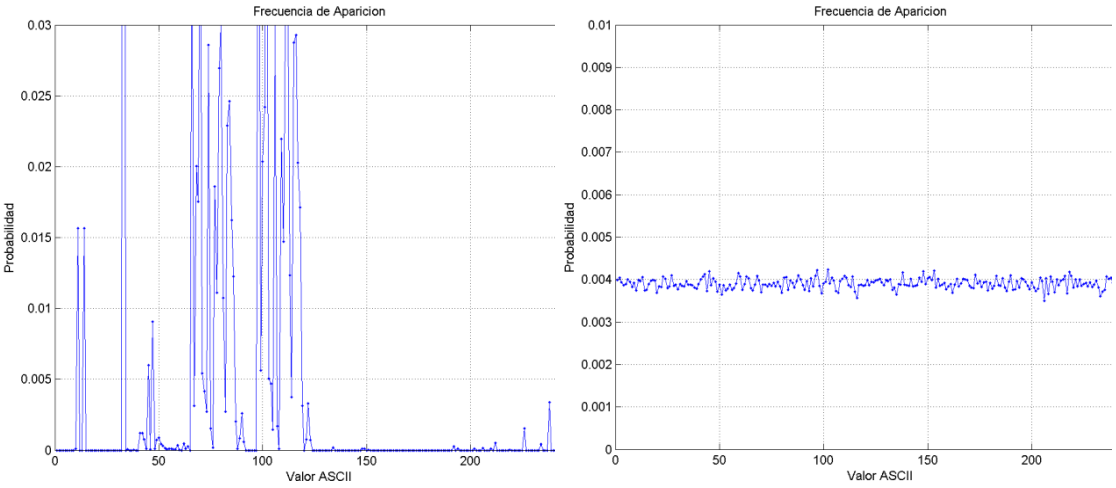
De modo que las $x_{i,j}$ con $j = 0,\ldots,3,$ constituyen los 64 bits que conforman el bloque B_0. Cada $x_{i,j}$ contiene 16 bits.

Tabla II Comparación del Resumen de los archivos.

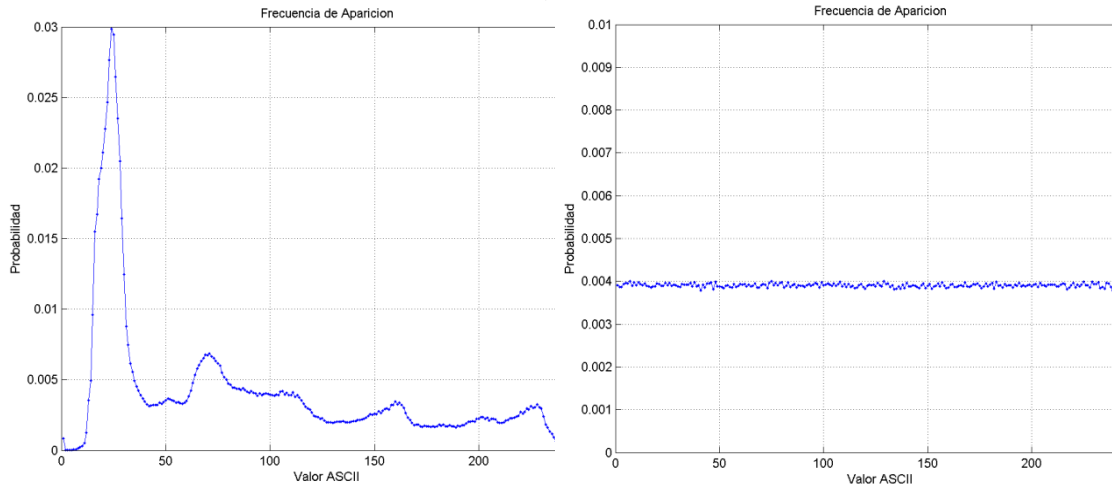
Tipo de archivo		Resumen MD5	Idénticos
TXT	Original	620d9e70dde7d67fc982924ecbe0c582	Sí
	Descifrado	620d9e70dde7d67fc982924ecbe0c582	
DOC	Original	4d6458c1e381a62651d57ef21f2a20b3	Sí
	Descifrado	4d6458c1e381a62651d57ef21f2a20b3	
RTF	Original	761f722320c5d21b5dafaa15835a746b	Sí
	Descifrado	761f722320c5d21b5dafaa15835a746b	
XML	Original	f71005c0610ec2e590be661c5584d623	Sí
	Descifrado	f71005c0610ec2e590be661c5584d623	
PPT	Original	2bddc62e67a5a3bb83a8678937ee1e5a	Sí
	Descifrado	2bddc62e67a5a3bb83a8678937ee1e5a	
RTFPDF	Original	ef0c169ede9fee819597b612f70749e7	Sí
	Descifrado	ef0c169ede9fee819597b612f70749e7	
XMLPDF	Original	4b31bb81187387f7e4fc5a33e063e73e	Sí
	Descifrado	4b31bb81187387f7e4fc5a33e063e73e	
DOCPDF	Original	9a7c17dc2924e0b584ed5d17994aefd0	Sí
	Descifrado	9a7c17dc2924e0b584ed5d17994aefd0	
PPTPDF	Original	f6acfd114e0f99154b0272c783f01c6f	Sí
	Descifrado	f6acfd114e0f99154b0272c783f01c6f	
BMP	Original	821342ad9411d307e4f305bc74d827e5	Sí
	Descifrado	821342ad9411d307e4f305bc74d827e5	
JPG	Original	c1b0078aea7ab45ab85c4cb353d98207	Sí
	Descifrado	c1b0078aea7ab45ab85c4cb353d98207	
PNG	Original	6ab2f6edb6be8173e3c5ac3d44002113	Sí
	Descifrado	6ab2f6edb6be8173e3c5ac3d44002113	
TIF	Original	8f6bfef6c2e64aec1dcb6fa92203a7dd	Sí
	Descifrado	8f6bfef6c2e64aec1dcb6fa92203a7dd	
MP3	Original	e408f15284e648bba81eab76d8abe871	Sí
	Descifrado	e408f15284e648bba81eab76d8abe871	
MP3-64	Original	b463984e0cac5319197ac93ecd6281b2	Sí
	Descifrado	b463984e0cac5319197ac93ecd6281b2	
MP3-128	Original	77f666ca44ab117808aaac0a673abd7d	Sí
	Descifrado	77f666ca44ab117808aaac0a673abd7d	
MP3-256	Original	c8d22965fc1458bf960f2a647bb6448b	Sí
	Descifrado	c8d22965fc1458bf960f2a647bb6448b	
WAV	Original	28163a2d1f4d41b86d550e419155678a	Sí
	Descifrado	28163a2d1f4d41b86d550e419155678a	
WMV	Original	257b2e61c7d4484bddcead4adb6d834d	Sí
	Descifrado	257b2e61c7d4484bddcead4adb6d834d	

Originales

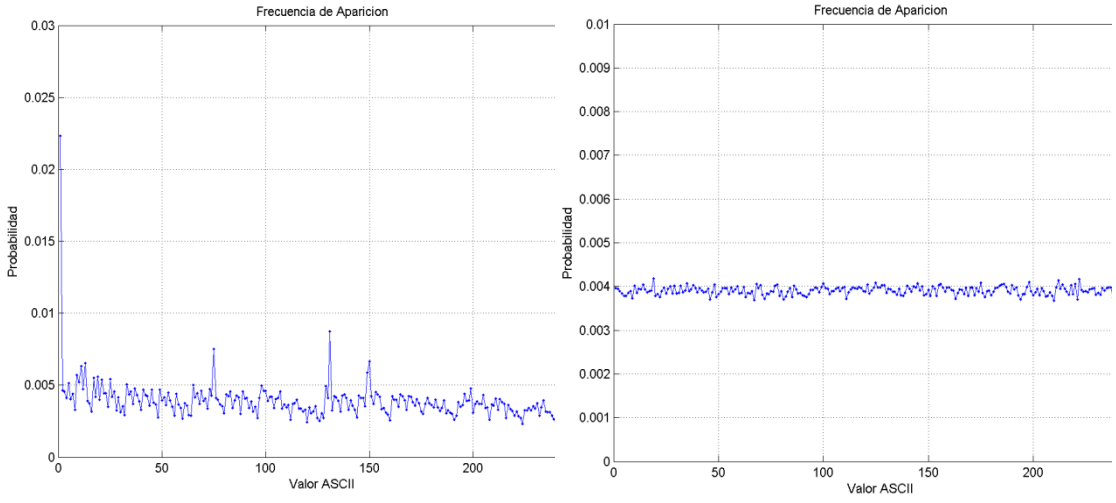
Cifrados



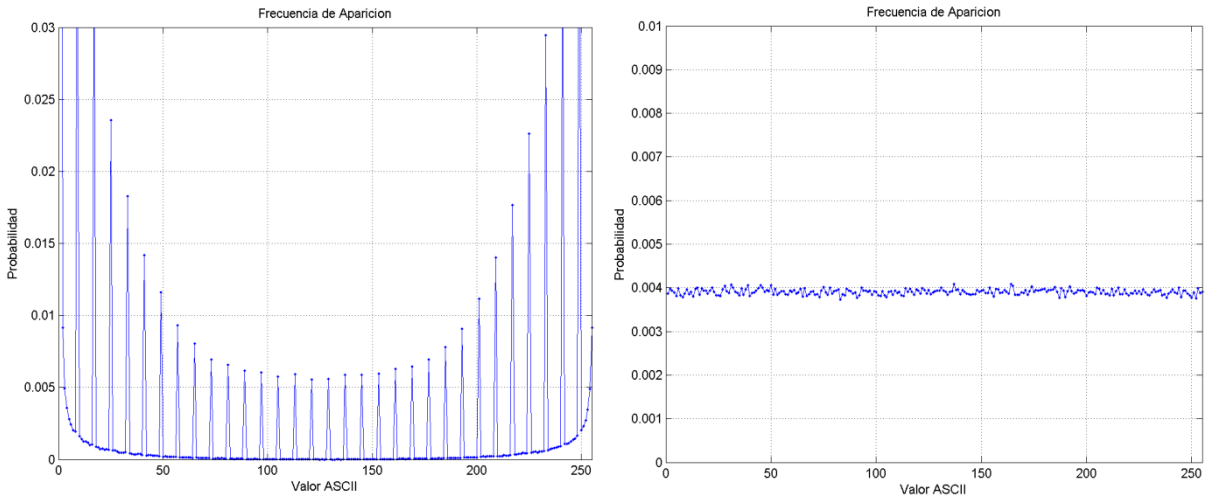
a)



b)



c)



d)

Figura 4.13 Histogramas de diferentes tipos de archivos. a)TXT, b)BMP, c)MP3-64 y d)WAV

Para este trabajo el proceso de cifrado se compone de 8 rondas, r=8. Cada ronda se compone de una red de Feistel desbalanceada que utiliza una subclave de 64 bits. La ronda se puede definir como un conjunto de funciones de transformación como sigue:

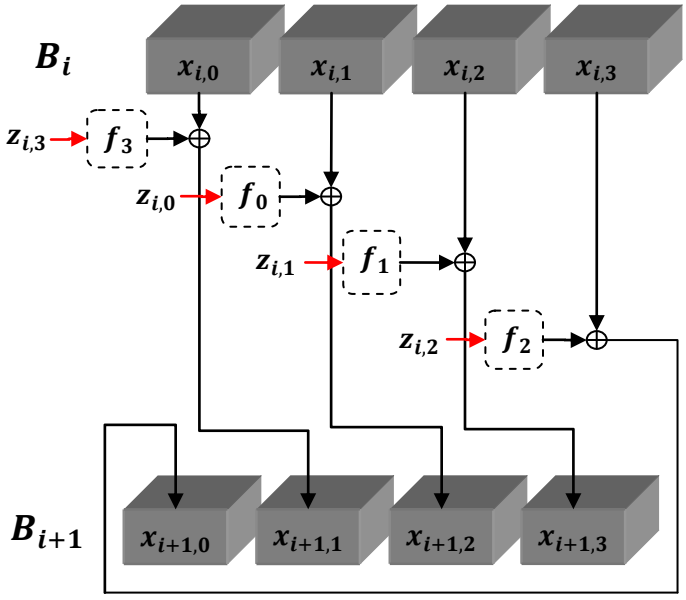


Figura 4.14 Proceso de Cifrado con 4 bloques de 16 bits cada uno.

$$
\begin{aligned}
x_{i+1,2} &= x_{i,1} \oplus f_0\left(z_{i-1,0}\right), \\
x_{i+1,3} &= x_{i,2} \oplus f_1\left(x_{i,1} \oplus z_{i-1,1}\right), \\
x_{i+1,0} &= x_{i,3} \oplus f_2\left(x_{i,1} \oplus x_{i,2} \oplus z_{i-1,2}\right), \\
x_{i+1,1} &= x_{i,0} \oplus f_3\left(x_{i,1} \oplus x_{i,2} \oplus x_{i,3} \oplus z_{i-1,3}\right).
\end{aligned} \qquad \textbf{(31)}
$$

Los valores $x_{i+1,1}, x_{i+1,2}, \ldots, x_{i+1,j}$, representan los bits del bloque del criptograma. Las f_j suman (suma lógica) el bloque o los bloques anteriores con la subclave correspondiente y el resultado se le aplica la transformación caótica Tent, excepto la f_0, ver la figura 4.15. Esto se puede apreciar mejor en cada uno de los esquemas de cifrado representados por las figuras 4.16 a 4.18.

Al igual que con el cifrador de 8 bloques, la función caótica utilizada para la construcción del cifrador está contenida en *F*, siendo *F* misma la función de transformación Tent. (ver ecuación 17), con $\mu \in (0,1)$ y $x \in (0,1)$. Sin embargo, y como recordaremos, en este dominio el cifrador no puede trabajar, ya que para cifrar, el dominio utilizado es el del alfabeto ASCII extendido, el cual se encuentra definido en el intervalo [0, 255], pero en este caso, estamos utilizando bloques de 16 bits, es decir 2^n, lo que es lo mismo, 2^{16}. Por lo tanto, la transformación Tent se tiene que escalar y discretizar del intervalo (0, 1) en los números reales, al intervalo (0, 65535) en los números enteros, como se ve en la ecuación 21, con *n=16.*

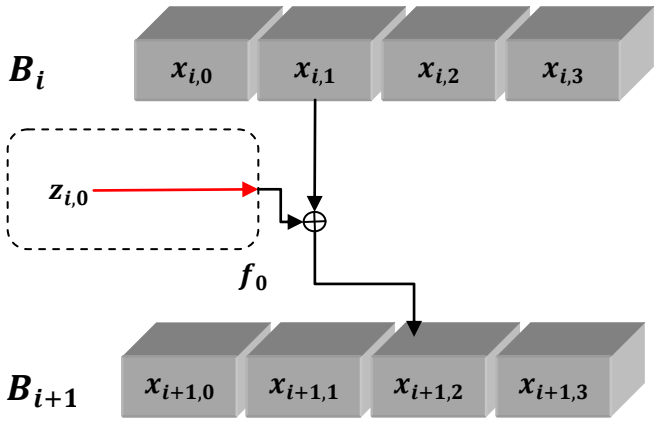


Figura 4.15 Esquema de cifrado de 4 bloques para f_0.

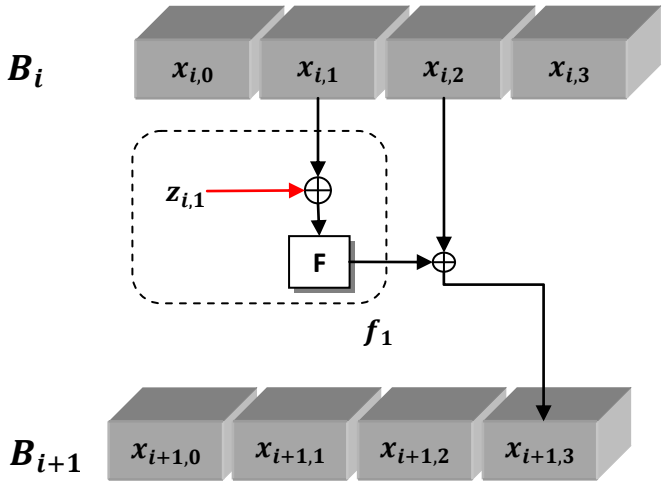


Figura 4.16 Esquema de cifrado de 4 bloques para f_1.

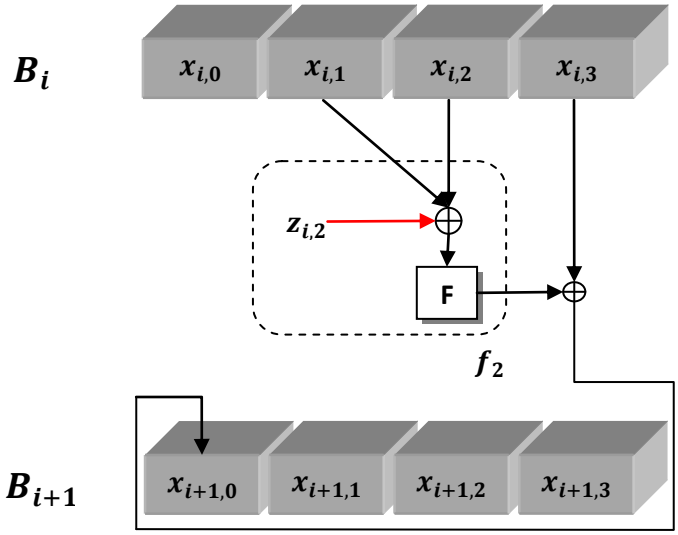


Figura 4.17 Esquema de cifrado de 4 bloques para f_2.

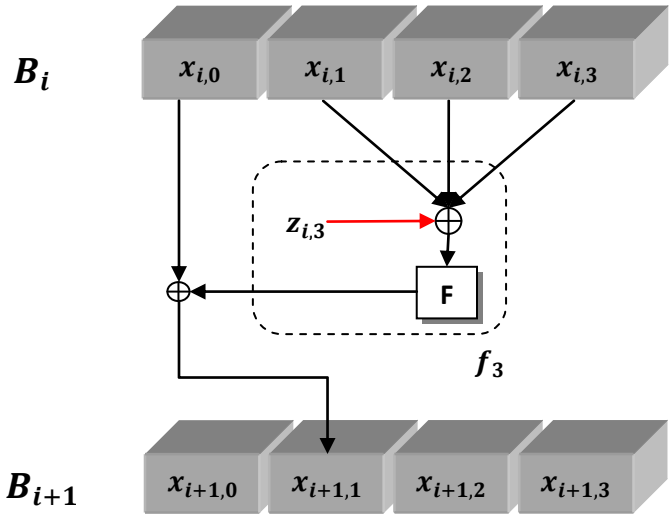


Figura 4.18 Esquema de cifrado de 4 bloques para f_3.

A continuación en la figura 4.20, se muestran las gráficas correspondientes a la parábola de la transformación escalada y discretizada. Esto es la cardinalidad del dominio es, 2_n, siendo *n* el número de bits, en este caso n=16, y hay que observar que μ está en función de *n*. Es importante tener en cuenta que la transformación discretizada es una aproximación a la transformación caótica Tent original.

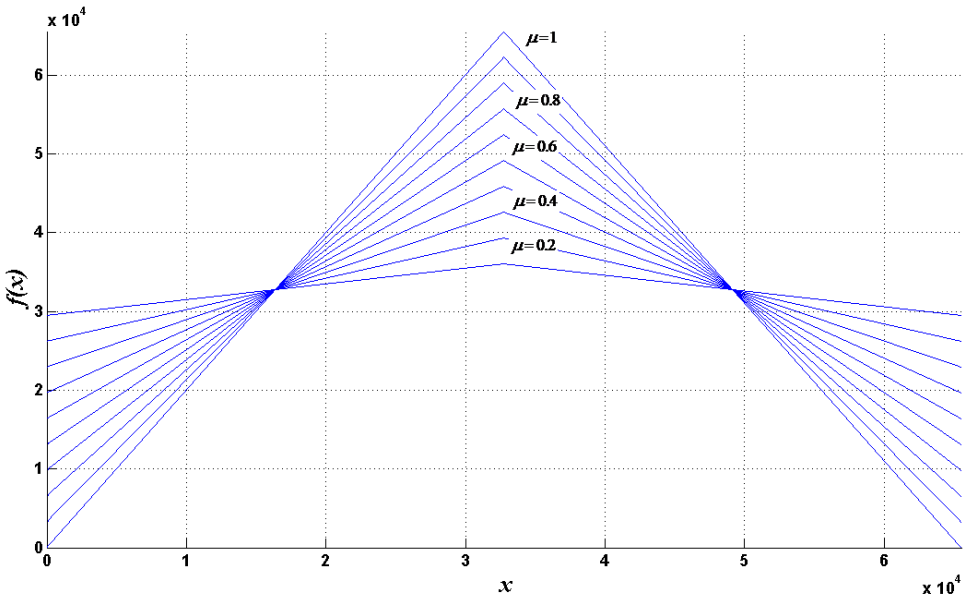


Figura 4.19 Familia de la transformación Tent escalada en el intervalo [0, 65535], con valores de μ=0.1 hasta μ=1 con incrementos de 0.1.

La transformación queda escalada y discretizada en el intervalo [0,65535], que es el dominio de definición de 2 bytes, es decir 2^{16}. Es importante mencionar que el escalamiento y la discretización en este intervalo ofrece un adecuado comportamiento, como se ve en las distintas gráficas de la figura 4.21, ya que cuando se usan 16 bits la Tent resultante es una muy buena aproximación a la Tent no discretizada. El diagrama de bifurcación, para cuando se usan 16 bits, es totalmente denso, como se puede ver en la figura 4.20, lo cual repercute en la distribución estadística de la señal generada usando la transformación Tent. Note que en cada caso la altura máxima de la parábola está definida por el parámetro μ .

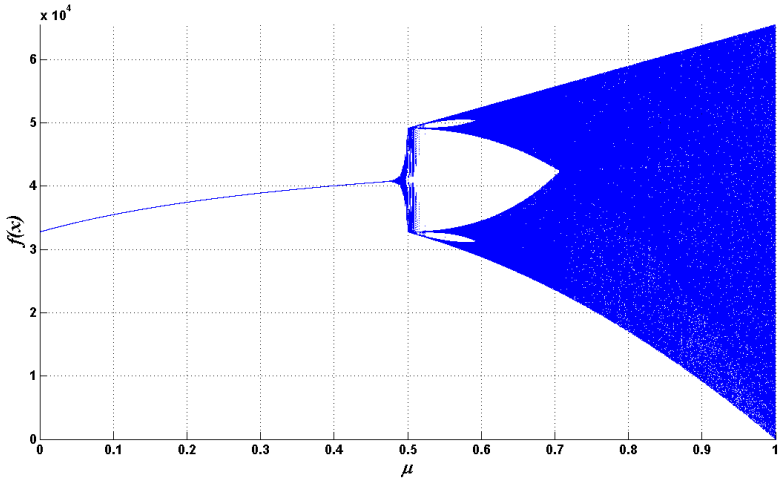
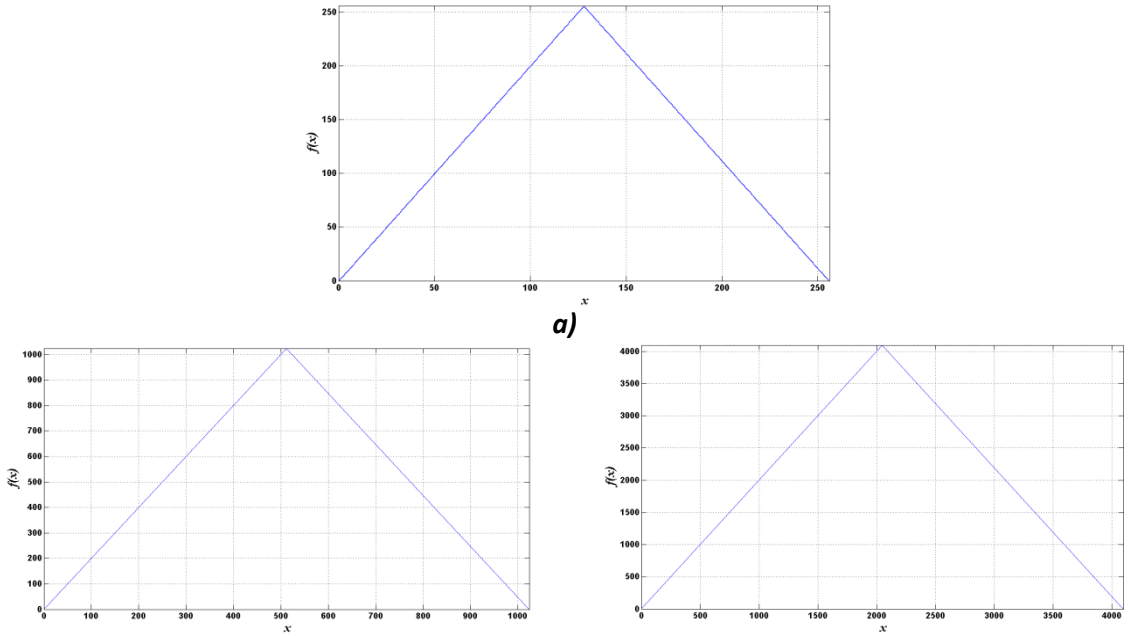


Figura 4.20 Diagrama de Bifurcación de la transformación caótica Tent, escalada y discretizada a 2^{16}.

El procedimiento de descifrado es similar: se aplican 4 rondas de descifrado sobre el bloque del criptograma B_j para así producir el bloque del mensaje original B_{j-1}. Las rondas de las subclaves son aplicadas ahora en forma inversa. El proceso de descifrado se muestra en la figura 4.22.

Proceso de generación de subclaves

El proceso de generación de subclaves, al igual que en el cifrador de 8 bloques, se aplica por la necesidad de quitar un problema al criptosistema, la **redundancia**. Son muy similares, con la única diferencia de tener 8 bloques de 8 bits, tenemos 4 bloques de 16 bits, como ya acabamos de ver.



a) ***b)*** ***c)***

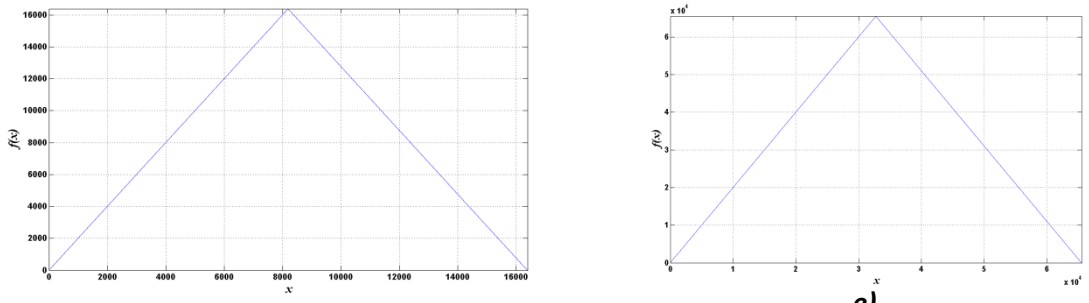


Figura 4.21 Curva de la transformación Tent escalada con valores de μ=1 usando distintos valores de n. a) 2^8 bits, b) 2^{10} bits, c) 2^{12} bits, d) 2^{14} bits, e) 2^{16} bits.

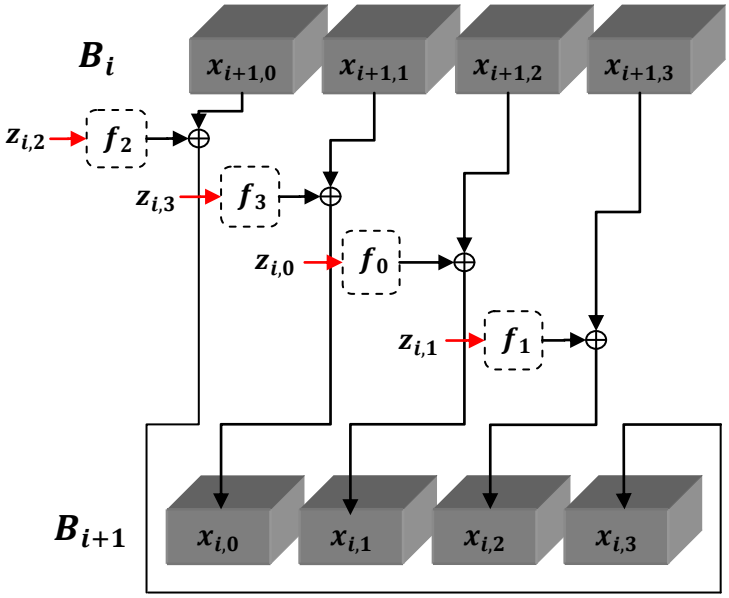


Figura 4.22 Esquema de descifrado para cifrador de 4 bloques.

Pruebas de funcionalidad del cifrador de 4 bloques

Al igual que con el cifrador de 8 bloques, se cifraron diferentes archivos; texto, audio, imágenes, etc., en diferentes formatos. Se seleccionan diferentes tipos de archivos para demostrar que sin importar como sea su huella estadística al ser cifrados tendrán una huella estadística parecida a una señal de ruido. Se calculan el histograma de los archivos originales y de los archivos cifrados y se comparan a fin de poder apreciar el grado de dispersión efectuado sobre los archivos cifrados con el proceso de transformación.

En la Tabla 3 se observan los tipos de archivos usados para probar el criptosistema. La primera prueba de funcionalidad es probar que en el proceso de cifrado y descifrado el criptograma

descifrado no pierda la información del mensaje original. Esta propiedad la deben de cumplir todos los criptosistemas.

Para verificar esta propiedad, se trabajan con 19 archivos de referencia de diferentes tipos, los cuales se cifran y descifran y se verifica su tamaño.

Tabla III Tamaño de los archivos de referencia.

Tipos de archivos	Tamaño del archivo original	Tamaño del archivo cifrado	Tamaño del archivo descifrado
TXT	213 KB	213 KB	213 KB
DOC	356 KB	356 KB	356 KB
RTF	3,124 KB	3,124 KB	3,124 KB
XML	997 KB	997 KB	997 KB
PPT	477 KB	477 KB	477 KB
RTFPDF	232 KB	232 KB	232 KB
XMLPDF	232 KB	232 KB	232 KB
DOCPDF	233 KB	233 KB	233 KB
PPTPDF	372 KB	372 KB	372 KB
BMP	2,236 KB	2,236 KB	2,236 KB
JPG	65 KB	65 KB	65 KB
PNG	1,038 KB	1,038 KB	1,038 KB
TIF	1,792 KB	1,792 KB	1,792 KB
MP3	3,346 KB	3,346 KB	3,346 KB
MP3-64	409 KB	409 KB	409 KB
MP3-128	816 KB	816 KB	816 KB
MP3-256	1,630 KB	1,630 KB	1,630 KB
WAV	820 KB	820 KB	820 KB
WMV	3,937 KB	3,937 KB	3,937 KB

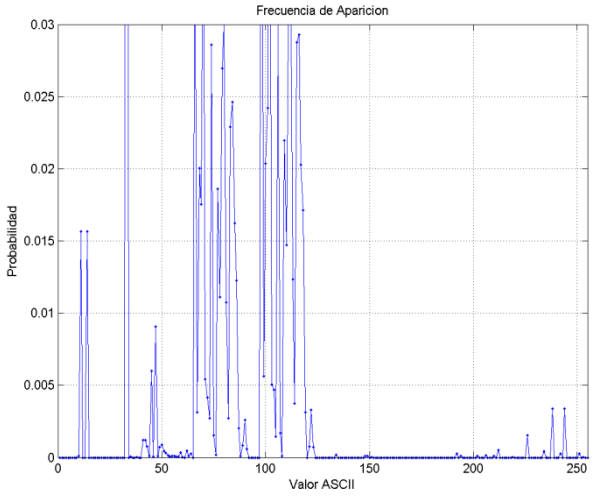
Este análisis no dice qué tan íntegra queda la información después de haber pasado por el criptosistema. Para probar esto, se utiliza una función HASH, en este caso la MD5. Se compara el resumen de los archivos originales y los descifrados. En la Tabla 4 se observan los resúmenes de cada archivo, y de todos los archivos después de descifrar mantienen su integridad. Por último, en la figura 4.23, se muestran las frecuencias relativas de 4 archivos, tanto los originales como su criptograma. En las gráficas de la figura 4.23 se puede apreciar, desde un punto de vista estadístico, el comportamiento de los archivos de originales. Las figuras del lado izquierdo corresponden a los archivos originales. Las figuras del lado derecho corresponden a los archivos cifrados con el cifrador de 4 bloques propuesto. Las gráficas de los archivos originales muestran huellas estadísticas particulares para cada tipo de archivo. Obsérvese que para archivos que tienen un proceso de compresión, como el MP3, sus histogramas de origen se parecen a una señal de ruido. Los procesos de compresión quitan redundancia, por eso los histogramas de estos archivos son casi uniformes.

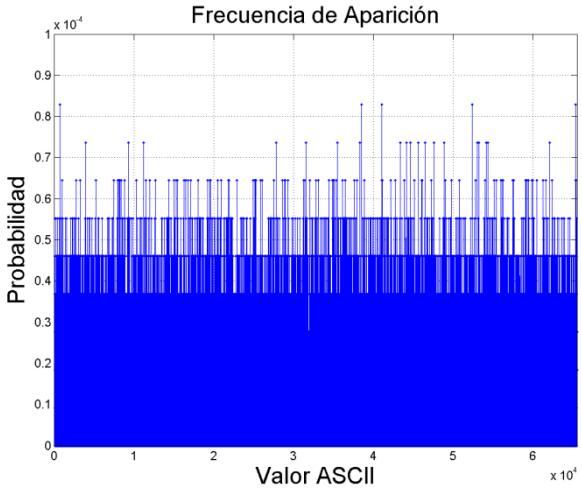
Tabla IV Comparación del Resumen de los archivos.

Tipo de archivo		Resumen MD5	Idénticos
TXT	Original	620d9e70dde7d67fc982924ecbe0c582	Sí
	Descifrado	620d9e70dde7d67fc982924ecbe0c582	
DOC	Original	4d6458c1e381a62651d57ef21f2a20b3	Sí
	Descifrado	4d6458c1e381a62651d57ef21f2a20b3	
RTF	Original	761f722320c5d21b5dafaa15835a746b	Sí
	Descifrado	761f722320c5d21b5dafaa15835a746b	
XML	Original	f71005c0610ec2e590be661c5584d623	Sí
	Descifrado	f71005c0610ec2e590be661c5584d623	
PPT	Original	2bddc62e67a5a3bb83a8678937ee1e5a	Sí
	Descifrado	2bddc62e67a5a3bb83a8678937ee1e5a	
RTFPDF	Original	ef0c169ede9fee819597b612f70749e7	Sí
	Descifrado	ef0c169ede9fee819597b612f70749e7	
XMLPDF	Original	4b31bb81187387f7e4fc5a33e063e73e	Sí
	Descifrado	4b31bb81187387f7e4fc5a33e063e73e	
DOCPDF	Original	9a7c17dc2924e0b584ed5d17994aefd0	Sí
	Descifrado	9a7c17dc2924e0b584ed5d17994aefd0	
PPTPDF	Original	f6acfd114e0f99154b0272c783f01c6f	Sí
	Descifrado	f6acfd114e0f99154b0272c783f01c6f	
BMP	Original	821342ad9411d307e4f305bc74d827e5	Sí
	Descifrado	821342ad9411d307e4f305bc74d827e5	
JPG	Original	c1b0078aea7ab45ab85c4cb353d98207	Sí
	Descifrado	c1b0078aea7ab45ab85c4cb353d98207	
PNG	Original	6ab2f6edb6be8173e3c5ac3d44002113	Sí
	Descifrado	6ab2f6edb6be8173e3c5ac3d44002113	
TIF	Original	8f6bfef6c2e64aec1dcb6fa92203a7dd	Sí
	Descifrado	8f6bfef6c2e64aec1dcb6fa92203a7dd	
MP3	Original	e408f15284e648bba81eab76d8abe871	Sí
	Descifrado	e408f15284e648bba81eab76d8abe871	
MP3-64	Original	b463984e0cac5319197ac93ecd6281b2	Sí
	Descifrado	b463984e0cac5319197ac93ecd6281b2	
MP3-128	Original	77f666ca44ab117808aaac0a673abd7d	Sí
	Descifrado	77f666ca44ab117808aaac0a673abd7d	
MP3-256	Original	c8d22965fc1458bf960f2a647bb6448b	Sí
	Descifrado	c8d22965fc1458bf960f2a647bb6448b	
WAV	Original	28163a2d1f4d41b86d550e419155678a	Sí
	Descifrado	28163a2d1f4d41b86d550e419155678a	
WMV	Original	257b2e61c7d4484bddcead4adb6d834d	Sí
	Descifrado	257b2e61c7d4484bddcead4adb6d834d	

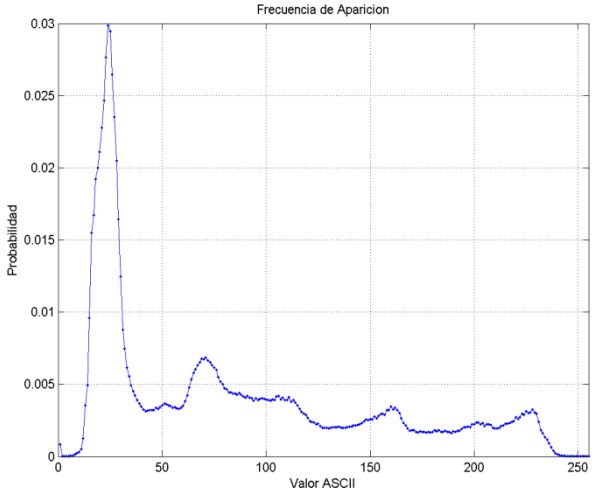
Originales

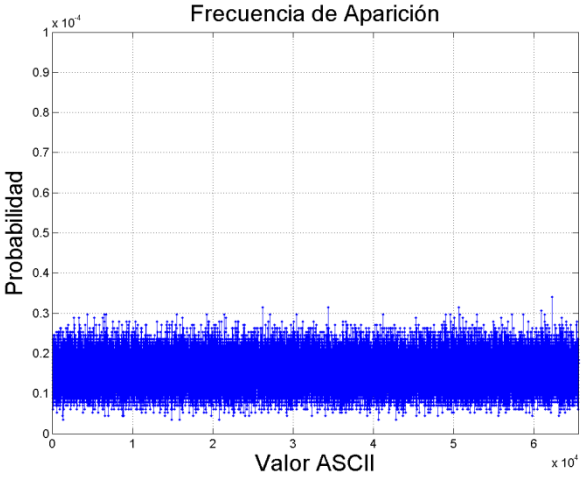
Cifrados



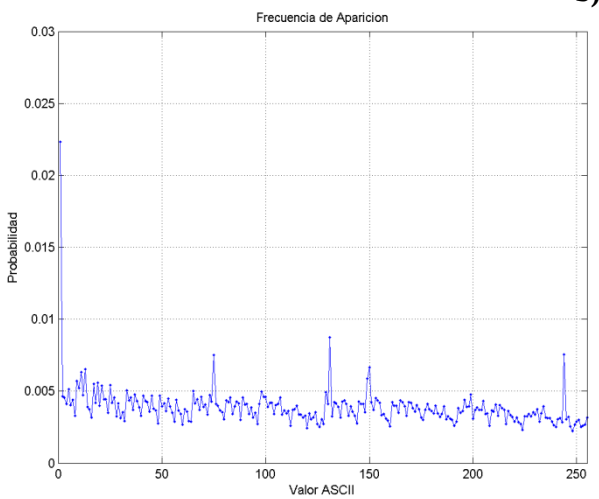


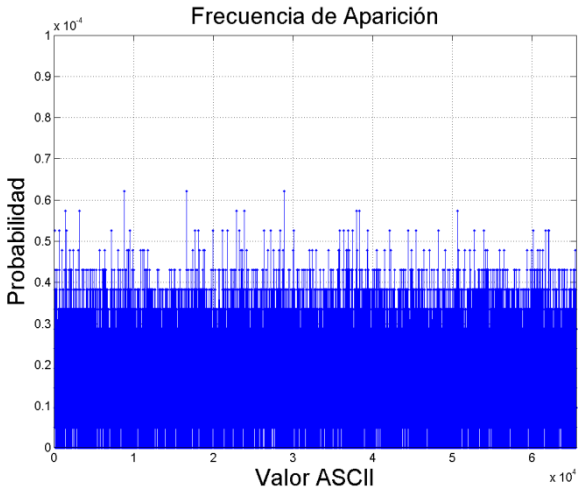
a)



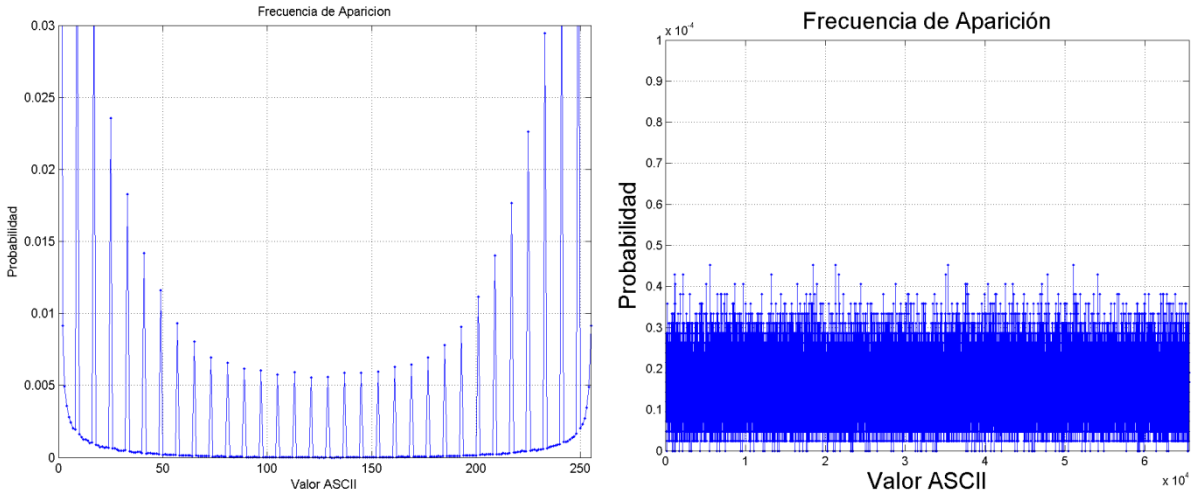


b)





c)



d)

Figura 4.23 Histogramas de diferentes tipos de archivos. a) TXT, b)BMP, c)MP3-64 y d)WAV

Todas las gráficas de los archivos cifrados presentan señales semejantes, a tal grado que son casi uniformes. Esto quiere decir que el cifrador de 4 bloques propuesto cambia exitosamente las propiedades de la semántica, la sintáctica y la estadística de los archivos.

4.3 Módulo de verificación de clave

Cifrado

Después de realizar el criptosistema de 4 bloques y haberlo evaluado satisfactoriamente con las herramientas de la mecánica estadística, así como la evaluación del mismo empleando conceptos de la teoría de la información como son Entropía e Información mutua del mensaje, como se verá un poco más adelante, se procedió a realizar un módulo de seguridad que ayuda a verificar la autenticación de la clave de descifrado, así como la autenticación del número de rondas utilizadas en el cifrado, y en caso de que no sean similares cualquiera de las dos, salir del proceso de descifrado sin realizar éste, ya que originalmente se realizaba el mencionado proceso, y aunque el resultado del descifrado no era en texto claro, un criptoanalista haciendo un análisis de los resultados, podría llegar a obtener el texto plano.

Para realizar el mencionado módulo de seguridad, se añadió en el mensaje cifrado C, la clave de cifrado y el número de rondas utilizados, obteniendo, evidentemente, un archivo más grande que el original.

Para realizar este proceso, una vez obtenido el mensaje cifrado, se procede a aplicarle a éste la función hash SHA-256, *h(C)*; así mismo, se le aplica la misma función hash a la clave de cifrado, *h(k)*. Al obtener los dos valores, estos se concatenan, para aplicar, a este nuevo valor, la función hash SHA-256. A este resultado, que es hexadecimal, se convierte a decimal, y se procede a cifrarlo con la función unidimensional de la Tent, previamente explicada, para posteriormente, el resultado de este cifrado, concatenarlo al mensaje cifrado. Se hace lo mismo para el número de rondas, cambiando *h(k)*, por *h(r)*. Ver figura 4.24.

Descifrado

Para realizar el proceso de verificación de la clave de descifrado y el número de rondas, contra las que fue cifrado el archivo, es necesario descomponer el mensaje cifrado en sus tres partes, es decir, el texto cifrado original, y las concatenaciones de la clave de cifrado y el número de rondas; una vez que se cuenta con esto, a la parte que corresponde al texto cifrado se le aplica la función hash SHA-256, $h(C)$, así como a la clave de descifrado $h(k)$, y una vez con estos resultados, se concatenan, y se le aplica la función hash SHA-256.

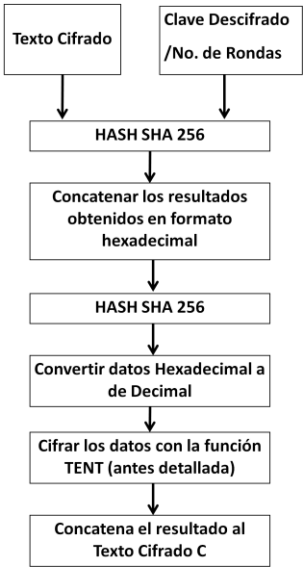


Figura 4.24 Diagrama del proceso de concatenación de la clave de cifrado y el número de rondas de cifrado, al mensaje cifrado.

El resultado se compara contra el valor concatenado de la clave de cifrado que se obtiene al descifrar la parte correspondiente a la concatenación de la clave de cifrado; si son iguales, se repite el proceso, pero ahora con los datos del número de rondas $k(r)$. Si no son iguales, se aborta el descifrado, sin que se entregue ningún resultado, para prever que éste pueda ser objeto de un criptoanálisis. Ver figura 4.25.

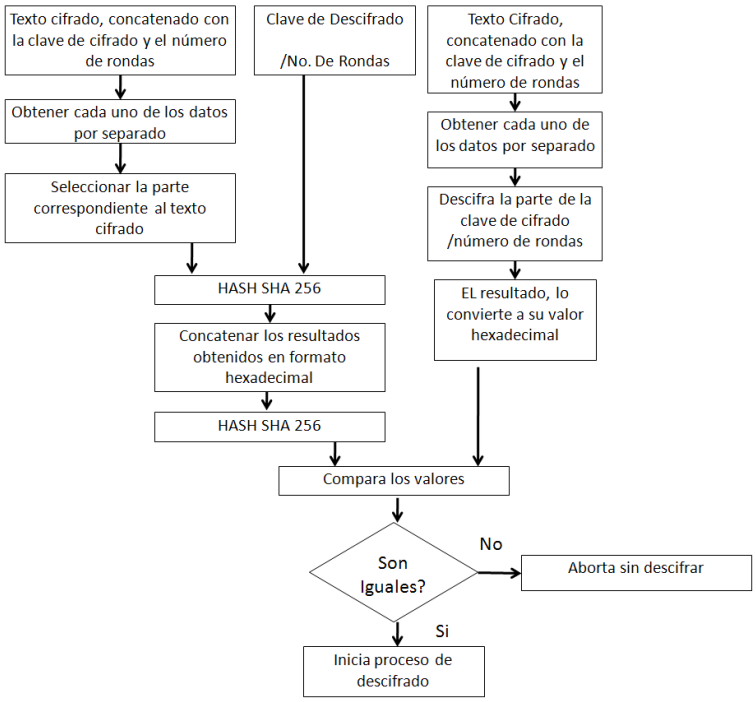


Figura 4.25 Diagrama del proceso de verificación de la clave de descifrado y el número de rondas contra las que fue cifrado el archivo.

4.4 Evaluación del Criptosistema utilizando la Teoría de la Información

Cálculo de la Entropía

Para evaluar qué tan eficiente es el cifrador propuesto, se cifraron 19 archivos digitales de diferentes tipos con los criptosistemas AES, DES, TRIPLE DES, BLOWFISH, IDEA Y SKIPJACK. Estos criptosistemas tienen propiedades semejantes a la propuesta, de ahí la elección de ellos. Los 19 archivos fueron cifrados con la misma clave.

Los archivos se cifran con los 2 cifradores propuestos, el cifrador de 8 bloques y el de 4 bloques; así como con cada uno de los diferentes tipos de criptosistemas mencionados.

Primero se evalúa la entropía de cada criptograma generado. Para saber cuál es la entropía máxima para el universo de los ASCII Extendidos, se desarrolla la ecuación 8, quedando como:

$$H(C) = -\sum_{i=1}^{n} P(x_1)\log_2[P(x_i)], \qquad donde \qquad P(x_i) = \frac{1}{256}$$
$$= -\sum_{i=1}^{256} \frac{1}{256}\log_2\left[\frac{1}{256}\right] = 8. \qquad \textbf{(32)}$$

Los resultados de las Entropías obtenidas se observan en la Tabla 5. Las entropías resaltadas son los mejores resultados para cada tipo de archivos, estos valores son los que se acercan más a la entropía máxima del sistema.

Las entropías del criptosistema propuesto son muy cercanas a las entropías de los demás criptosistemas. El criptosistema propuesto del cifrador con 8 bloques, obtiene 5 de las 19 mejores entropías más cercanas a la entropía máxima. La transformación senoidal tiene 4 de 19 de estas entropías, DES, Triple DES, Blowfish y Skipjack tienen 2 cada una; y finalmente AES e IDEA tienen solo 1 de las 19 entropías cada uno; el cifrado de 4 bloques y el logístico no tuvieron ningún mejor resultado.

También cabe mencionar que para hacer esta comparación, a los 2 cifradores propuestos NO se les agregó la parte del criterio de autenticación, para poder realizar una comparación real con los otros cifradores, ya que ellos NO cuentan con un módulo de validación.

Cálculo de la Información Mutua

Después se evalúan los criptogramas con la información mutua. En la Tabla 6 se observan los resultados. En esta se puede observar que la información mutua del cifrador de 8 bloques obtiene 1 de los mejores resultados; más sin embargo, la transformación caótica senoidal es quien obtiene el mejor resultado de esta prueba al obtener 5 mejores resultados; Cabe mencionar que el cifrador de 4 bloques no obtuvo ningún mejor resultado con respecto a todos los cifradores

En resumen se puede observar que se obtienen buenos resultados en la entropía con la transformación caótica de la Tent y con la transformación caótica senoidal se tiene mejores resultados en la información mutua.

4.5 Evaluación del Criptosistema utilizando la Teoría Estadística

Prueba de Aleatoriedad del NIST 800-22

Descripción

Diversas pruebas estadísticas pueden aplicarse a una secuencia para intentar compararla y evaluarla con una secuencia verdaderamente aleatoria. Aleatoriedad es una propiedad probabilística; es decir, las propiedades de una secuencia aleatoria pueden ser caracterizadas y descritas en términos de probabilidad. El resultado probable de las pruebas estadísticas, cuando se aplica a una secuencia verdaderamente aleatoria, se conoce a *priori* y puede describirse en términos probabilísticos. Hay un número muy grande de pruebas estadísticas, cada una evalúa la presencia o la ausencia de un ‘‘patrón’’ que, si se detecta, indicaría que la secuencia es no aleatoria. Debido a que hay tantas pruebas para juzgar si una secuencia es aleatoria, o no, no se puede establecer que exista un conjunto finito de pruebas se considere ‘‘completo’’. Además, los resultados de las pruebas estadísticas deben interpretarse con cuidados para evitar conclusiones incorrectas acerca de un generador específico.

Una prueba estadística está formulada para probar una hipótesis nula específica *(H0)*. En este documento, la hipótesis nula se refiere a que la secuencia que se está probando es aleatoria. Asociada a esta hipótesis nula, esta la hipótesis alternativa *(Ha)*, que se refiere a que la secuencia que se está probando no es aleatoria. Para cada prueba aplicada, una decisión o conclusión se deriva de que se acepta o rechaza la hipótesis nula; es decir, si el generador está o no produciendo valores aleatorios, basado en la secuencia que fue producida.

Para cada prueba debe escogerse una estadística relevante de aleatoriedad para determinar la aceptación o el rechazo de la hipótesis nula. Bajo el supuesto de aleatoriedad, una estadística tiene una distribución de valores posibles, la cual se puede determinar, bajo la hipótesis nula, por métodos matemáticos. De esta distribución de referencia, se determina un valor crítico (típicamente, este valor no se considera los extremos de la distribución, por ejemplo el 99%).

Tabla V Entropía Relativa de los 19 archivos.

ARCHIVOS	ORIGINAL	CIFRADOR PROPUESTO		LOGÍSTICO	SENOIDAL con PGSC	AES	DES	TRIPLE DES	BLOWFISH	IDEA	SKIPJACK
		8 BLOQUES	4 BLOQUES								
TXT	5.0802508	**0.9999062**	**0.9999014**	0.9995625	0.9998944	0.9998652	0.9998866	0.9998820	0.9999046	0.9999017	0.9999026
DOC	5.8174680	**0.9999468**	**0.9999332**	0.9969000	0.9999330	0.9999356	0.9999333	0.9999346	0.9999322	0.9999411	0.9999375
RTF	3.5727449	**0.9999923**	**0.9999719**	0.9994375	0.9999931	0.9999933	0.9999928	0.9999943	0.9999924	0.9999930	0.9999922
PPT	6.6895915	**0.9999512**	**0.9999478**	SIN REFERENCIA	0.9999579	0.9999447	0.9999545	0.9999482	0.9999557	0.9999578	0.9999558
XML	5.4487835	**0.9999771**	**0.9999745**	SIN REFERENCIA	0.9999791	0.9999770	0.9999746	0.9999792	0.9999784	0.9999761	0.9999771
DOCPDF	7.5195080	**0.9999045**	**0.9998908**	SIN REFERENCIA	0.9998956	0.9999106	0.9999047	0.9999091	0.9999164	0.9998994	0.9999055
RTFPDF	7.5197135	**0.9998954**	**0.9998972**	SIN REFERENCIA	0.9999026	0.9999044	0.9999163	0.9999060	0.9998961	0.9998970	0.9999057
PPTPDF	7.6155933	**0.9999408**	**0.9999374**	SIN REFERENCIA	0.9999380	0.9999442	0.9999349	0.9999389	0.9999386	0.9999414	0.9999449
XMLPDF	7.5188643	**0.9999011**	**0.9999019**	SIN REFERENCIA	0.9999191	0.9999126	0.9999087	0.9999048	0.9999129	0.9999018	0.9999160
JPG	7.9432769	**0.9996308**	**0.9996614**	0.9992375	0.9997045	0.9996396	0.9996564	0.9996670	0.9996340	0.9996708	0.9996372
TIF	7.7325211	**0.9999890**	**0.9999860**	0.9999875	0.9999869	0.9999855	0.9999868	0.9999870	0.9999874	0.9999866	0.9999872
PNG	7.9724736	**0.9999814**	**0.9999797**	SIN REFERENCIA	0.9999778	0.9999794	0.9999771	0.9999766	0.9999809	0.9999775	0.9999789
BMP	7.3018177	**0.9999892**	**0.9999896**	0.9999875	0.9999898	0.9999898	0.9999901	0.9999903	0.9999909	0.9999910	0.9999890
MP3	7.9634176	**0.9999943**	**0.9999935**	SIN REFERENCIA	0.9999939	0.9999931	0.9999931	0.9999940	0.9999943	0.9999934	0.9999929
MP3-64	7.9265755	**0.9999483**	**0.9999418**	0.9998375	0.9999436	0.9999514	0.9999456	0.9999396	0.9999467	0.9999425	0.9999489
MP3-128	7.8660824	**0.9999723**	**0.9999781**	0.9985375	0.9999786	0.9999675	0.9999729	0.9999721	0.9999732	0.9999723	0.9999736
MP3-256	7.9282939	**0.9999873**	**0.9999877**	0.9999250	0.9999875	0.9999846	0.9999848	0.9999873	0.9999845	0.9999867	0.9999879
WAV	4.5080543	**0.9999743**	**0.9999673**	0.9999875	0.9999725	0.9999728	0.9999765	0.9999741	0.9999751	0.9999722	0.9999712
WMV	7.9626926	**0.9999939**	**0.9999945**	SIN REFERENCIA	0.9999944	0.9999935	0.9999944	0.9999944	0.9999950	0.9999947	0.9999945
Total		**5/19**	**0/19**	**0/19**	**4/19**	**1/19**	**2/19**	**2/19**	**2/19**	**1/19**	**2/19**

Tabla VI Información Mutua de los 19 archivos.

ARCHIVOS	CIFRADOR PROPUESTO		LOGÍSTICO	SENOIDAL con PGSC	AES	DES	TRIPLE DES	BLOWFISH	IDEA	SKIPJACK
	8 BLOQUES	4 BLOQUES								
TXT	**0.0697607**	**0.0703152**	0.1777000	0.0688439	0.0710524	0.0700894	0.0699289	0.0718758	0.0695141	0.0714406
DOC	**0.1436924**	**0.1454969**	0.7356000	0.1430138	0.1434934	0.1440087	0.1438502	0.1437636	0.1449023	0.1438782
RTF	**0.0047022**	**0.0057360**	0.8744000	0.0046886	0.0047777	0.0048416	0.0047844	0.0047694	0.0047549	0.0048623
PPT	**0.1015175**	**0.1021768**	SIN REFERENCIA	0.1023523	0.1013163	0.1018438	0.1015368	0.1004735	0.1014454	0.1017251
XML	**0.0177862**	**0.0178172**	SIN REFERENCIA	0.0177533	0.0177723	0.0177288	0.0178956	0.0178747	0.0177751	0.0178305
DOCPDF	**0.2166705**	**0.2179811**	SIN REFERENCIA	0.2169031	0.2165742	0.2177017	0.2156614	0.2160382	0.2175431	0.2174396
RTFPDF	**0.2182947**	**0.2156433**	SIN REFERENCIA	0.2156085	0.2173077	0.2196499	0.2180281	0.2164405	0.2179679	0.2155744
PPTPDF	**0.1296359**	**0.1311940**	SIN REFERENCIA	0.1318094	0.1301556	0.1300224	0.1286948	0.1295876	0.1303589	0.1316981
XMLPDF	**0.2171596**	**0.2178841**	SIN REFERENCIA	0.2210456	0.2172430	0.2177842	0.2178913	0.2169651	0.2171931	0.2165054
JPG	**0.8046095**	**0.7952232**	0.3774000	0.7974379	0.7997339	0.7968874	0.7966479	0.8034175	0.7949750	0.8002418
TIF	**0.0257145**	**0.0257632**	0.0135000	0.0259903	0.0259519	0.0256978	0.0257062	0.0258542	0.0259238	0.0259979
PNG	**0.0443802**	**0.0447340**	SIN REFERENCIA	0.0447918	0.0448896	0.0446199	0.0446353	0.0447225	0.0446728	0.0446759
BMP	**0.0201754**	**0.0203473**	0.0219000	0.0203074	0.0201125	0.0200896	0.0202369	0.0200715	0.0200860	0.0203488
MP3	**0.0138111**	**0.0137530**	SIN REFERENCIA	0.0137731	0.0136527	0.0139307	0.0137321	0.0138242	0.0138376	0.0137762
MP3-64	**0.1155504**	**0.1158821**	0.2567000	0.1154685	0.1169477	0.1158613	0.1164297	0.1163459	0.1163869	0.1159267
MP3-128	**0.0571392**	**0.0575856**	0.2049000	0.0575314	0.0569216	0.0572494	0.0572891	0.0564386	0.0575616	0.0564292
MP3-256	**0.0282671**	**0.0283095**	0.0696000	0.0280030	0.0285086	0.0283371	0.0281878	0.0284937	0.0282826	0.0281313
WAV	**0.0450053**	**0.0449813**	0.0376000	0.0452828	0.0451589	0.0451134	0.0451638	0.0452104	0.0451912	0.0449093
WMV	**0.0116217**	**0.0117358**	SIN REFERENCIA	0.0117974	0.0115693	0.0116329	0.0117380	0.0117136	0.0117131	0.0116312
Total	**1/19**	**0/19**	**3/19**	**5/19**	**2/19**	**1/19**	**2/19**	**2/19**	**0/19**	**3/19**

Durante una prueba, se calcula un valor que se compara con el valor crítico. Si el valor de la prueba estadística supera el valor crítico, se rechaza la hipótesis nula de aleatoriedad. De lo contrario, no se rechaza la hipótesis nula, es decir, se acepta la hipótesis.

Valoración del algoritmo propuesto

Para realizar la valoración del NIST se crearon criptogramas con el criptosistema propuesto, el AES y el DES. Los criptogramas obtenidos se binarizaron y se obtuvo una secuencia binaria de 100,000,000 de longitud. De acuerdo con el NIST 800-22 [66] la opción más viable para los valores de entrada son: 100 números de cadenas de bits generados con longitud de 1,000,000; en pocas palabras 100 cadenas de bits de 1,000,000. Las pruebas de bloque de frecuencia, no superposición de secuencias, superposición de secuencias, aproximación de la entropía, serial, universal y complejidad lineal, son pruebas las cuales se tienen que seleccionar parámetros de acuerdo a su longitud de cadenas de bits, estos valores son mencionado en [2] y cada prueba tiene un método propio para obtenerlo. Los parámetros seleccionados se muestran adelante del nombre de la prueba. Para el caso de las pruebas de excursión aleatoria y variante de excursión aleatoria se tienen 8 y 18 p-valores resultantes respectivamente, la x representa el número de p-valor seleccionado. En el caso de la prueba serial se tienen 2 p-valores resultantes.

Tabla VII Resultados de las Pruebas del NIST.

No.	PRUEBA	TENT		SENOIDAL		LOGISTICO	AES	DES	REVISION
		8-BITS	16-BITS	Sin PGSC	Con PGSC				
1	Frequency	0.699313	0.595549	0.051942	0.096578	0.2826	0.946308	0.015598	PASA
2	Block-Frequency m=1000	0.816537	0.366918	0.494392	0.983453	0.3392	0.455937	0.739918	PASA
3	Cumulative-sums Forward	0.319084	0.935716	0.030806	0.719747	0.8816	0.924076	0.350485	PASA
	Cumulative-sums Reverse	0.867692	0.719747	0.002374	0.383827	0.2248	0.779188	0.075719	PASA
4	Runs	0.964295	0.935716	0.002971	0.851383	0.3369	0.955835	0.23681	PASA
5	Longest-Runs of Ones	0.834308	0.657933	0.026948	0.983453	0.3431	0.759756	0.637119	PASA
6	Rank	0.145326	0.350485	0.085587	0.678686	0.407	0.350485	0.319084	PASA
7	Spectral fft	0.534146	0.042808	0.181557	0.090936	0.4559	0.075719	0.289667	PASA
8	Non-Overlapping-templates	0.504952	0.452137	0.419021	0.137282	0.5341	0.719747	0.224821	PASA
9	Overlapping Templates m=10	0.514124	0.779188	0.010988	0.437274	0.7981	0.383827	0.419021	PASA
10	Universal	0.554420	0.334538	0.071177	0.455937	0.2133	0.678686	0.924076	PASA
11	Aproximate Entropy m=10	0.020548	0.401199	0.004301	0.574903	0.7177	0.759756	0.350485	PASA
12	Random-Excursions	0.453897	0.504708	0.007694	0.602458	0.9642	0.834308	0.304126	PASA
13	Random-Excursions-Variant	0.400951	0.378937	0.137282	0.350485	0.9114	0.77276	0.12962	PASA
14	Linear ∇complexity m=1000	0.085587	0.595549	0.383827	0.366918	0.6392	0.55442	0.213309	PASA
15	Serial ∇Ψ_m^2 (obs)	0.419021	0.798139	0.051942	0.334538	0.0145	0.946308	0.015598	PASA
	Serial ^2 Ψ_m^2 (obs)	0.401199	0.935716	0.000001	0.514124	0.9642	0.000001	0.000055	PASA

Comentarios al capítulo

En este capítulo se conoció el algoritmo de cifrado propuesto por Kocarev, y se propuso el algoritmo de cifrador de bloques utilizando la transformación caótica de la Tent en sus dos versiones. Al final de este capítulo se dan a conocer los resultados de esta implementación comparándolos con los obtenidos con las transformaciones caóticas logística y senoidal; y otros cifradores comerciales.

CONCLUSIONES Y COMENTARIOS

Los criptosistemas propuestos en esta tesis fueron evaluados en dos sentidos, en funcionalidad y en eficiencia, para lo cual se usaron los conceptos de la Teoría de la Información. Los criptosistemas propuestos cifraron y descifraron correctamente, según se pudo apreciar en las tablas 1 a la 4. Cabe mencionar que el software con que fue desarrollado este criptosistema es Matlab©, por lo que los procesos los ejecutaban lentamente, ya que no es propiamente un lenguaje, sino un intérprete. Al evaluar los criptosistemas propuestos con los conceptos de la Teoría de la Información, se calculó la entropía de los criptogramas y se comparó con la entropía de los mensajes originales; en todos los casos, la entropía de los criptogramas fue superior, tendiendo a la entropía máxima que es 8. El archivo que tuvo la máxima entropía para el cifrador de 8 bloques fue el RTF con una entropía relativa máxima de 0.9999923, y para el cifrador de 4 bloques, el archivo fue MP3 con un valor de 0.9999935. Por otra parte, el archivo que tuvo la menor Información Mutua, en el cifrador de 8 bloques fue el RTF, que tuvo un valor de 0.0047022, y para el cifrador de 4 bloques, también el archivo fue el de RTF con un valor de 0.0057360. Esto pudo ser corroborado a partir de los resultados obtenidos de las pruebas estadísticas del NIST. Al analizar los criptogramas como secuencias generadas por el criptosistema propuesto y las secuencias de los criptogramas generados por los criptosistemas más usados, se observa que para el criptosistema propuesto todas las pruebas rebasan el mínimo valor requerido para considerar que existe aleatoriedad en esa secuencia. Por ello, los criptogramas generados por el criptosistema propuesto se consideran aleatorios, y con ello se consideran sus distribuciones estadísticas cercanas a la distribución uniforme.

La seguridad de los criptosistemas que se están proponiendo en esta tesis, se ve incrementada gracias al módulo de verificación de clave agregado, ya que en caso de estar sufriendo un ataque por parte de alguna persona no autorizada, si esta no introduce la clave de cifrado correctamente, así como el número de rondas de cifrado, el proceso de descifrado no inicia.

Así mismo, aunque se esperaba una ganancia sustancial al contar con 4 bloques de cifrado con 16 bits cada uno, al tener los resultados no es tan evidente, por lo que realmente no valió la pena realizar el cifrador de 4 bloques; sin embargo se tuvo que realizar para poder concluir esto último.

La transformación caótica utilizada en esta tesis, la Tent, ofrece mejores condiciones de comportamiento en un criptosistema caótico de bloques, que cuando se usan otras transformaciones, como la logística, y la senoidal, incluso cuando se utilizan criptosistemas comerciales actuales, según lo reportado en los

resultados de la Entropía relativa, en donde se puede observar que en el caso de el cifrador de 8 bloques, tuvo mejor resultado en 5 tipos de archivo de 19.

Sin embargo, en el cálculo de la Información Mutua, la transformación caótica senoidal tuvo mejores resultados, obteniendo 5 mejores resultados de 19, contra 1 solo del cifrador propuesto de 8 bloques.

Así pues, se han cumplido con todos los objetivos generales y particulares de esta tesis, concluyendo satisfactoriamente el estudio que se realizó de la Transformación Caótica Unidimensional de la Tent, para su utilización en criptosistemas caóticos de bloques.

TRABAJO A FUTURO

- Este trabajo se desarrolló en Matlab©, haciendo lento la ejecución de los programas, por lo que uno de los trabajos a futuro, es desarrollar el software en un lenguaje de bajo nivel, probablemente Java, y con esto, bajar el tiempo de ejecución.

- Así mismo, sería conveniente realizar este desarrollo en su versión de hardware, para tratar de bajar aún más los tiempos de ejecución.

- Por otra parte, hemos notado que al escalar y discretizar la función para que podamos utilizarla en el universo de los ASCII Extendido, no es tan eficiente, por lo que se recomienda utilizar otras técnicas para lograr hacer esto.

REFERENCIAS

[1] **Stalling W.**, "*Fundamentos de seguridad en redes. Aplicaciones y Estándares*". Segunda Edición. Pearson Education, S.A. Madrid, 2004.

[2] **National Institute of Standards and Technology.** "*A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*" Revision 1, Special Publication 800-22. Revised April 2010.

[3] **Badillo R., Campos I., Campos E.,** "*Transmisión y Recepción de Voz Empleando Caos*" Encuentro de Investigación en Ingeniería Eléctrica, Zacatecas, Zac, Abril 5-7, 2006

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

[5] **Kocarev L., Jakimoski G.,** *"Logistic Map as a Block Encryption Algorithm"*, Physics Letters, October 2001.

[6] **Chengqing Li, Shujun Li, Gonzalo Álvarez, Guanrong Chen and Kwok-Tung Lo,** "*Cryptanalysis of a chaotic block cipher with external key and its improved version*", Chaos, Solitons & Fractals, Volume 37, Issue 1, July 2008, Pages 299-307.

[7] **Huaqian Yang, Xiaofeng Liao, Kwok-wo Wong, Wei Zhang and Pengcheng Wei, "***A new block cipher based on chaotic map and group theory***",** Chaos, Solitons & Fractals, Volume 40, Issue 1, 15 April 2009, Pages 50-59.

[8] **William C. Saphir and Hiroshi H. Hasegawa, "***Spectral representations of the Bernoulli map***",** Physics Letters A, Volume 171, Issues 5-6, 14 December 1992, Pages 317-322.

[9] **Richard Bedient and Michael Frame**, "Carrying surfaces for return maps of averaged logistic maps", Computers & Graphics, Volume 31, Issue 6, December 2007, Pages 887-895

[10] **Yong Wang, Kwok-Wo Wong, Xiaofeng Liao and Tao Xiang**, *"A block cipher with dynamic S-boxes based on tent map",* Communications in Nonlinear Science and Numerical Simulation, Volume 14, Issue 7, July 2009, Pages 3089-3099.

[11] **Erez Petrank and Charles Rackoff**, "*CBC MAC for Real-Time Data Sources*", Journal of Cryptology, Volume 13, Number 3 / diciembre de 2000, pages 315-338.

[12] **M. Bellare, J. Kilian and P. Rogaway**. "The security of Cipher Block Chaining. *Advances in Cryptology – Crypto '94 Proceedings*", Lecture Notes in Computer Science Vol. 839, Springer-Verlang, Y. Desmendt, Ed., pp. 340-358, 1994.

[13] **Thomas Jakobsen and Lars R. Knudsen,** "*Attacks on Block Ciphers of Low Algebraic Degree*", Journal of Cryptology, Volume 14, Number 3 / diciembre de 2001, pages 197-210.

[14] **K. Nyberg and L.R. Knudsen,** "*Provable security against a differential attack*", The Journal of Cryptology, 8(1):27-38, 1995.

[15] **Knudsen Lars R**., "*The security of Feistel ciphers with six rounds or less*" Journal of cryptology, 2002, vol. 15, no3, pp. 207-222 (31 ref.)

[16] **Ju-Sung Kang, Bart Preneel, Heuisu Ryu, Kyo II Chung, and Chee Hang Park,** "*Pseudorandomness of Basic Structures in the Block Cipher KASUMI*" ETRI Journal, vol.25, no.2, Apr. 2003, pp.89-100.

[17] **Serge Vaudenay**, "*Decorrelation: A Theory for Block Cipher Security*", Journal of Cryptology, vol. 16, num. 4 / septiembre de 2003, p. 249-286.

[18] **John Black and Phillip Rogaway,** "*CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions*", Journal of Cryptology, Volume 18, Number 2 / abril de 2005, pages 111-131.

[19] **Shiguo Lian, Jinsheng Sun and Zhiquan Wang**, "*A block cipher based on a suitable use of the chaotic standard map*",Chaos, Solitons & Fractals, Volume 26, Issue 1, October 2005, Pages 117-129.

[20] **Muhammad Asim and Varun Jeoti**, "*Efficient and Simple Method for Designing Chaotic S-Boxes*", ETRI Journal, vol.30, no.1, Feb. 2008, pp.170-172.

[21] **Charanjit S. Jutla**, "*Encryption Modes with Almost Free Message Integrity*", Journal of Cryptology, Volume 21, Number 4 / octubre de 2008, pages 547-578.

[22] **L. M. Pecora and T. L. Carroll**, "*Synchronization in chaotic systems"*. Phys. Rev. Lett., 64:821–824, 1990.

[23] **L. Kocarev, K. S. Halle, K. Eckert, L. O. Chua, and U. Parlitz**. "*Experimental demonstration of secure communications via chaotic synchronization*". Int. J. Bifurc. Chaos, 2:709–713, 1992.

[24] **C. W. Wu and L. O. Chua**. "*A simple way to synchronize chaotic systems with applications to secure communications systems*". Int. J. Bifurc. Chaos, 3:1619–1627, 1993.

[25] **K. M. Cuomo, A. V. Openheim, and S. H. Strogatz**. "*Synchronization of lorenz-based chaotic circuits with applications to communications*". IEEE Trans. Circuits Syst – II, 40:626–633, 1993

[26] **O. Morgul and M. Feki**. "*A chaotic masking scheme by using synchronized chaotic systems*". Phys. Lett. A, 251:169–176, 1999.

[27] **S. M. Shahruz, A. K. Pradeep, and R. Gurumoorthy**. "*Design of a novel cryptosystem based on chaotic oscillators and feedback inversion"*. J. Sound and Vibration, 250:762–771, 2002.

[28] **H. Dedieu, M. P. Kennedy, and M. Hasler**. "*Chaos shift keying: Modulation and demodulation of a chaotic carrier using self-synchronizing*". IEEE Trans. Circuits and Systems – II, 40:634–641, 1993.

[29] **U. Parlitz, L. O. Chua, L. Kocarev, K. S. Halle, and A. Shang**. "*Transmission of digital signals by chaotic synchronization*". Int. J. Bifurcation. Chaos, 2:973–977, 1992.

[30] **K. S. Halle, C. W. Wu, M. Itoh, and L. O. Chua**. "*Spread spectrum communication through modulation of chaos*". Int. J. Bifurcation. Chaos, 3:469–477, 1993.

[31] **K.M. Cuomo and A. V. Openheim**. "*Circuit implementation of synchronized chaos with applications to communications*". Phys. Rev. Lett., 71:65–68, 1993.

[32] **M. Feki**. "*An adaptive chaos synchronization scheme applied to secure communication*". Chaos, Solitons and Fractals, 18:141–148, 2003.

[33] **J. Y. Chen, K. W. Wong, L. M. Cheng, and J. W. Shuai**. "*A secure communication scheme based on the phase synchronization of chaotic systems*". Chaos, 13:508–514, 2003.

[34] **S. Hayes, C. Grebogi, and E. Ott**. "*Communicating with chaos*". Phys. Rev.Lett., 70:3031–3034, 1993.

[35] **S. Hayes, C. Grebogi, E. Ott, and A. Mark**. "*Experimental control of chaos for communication*". Phys. Rev. Lett., 73:1781–1784, 1994.

[36] **C. Grebogi, Y. Lai, and E. Bolt**. "*Communicating with chaos using two-dimensional symbolic dynamics"*. Phys. Lett. A, 255:75–81, 1999.

[37] **M. S. Baptista**. "*Cryptography with chaos*". Phys. Lett. A, 240:50–54, 1998.

[38] **J. Fridrich**. "*Symmetric ciphers based on two-dimensional chaotic maps*". Int. J. Bifurc. Chaos, 8:1259–1284, 1998.

[39] **N. K. Pareek, V. Patidar, and K. K. Sud**. "*Discrete chaotic cryptography using external key*". Phys. Lett. A, 309:75–82, 2003.

[40] **W. Wong, L. Lee, and K. Wong**. "*A modified chaotic cryptographic method*". Computer Physics Communications, 138:234–236, 2001.

[41] **E. Álvarez, A. Fernández, P. García, J. Jiménez, and A. Marcano**. "*New approach to chaotic encryption*". Phys. Lett. A, 263:373–375, 1999.

[42] **P. García and J. Jiménez**. "*Communication through chaotic map systems*". Phys. Lett. A, 298:35–40, 2002.

[43] **Di Xiao, Xiaofeng Liao, Shaojiang Deng**, "*A novel key agreement protocol based on chaotic maps*", Information Sciences 177 (2007) 1136–1142.

[44] **Bonseok Koo, Gwonho Ryu, Taejoo Chang and Sangjin Lee,** "*Design and Implementation of Unified Hardware for 128-Bit Block Ciphers ARIA and AES*", ETRI Journal, vol.29, no.6, Dec. 2007, pp.820-822.

[45] **S. Behnia, A. Akhshani, H. Mahmodi and A. Akhavan** "*A novel algorithm for image encryption based on mixture of chaotic maps*", Chaos, Solitons & Fractals, Volume 35, Issue 2, January 2008, Pages 408-419.

[46] **S. Behnia, A. Akhshani, A. Akhavan and H. Mahmodi,** "*Applications of tripled chaotic maps in cryptography*", Chaos, Solitons & Fractals, Volume 40, Issue 1, 15 April 2009, Pages 505-519.

[47] **Ali Kanso and Nejib Smaoui,** "*Logistic chaotic maps for binary numbers generations*", Chaos, Solitons & Fractals, Volume 40, Issue 5, 15 June 2009, Pages 2557-2568.

[48] **Huaqian Yang, Xiaofeng Liao, Kwok-wo Wong, Wei Zhang and Pengcheng Wei,** "*A new cryptosystem based on chaotic map and operations algebraic*", Chaos, Solitons & Fractals, Volume 40, Issue 5, 15 June 2009, Pages 2520-2531.

[49] **Fuyan Sun and Shutang Liu,** "*Cryptographic pseudo-random sequence from the spatial chaotic map*", Chaos, Solitons & Fractals, Volume 41, Issue 5, 15 September 2009, Pages 2216-2219.

[50] **Beker H, Piper F.,** "*Cipher systems: the protection of communications*". New York: van Nostrand Reinhold; 1982.

[51] **NIST. Federal Information Processing Standards Publication (FIPS140-1).** Security requirements for cryptographic modules; 1994.

[52] **Cruselles Forner, Ernesto J., Melus Moreno, José Luis,** *"Secuencias pseusoaleatorias para telecomunicaciones"* Ediciones UPC, S.L. 1. ed.(09/1996).

[53] **NIST. Federal Information Processing Standards** Publication 185, 1994.

[54] **Schneier, Bruce y Kelsey, John. "***Unbalanced Feistel Networks and Block-Cipher Design".* Springer Berlin, 2006.

[55] **Lucena López, Manuel J.** *Criptografía y Seguridad en Computadores.* 2010.

[56] Wikipedia. http://es.wikipedia.org/wiki/Teor%C3%ADa_de_la_informaci%C3%B3n.

[57] **Vázquez Medina, Rubén**, *"Mapeos caóticos unidimensionales aplicados a la generación de ruido"* Tesis de Doctorado, Octubre 2008, Universidad Autónoma Metropolitana. México.

[58] **Luengo Garcia, David,** *"Estimación óptima de secuencias caóticas con aplicación de comunicaciones",* Tesis de Doctorado, Septiembre 2006, Universidad de Cantabria. España.

[59] **Hilborn, Robert C.,** "*Chaos and Nonlinear Dynamics An Introduction for Scientists and Engineers*", Department of Physics, Amherst College, Second Edition, Oxford University Press, 2000, Pág. 185.

[60] **Benítez Barrón, Luz M.,** *"Mapeo caótico Senoidal aplicado al cifrado de bloques",* Tesis de Maestría en Ciencias, Septiembre 2010, Instituto Politécnico Nacional, México.

[61] **Feistel, H.** *Cryptography and computer privacy.* s.l. : Scientific American, 1973.

[62] **Blaze, Matt y Schneier, Bruce.** *The MacGuffin block cipher algorithm.* s.l. : In Bart Preneel, 1994.

[63] **Anderson, R. y Biham, E.** *Two Practical and provably secure block ciphers: BEAR and LION.* s.l. : Springer-Verlag, 1996.

[64] **Lai, X. y Massey, J.L.** *A proposal for a new block encryption standard.* s.l. : Springer-Verlag, 1991.

[65] **J.L., Massey.** *SAFER K-64: A byte orientated block-ciphering algorithm.* s.l. : Springer, 1993.

[66] **National Institute of Standards and Technology,** "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptography Applications". Revision 1, 2008, Vols. Publicación especial 800-22.

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

APÉNDICE A

Publicaciones en Congresos

"Cryptochaos: Algoritmo de cifrado de bloques con transformaciones caóticas unidimensionales", **C. E. Rojas-López**, C. Cortés-Bazán, J. A. Martínez-Ñonthe y R. Vázquez-Medina. II Jornadas de Ingeniería y Tecnología, Refinería "Miguel Hidalgo", en Tula de Allende, Hgo., celebrado del 30 de noviembre al 2 de diciembre de 2010.

"*Cifrador caótico de bloques Tent con criterio de autenticación y precisión de 16 bits*", **C. E. Rojas-López,** J. A. Martínez-Ñonthe, R. Vázquez-Medina. Vigésima primera Reunión de Otoño de Comunicaciones, Computación, Electrónica y Exposición Industrial 2010, IEEE Sección México, Acapulco, Gro., celebrado del 28 de noviembre al 4 de diciembre de 2010.

"*Análisis comparativo de criptosistemas caóticos: analógicos y digitales*", L. Palacios-Luengas, **C. E. Rojas-López,** J. A. Martínez-Ñonthe, A. Castañeda-Solís, R. Vázquez-Medina. Vigésima primera Reunión de Otoño de Comunicaciones, Computación, Electrónica y Exposición Industrial 2010, IEEE Sección México, Acapulco, Gro., celebrado del 28 de noviembre al 4 de diciembre de 2010.

"*Criptosistema caótico de bloques usando la transformación Tent*", J. A. Martínez-Ñonthe, **C. E. Rojas-López,** J. L. Del Rio-Correa, J. A. Díaz-Méndez, R. Vázquez-Medina. LIII Congreso Nacional de Física, Boca del Río. Ver., celebrado del 25 al 29 de octubre de 2010.

"Estrategias de construcción de claves para cifradores de bloques", L. M. Benítez-Barrón, J. A. Martínez-Ñonthe, **C. E. Rojas-López**, R. Vázquez-Medina. Vigésima Reunión de Otoño de Comunicaciones, Computación, Electrónica y Exposición Industrial 2009, IEEE Sección México, Acapulco, Gro., celebrado del 29 de Noviembre al 5 de Diciembre de 2009.

"Cifrador caótico de bloques logístico", J. A. Martínez-Ñonthe, L. M. Benítez-Barrón, **C. E. Rojas-López,** R. Vázquez-Medina, Vigésima Reunión de Otoño de Comunicaciones, Computación, Electrónica y Exposición Industrial 2009, IEEE Sección México, Acapulco, Gro., celebrado del 29 de Noviembre al 5 de Diciembre de 2009.

Cryptochaos: Algoritmo de Cifrado de Bloques con Transformaciones Caóticas Unidimensionales

C. E. Rojas-López, C. Cortés-Bazán, J. A. Martínez-Ñonthe y R. Vázquez-Medina, *Member, IEEE*

***Resumen*— Se presenta una aplicación informática denominada Cryptochaos, la cual es una implementación en Java de un algoritmo de cifrado de bloques. Cryptochaos opera con bloques de datos de 64 bits, que para su procesamiento se dividen en 8 sub-bloques, cada uno de 8 bits. El algoritmo de cifrado sigue la estructura desbalanceada de Kocarev y las funciones generadoras de ruido se construyen usando transformaciones caóticas unidimensionales. Estas transformaciones emplean un parámetro de control que les permite operar en la región caótica, la cual garantiza que cada sub-bloque de datos se mezcle con ruido pseudoaleatorio impredecible. Cryptochaos se ha evaluado empleando conceptos de la teoría de la información como son la entropía como medida de difusión en el proceso de cifrado, y la información mutua como medida de la relación que existe entre la entrada y la salida del cifrador. La aleatoriedad de la salida generada por Cryptochaos fue evaluada aplicando la batería de pruebas de aleatoriedad del Instituto Nacional de Estándares y Tecnología, NIST, por sus siglas en inglés (National Institute of Standards and Technology). Esta aplicación es de uso general para individuos y organizaciones, públicas o privadas, que requieran incorporar confidencialidad en su información sensible. El uso de esta herramienta asegura la privacidad de los datos que tenga almacenados en su computadora o que se envíen por correo electrónico. Con Cryptochaos, se verán beneficiadas las Aplicaciones Informáticas en la Industria Petrolera (AIP), ya evita que aquellos individuos que no son destinatarios o usuarios legítimos de la información protegida, recuperen la información original.**

***Índices*—Algoritmos de cifrado, Cifradores, Criptografía Caótica, Criptosistemas.**

Este trabajo fue apoyado por el Instituto Politécnico Nacional a través del financiamiento del proyecto de investigación institucional SIP 20101510.

C. E. Rojas-López, C. Cortés-Bazán, J. A. Martínez-Ñonthe y R. Vázquez-Medina, colaboran en la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacan del Instituto Politécnico Nacional, Av. Santa Ana 1000, Col. San Francisco Culhuacan, CP 04430 Coyoacán, D.F., México. (e-mail: [crojas, ccortesb, jmartinezn9800 y ruvazquez]@ipn.mx).

I. Introducción

ESTE trabajo ofrece una aplicación informática que permite el aseguramiento de la información, otorgando confidencialidad a la información a través del uso de un algoritmo de cifrado de bloques. Esta aplicación se ha denominado Cryptochaos debido a que emplea transformaciones caóticas unidimensionales para generar el ruido dentro de la estructura desbalanceada del algoritmo de cifrado de bloques que contiene en su interior. Inicialmente, el algoritmo de cifrado fue desarrollado en el intérprete de MatLab™ y se evaluó su efectividad de cifrado.

Debido a que con dicha versión se lograba afectar satisfactoriamente la semántica, la sintáctica y la estadística de los mensajes, para hacerlos incomprensibles para aquellos distintos a los destinatarios, se decidió implementar una versión en JAVA™, la cual optimizará el tiempo de cifrado y así, procesar archivos de diversos tamaños.

Cryptochaos se compara con algoritmos comerciales semejantes. Para ello, usa los conceptos de entropía e información mutua, de acuerdo a lo que sugiere la teoría de la información. En esta comparación se emplean archivos de referencia de distintos formatos y de diversos contenidos, los cuales son cifrados por Cryptochaos y los algoritmos comerciales similares. En cada caso, la salida se espera que sea muy parecida a una señal de ruido con una distribución estadística muy parecida a la uniforme. Por ello, se determina para cada caso que tan cercana está la entropía del mensaje cifrado del valor de entropía máxima esperado (8 para este caso), y que tan cercano está el valor de la información mutua del criptosistema del valor de información mutua mínima esperado (cero en este caso). Adicionalmente, se mide la aleatoriedad de los mensajes cifrados haciendo uso de la batería de pruebas de aleatoriedad NIST [3] y con ello, determinar si cumple con el criterio de criptosistema seguro de Shannon.

Debido al creciente uso de servicios que proporcionan alojamiento de información en equipos de cómputo

geográficamente distantes y descentralizados del dueño de la información (Cloud computing), la seguridad e integridad de dicha información se ve comprometida por lo que Cryptochaos proporciona un complemento a los servicios antes mencionados, de tal manera que la información no quede expuesta a los usos inadecuados de terceros o a un probable robo de información, tanto en servicios para dispositivos móviles (Smartphone, handlets, pda's, etc.), como para servicios que se ofrecen en Internet ya sea que tengan algún costo o se ofrezcan de manera gratuita (hosting).

II. Algoritmo Propuesto

El desarrollo del algoritmo planteado en este articulo toma como base lo expuesto por Ljupco Kocarev y Goce Jakimoski en 2001 [1]. En este algoritmo se define que el tamaño de bloque del mensaje original es de 64 bits de longitud (L= 8 bytes). Los bloques de datos quedan representados por $B_j=x_{j,0}, \ldots x_{j,7}$. De modo que, el cifrado consiste de r rondas de transformaciones caóticas idénticas y la clave de cifrado aplicadas sobre el bloque del mensaje original, obteniéndose $B_{j+1}=x_{j+1,0}, \ldots x_{j+1,7}$. como el bloque correspondiente de texto cifrado (Ver Fig. 1). De igual manera, el descifrado consiste de r rondas con la aplicación inversa de las transformaciones caóticas [2] y la clave de cifrado para obtener B_j a partir de B_{j+1} (Ver Fig. 2). Así, la función de cifrado está dada por la ecuación:

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

Cada ronda se compone de una red desbalanceada, que utiliza una subclave de 64 bits. Esta ronda se puede definir como un conjunto de transformaciones. Las funciones de cifrado quedan definidas con la siguiente ecuación:

$$\begin{aligned}
x_{i+1,2} &= x_{i,1} \oplus f_0(z_{i-1,0}) \\
x_{i+1,3} &= x_{i,2} \oplus f_1(x_{i,1} \oplus z_{i-1,1}) \\
x_{i+1,0} &= x_{i,3} \oplus f_2(x_{i,1} \oplus x_{i,2} \oplus z_{i-1,2}) \\
x_{i+1,1} &= x_{i,0} \oplus f_3(x_{i,1} \oplus x_{i,2} \oplus x_{i,3} \oplus z_{i-1,3} \\
&\vdots \\
x_{i,k+1} &= x_{i-1,k} \oplus f_{k-1}(x_{i-1,1}, \ldots, x_{i-1,k-1}, z_{i-1,k-1})
\end{aligned}$$

Los valores $x_{i,0}, x_{i,1}, x_{i,2}, \ldots x_{i,7}$ representan los bits del bloque de texto claro. Los valores $x_{j+1,0}, x_{j+1,1}, x_{j+1,2} \ldots x_{i+1,7}$ representan los bits del bloque de texto cifrado, obtenidos de la transformación expresada por la fig. 1.

Es importante mencionar que la transformación caótica unidimensional utilizada es caótica en su intervalo de definición (a, b) en los números reales. Sin embargo, para su aplicación en criptografía estas funciones deben escalarse y discretizarse, para que puedan utilizarse en el intervalo (0, 255) en los números enteros, ya que es en ese universo donde están definidos todos los archivos que se emplean digitalmente.

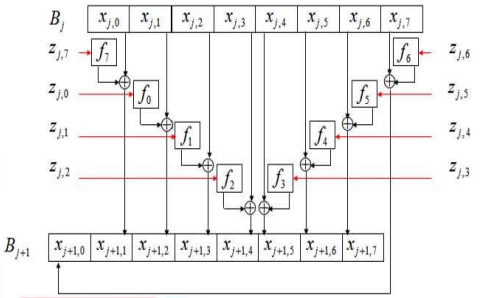


Fig. 1. Diagrama del proceso de cifrado usando las ocho variantes de la transformación caótica seleccionada.

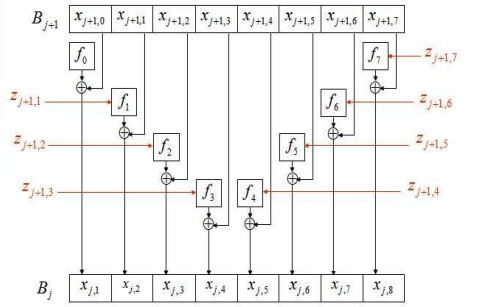


Fig. 2. Diagrama del proceso de descifrado usando las ocho variantes de la transformación caótica seleccionada.

III. Evaluación del Cifrador Propuesto

La evaluación del algoritmo se ha realizado usando fundamentalmente tres conceptos: la entropía de los mensajes cifrados, la información mutua del algoritmo de cifrado y la aleatoriedad de los mensajes cifrados [4]. Para esta evaluación se considera la comparación de Cryptochaos con algoritmos comerciales como AES (Advanced Encryption Standard), DES (Data Encryption Standard), 3-DES, BLOWFISH, IDEA y SKIPJACK. En esta evaluación se consideran 19 archivos

de referencia de distintos formatos y contenidos.

A. Resultados para la Entropía.

Desde el punto de vista de la Teoría de la Información, la entropía es la cantidad de información promedio que contiene un mensaje. La entropía se utiliza para estimar que tan lejos está el mensaje cifrado de ser un sistema equiprobable. De manera que, si la entropía del mensaje cifrado es máxima significa que el mensaje cifrado puede tener una distribución estadística uniforme. Esta condición busca una alta difusión y confusión y se considera que el sistema es llevado al estado de mayor incertidumbre. El valor más alto que puede tener la entropía máxima esperada es de $H(X)=8.0$, cuando se trabaja con un alfabeto de 256 caracteres. En este caso, el alfabeto es de 256 caracteres por que se está empleando el código ASCII extendido. La tabla 1 muestra los resultados obtenidos para la entropía de los criptosistemas Cryptochaos, AES, DES, 3-DES, BLOWFISH, IDEA y SKIPJACK. Todos ellos exhiben una entropía muy cercana a la entropía máxima. En la primera columna de la tabla se coloca el nombre del archivo, la segunda corresponde a la entropía del archivo original y la tercera a la entropía del mensaje cifrado con Cryptchaos. Las demás columnas contienen la entropía de los mensajes cifrados con los otros criptosistemas.

TABLA I.
VALORES DE ENTROPÍA DE LOS MENSAJES CIFRADOS CON CRYPTOCHAOS

ENTROPIA DE ARCHIVOS								
ARCHIVOS	ORIGINAL	CRYPTOCHAOS	AES	DES	TRIPLE DES	BLOWFISH	IDEA	SKIPJACK
TXT	5.08025076	**7.99913272**	7.99892139	7.99909278	7.99905598	7.99923657	7.99921334	7.99922055
DOC	5.81746799	**7.99953080**	7.99948508	7.99946659	7.99947703	7.99945794	7.99952896	7.99950037
RTF	3.5727449	**7.99994638**	7.99994668	7.99994254	7.99995439	7.99993928	7.99994439	7.99993752
PPT	6.68959154	**7.99964115**	7.99955763	7.99963587	7.9995855	7.9996457	7.99966255	7.99964606
XML	5.44878349	**7.99980765**	7.99981632	7.99979686	7.99983399	7.99982748	7.99980911	7.99981661
DOCPDF	7.51950803	**7.99924343**	7.99928482	7.99923789	7.99927246	7.99933132	7.99919534	7.99924372
RTFPDF	7.51971352	**7.99927204**	7.99923496	7.99933021	7.99924838	7.99916847	7.99917598	7.99924595
PPTPDF	7.61559334	**7.99942038**	7.99955395	7.9994796	7.99951086	7.99950854	7.99953136	7.99955929
XMLPDF	7.5188641	**7.99911358**	7.99930098	7.99926973	7.99923824	7.99930299	7.99921404	7.99932789
JPG	7.94327689	**7.99710115**	7.99711719	7.99725185	7.99733575	7.99707198	7.99736654	7.9970975
TIF	7.73252108	**7.99989501**	7.99988403	7.99989469	7.99989592	7.9998992	7.99989291	7.99989799
PNG	7.97247355	**7.99981321**	7.99983497	7.99981643	7.99981306	7.99984725	7.9998196	7.99983105
BMP	7.30181771	**7.99992567**	7.99991806	7.99992058	7.99992216	7.99992705	7.99992768	7.99991175
MP3	7.96341756	**7.99994476**	7.99994498	7.99994501	7.99995236	7.99995405	7.99994741	7.99994298
MP3-64	7.92657551	**7.99952798**	7.99961158	7.99956477	7.9995169	7.99957367	7.99953963	7.99959135
MP3-128	7.86608236	**7.99977726**	7.99973969	7.99978306	7.99977666	7.99978589	7.99977828	7.99978904
MP3-256	7.92829386	**7.99987753**	7.99987662	7.99987853	7.99989826	7.99987586	7.99989345	7.99990306
WAV	4.50805425	**7.99977176**	7.99978239	7.99981196	7.99979262	7.99980049	7.99977774	7.99976948
WMV	7.96269262	**7.99995671**	7.99994838	7.999955	7.99995531	7.99995973	7.99995738	7.99995576

B. Resultados de Información Mutua

La información mutua, I, mide la cantidad de información que aporta sobre una variable el conocimiento de otra. Se dice que un criptosistema es seguro si la cantidad de información que aporta el hecho de conocer el criptograma C sobre la entropía del texto plano M vale cero. Por lo tanto, se busca que $I(C,M)=0$

La Tabla II muestra la información mutua. La primera columna muestra el tipo de archivo, la segunda es la información mutua para el Cryptochaos. Las demás columnas contienen la información mutua obtenida con los criptosistemas AES, DES, 3DES, BLOWFISH, IDEA y SKIPJACK. De acuerdo a la definición dada por Shannon para un criptosistema seguro, el valor ideal de la información mutua vale cero.

TABLA II.
VALORES DE INFORMACIÓN MUTUA OBTENIDOS CON CRYPTOCHAOS.

INFORMACION MUTUA DE ARCHIVOS						
ARCHIVOS	CRYPTOCHAOS	AES	DES	TRIPLE DES	BLOWFISH	IDEA
TXT	**0.069406136**	0.07105238	0.070089371	0.06992893	0.071875815	0.069514086
DOC	**0.144914526**	0.143493435	0.144008712	0.143850224	0.143763558	0.144902271
RTF	**0.004713835**	0.004777655	0.004841628	0.004784422	0.004769446	0.004754917
PPT	**0.100982858**	0.10131626	0.10184379	0.101536803	0.10047334	0.101445367
XML	**0.017863721**	0.017772327	0.017728817	0.017855584	0.01787472	0.017775054
DOCPDF	**0.217351648**	0.216374212	0.217701716	0.21566141	0.216038249	0.217548086
RTFPDF	**0.215907287**	0.217307663	0.219649894	0.219028143	0.216440482	0.217967864
PPTPDF	**0.132192808**	0.130155637	0.13002241	0.128694849	0.12958761	0.130358914
XMLPDF	**0.215881141**	0.217243043	0.217784237	0.217891314	0.216965131	0.217193147
JPG	**0.801711875**	0.799733882	0.796887403	0.796647911	0.803417451	0.794975001
TIF	**0.025688735**	0.025991866	0.025697819	0.025706223	0.025854177	0.025923808
PNG	**0.044430993**	0.044889558	0.044619854	0.044635295	0.044722483	0.044672809
BMP	**0.020177536**	0.020112512	0.020089636	0.020236929	0.020071535	0.02008604
MP3	**0.01360542**	0.013652743	0.01393065	0.01373209	0.01382423	0.013837568
MP3-64	**0.116767545**	0.116947742	0.115861252	0.116429694	0.116345874	0.116386857
MP3-128	**0.056931304**	0.056921624	0.057249363	0.05728909	0.056438553	0.057561648
MP3-256	**0.028304061**	0.028508646	0.028337052	0.02818777	0.028493741	0.028282573
WAV	**0.045353809**	0.045158937	0.045113427	0.045163795	0.045210389	0.045191231
WMV	**0.01163626**	0.011569279	0.011632858	0.011737971	0.011713588	0.011713129

C. Pruebas del NIST

La Tabla III, muestra los resultados de aplicar el conjunto de pruebas estadísticas del NIST a un archivo, el cual se cifró con cada criptosistema antes mencionado. Para poder aplicar las pruebas, los archivos cifrados se convirtieron a su correspondiente valor binario, obteniendo así una cadena de 100 millones de bits. La Tabla III muestra los P-valores obtenidos de los mensajes cifrados cuando se usa Cryptochaos con distintas transformaciones caóticas unidimensionales (Tent, Logística y Senoidal) y se compara los P-valores del mensaje cifrado con AES.

TABLA III.
P-VALORES PARA LOS MENSAJES CIFRADOS CON CRYPTOCHAOS USANDO DISTINTAS TRANSFORMACIONES CAÓTICAS UNIDIMENSIONALES

PRUEBAS DE PSUDOALEATORIEDAD DEL NIST						
No.	Prueba	P-VALUE				REVISION
		CRYPTOCHAOS	MAPEO SENOIDAL	MAPEO LOGISTICO	AES	DES	
1	Frequency	0.5955	0.6371	0.2826	0.9463	0.0155	PASA
2	Block-Frequency	0.3669	0.162	0.3392	0.4559	0.7399	PASA
3	Cumulative-sums Forward	0.9357	0.924	0.8816	0.924	0.3504	PASA
	Cumulative-sums Reverse	0.7197	0.8165	0.2248	0.7791	0.07571	PASA
4	Runs	0.9357	0.1372	0.3369	0.9558	0.2368	PASA
5	Longest-Runs of Ones	0.6579	0.1537	0.3431	0.7597	0.6371	PASA
6	Rank	0.3505	0.1372	0.407	0.3504	0.319	PASA
7	Spectral fft	0.0428	0.0965	0.4599	0.0757	0.2896	PASA
8	Overlapping-templates	0.7782	0.7197	0.5341	0.3838	0.419	PASA
9	Non-Overlapping-template	0.4461	0.6165	0.7981	0.5749	0.3838	PASA
10	Universal	0.3345	0.1537	0.2133	0.6786	0.924	PASA
11	Aproximate Entropy	0.4012	0.3345	0.7177	0.7597	0.3504	PASA
12	Random-Excursions	0.5047	0.7197	0.9642	0.02818	0.2492	PASA
18	Random-Excursions-Varian	0.8789	0.7597	0.9114	0.1005	0.1453	PASA
14	serial	0.8669	0.9463	0.6982	0.6371	0.01559	PASA
15	linear-complexity	0.5955	0.8504	0.9642	0.5544	0.2133	PASA

D. Resultados de la velocidad de cifrado

Por último, la Tabla IV muestra los resultados de los tiempos de cifrado de Cryptochaos en su versión de intérprete

de MatLab™ y en la versión en desarrollada en JAVA™.

TABLA IV. TIEMPOS DE CIFRADO

TIEMPOS DE CIFRADO				
ARCHIVO	CIFRADOR PROPUESTO		AES	DES
	MATLAB	CRYPTOCHAOS		
TXT	00:00:44.000	00:00:05.000	00:00:00.048	00:00:00.064
DOC	00:01:59.000	00:00:09.000	00:00:00.084	00:00:00.063
RTF	03:08:22.000	00:01:22.000	00:00:00.464	00:00:00.540
PPT	00:03:52.000	00:00:12.000	00:00:00.079	00:00:00.115
XML	00:18:49.000	00:00:25.000	00:00:00.155	00:00:00.198
DOCPDF	00:00:51.000	00:00:06.000	00:00:00.056	00:00:00.039
RTFPDF	00:00:51.000	00:00:07.000	00:00:00.145	00:00:00.059
PPTPDF	00:02:13.000	00:00:10.000	00:00:00.096	00:00:00.077
XMLPDF	00:00:51.000	00:00:05.000	00:00:00.070	00:00:00.063
JPG	00:00:07.000	00:00:01.000	00:00:00.027	00:00:00.032
TIF	01:02:09.000	00:00:47.000	00:00:00.261	00:00:00.366
PNG	00:20:23.000	00:00:27.000	00:00:00.138	00:00:00.225
BMP	01:34:55.000	00:00:59.000	00:00:00.423	00:00:00.429
MP3	03:35:01.000	00:01:28.000	00:00:00.541	00:00:00.643
MP3-64KB	00:02:44.000	00:00:11.000	00:00:00.075	00:00:00.161
MP3-128KB	00:12:18.000	00:00:21.000	00:00:00.144	00:00:00.127
MP3-256KB	00:51:02.000	00:00:42.000	00:00:00.236	00:00:00.409
WAV	00:12:34.000	00:00:21.000	00:00:00.148	00:00:00.166
WMV	04:59:25.000	00:01:41.000	00:00:00.499	00:00:00.686

IV. IMPLEMENTACIÓN EN JAVA

Dado que los tiempos obtenidos en la aplicación utilizando el intérprete de MatLab™, fueron relativamente largos, como se puede apreciar en la Tabla IV, se procedió a realizar la implementación del sistema utilizando el lenguaje de programación JAVA™. Los motivos de esta decisión fueron los que a continuación se enumeran.

1. Reducción de los tiempos en los procesos de cifrado y descifrado.
2. Portabilidad de la aplicación en diferentes sistemas operativos.
3. Hacer la aplicación 100% parametrizable.
4. Optimizar procesos en los algoritmos involucrados.
5. Implementación de una interfaz gráfica de usuario, para el fácil y rápido uso de la aplicación.

Cabe señalar que dicha implementación se realizó íntegra, cada algoritmo, método y procedimiento, fueron trasladados a JAVA™ tal y como se encontraban en su versión de MatLab™. Dado que en JAVA™ algunas funciones son implementadas de forma nativa, se pudo reducir el tamaño y complejidad de los códigos fuente, optimizando así, muchos de los procesos. Se explotaron al máximo las capacidades de JAVA™, al hacer el correcto uso de las clases y paquetes, el sistemas obtuvo una estructura organizada y completamente documentada. Se utilizaron hilos de proceso, los cuales permiten a la aplicación realizar varias tareas a la vez de manera concurrente, con lo que se puede cifrar y descifrar en un hilo de proceso y mantener la interfaz gráfica en otro, evitando así bloqueos inesperados en la aplicación.

A continuación se muestran algunas capturas de pantalla concernientes a la ejecución de la aplicación.

Fig. 3. Pantalla principal desde donde se puede cifrar o descifrar un archivo.

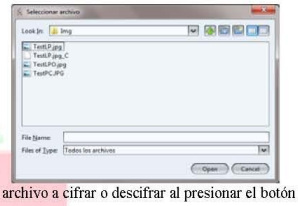


Fig. 4. Busca el archivo a cifrar o descifrar al presionar el botón que buscar.

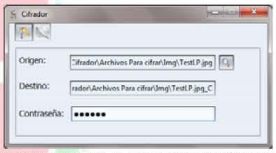


Fig. 5. De forma automática se llena el campo "Destino" y se procede a introducir una contraseña, para posteriormente cifrar el archivo.

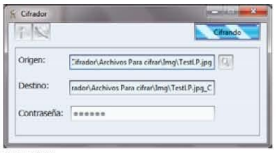


Fig. 6. Cifrado en proceso.

Fig. 7. Proceso terminado.

El proceso de descifrado muestra las mismas pantallas, al presionar el botón descifrar.

Los requerimientos del sistema en Windows son: un procesador Pentium 166 MHz o superior con un mínimo de 125 MB de espacio libre en el disco y un mínimo de 32 MB de RAM. Y los paquetes y las plataformas con que se desarrolló con las siguientes: Sun-java6-jdk (versión 1.6.0_10), Sun-java6-jre y NetBeans IDE 6.1

V. ¿POR QUÉ UTILIZAR CRYPTOCHAOS?

Cryptochaos se ha desarrollado considerando la norma internacional IEEE 829, la cual se refiere a la planificación y ejecución de pruebas del sistema.

Por otro lado, las actuales técnicas criptográficas normalmente se basan en la teoría de números o en algoritmos algebráicos. **La teoría del caos** es otro paradigma que parece prometedor en aplicaciones criptográficas. El caos es una rama del campo de la dinámica no lineal, ha sido ampliamente estudiado y ha encontrado un sin número de aplicaciones en diferentes áreas de la ciencia. Por ejemplo, en la economía, al estudiar el comportamiento de los mercados financieros [5], la medicina, en el estudio del sistema inmunitario humano [6], la biología, en el estudio de encimas y hormonas sujetas a la dinámica caótica [7], etc. Específicamente cuando se usan transformaciones caóticas unidimensionales, la implementación en criptosistemas es sencilla, y resultan sistemas efectivos, ya que se aprovechan dos condiciones importantes que dan robustez al sistema. Estas condiciones son la sensibilidad ante condiciones iniciales y la propiedad de mezclado.

Un gran número de aplicaciones en sistemas reales se desarrollan y estudian con base en sistemas dinámicos y teoría del caos; un ejemplo de estos sistemas son los osciladores caóticos [8]. El comportamiento caótico de un sistema no lineal 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.

Desde mediados de los años 90, se han presentado trabajos relacionados con la aplicación de la teoría del caos a la generación de secuencias cifrantes como podemos ver en [9-17]. Estas referencias muestran que está vigente el uso de la teoría del caos para proponer soluciones tecnológicas que ofrezcan la evaluación y diseño de comunicaciones seguras en secuencias generadoras de ruido para cifradores dentro de la criptografía.

VI. CONCLUSIONES

En este trabajo se muestra la implementación general y el comportamiento de Cryptochaos como un cifrador caótico de bloques. Cryptochaos puede utilizar distintas transformaciones caóticas unidimensionales. Aquí no se discute la ventaja de usar una u otra transformación. Sin embargo, por la experiencia que hemos tenido con el trabajo con estas funciones, aquellas que son lineales a tramos son muy buenas alternativas (como lo son las transformaciones de Tent y Bernoulli), ya que no poseen islas de estabilidad que llevarían a que los generadores de ruido usados se comportaran como generadores de señales periódicas. Esta última condición es indeseable dentro de un cifrador.

La mejora en la velocidad de procesamiento se logró con una implementación en JAVA™. Las transformaciones usadas en los generadores de ruido son en realidad aproximaciones a las transformaciones caóticas unidimensionales, ya que se usan las versiones discretas de aquellas transformaciones que en su versión contínua son efectivamente caóticas. La aportación de este trabajo radica en estudiar el comportamiento de dichas aproximaciones usando herramientas y conceptos fundamentales de la Mecánica Estadística. Se determinó que escalando y discretizando dichas transformaciones y usando una precisión de 8 bits en los números que las alimentan es posible alcanzar el comportamiento caótico que produce señales de ruido con una buena distribución estadística que pueda ser usada en aplicaciones criptográficas.

Para evaluar a Cryptochaos se usaron conceptos de Teoría de la información, y de los resultados se puede concluir que la entropía de los mensajes cifrados está altamente relacionada con su distribución estadística. De manera que, si la entropía es cercana a la entropía máxima, la distribución estadística del mensaje cifrado es muy parecida a una distribución estadística uniforme.

Se valoró la información mutua de Cryptochaos usando distintas secuencias provenientes de diversos mensajes en texto claro. Con los resultados obtenidos se puede concluir que la información mutua para Cryptochaos es muy cercana a cero, buscando apegarse al criterio de criptosistema seguro de Shannon. Lo que significa que el mensaje cifrado no posee evidencia del mensaje original después del proceso de cifrado.

Para la planificación y ejecución de pruebas del sistema se utilizó el estándar IEEE 829, el cual permite tener un control estricto del desarrollo de aplicaciones, así como los informes correspondientes derivados de cada prueba. Debido a la naturaleza de esta aplicación, el uso de este estándar fue de suma importancia para garantizar el correcto funcionamiento de este.

VII. REFERENCIAS

[1] L. Kocarev, Goce Jakimoski "Chaos and Cryptography: Block Encryption Ciphers Based on Chaotic Maps". IEEE Trans. on Circuits and Systems, 2001. Vol. 48(2)

[2] A. Tsuneda, K. Eguchi and T. Inoue "Design of Chaotic Binary Sequences with Good Statistical Properties based on Piecewise Linear into Maps", IEEE Transactions on Circuits and Systems.I: Regular papers, Vol 52, No. 2, February 2005, 454-462.

[3] NIST. Federal Information Processing Standards Publication (FIPS 140-1). Security requirements for cryptographic modules; 1994.
[4] R.A. Rueppel: "Analysis and Design of Stream Ciphers", Springer Verlag, (1986).
[5] Nieto de Alba, Ubaldo: "Predicción y Caos en Economía" Universidad Complutense. [2008]
[6] A.L. Goldberg: "Caos y fractales en la fisiología Humana," Investigación y Ciencia, Vol. 163. [2008]
[7] May, R. M.: "Theorical Ecology: principles and applications" Blackwell Scientific Publishers [1976].
[8] B. Rubén, C. Isaac, Campos. Eric: "Transmisión y Recepción de Voz Empleando Caos," Encuentro de Investigación en Ingeniería Eléctrica, Zacatecas, Abril 5-7. Facultad de ciencias, Departamento de Fisico Matemáticas Universidad Autónoma de San Luis Potosí. [2006]
[9] A. Tefas, A. Nikolaidis, N. Nikolaidis, V. Solachidis, S. Tsekeridou, I. Pitas:"Statistical analysis of Markov chaotic sequences for watermarking applications", Department of Informatic in Aristotle University of Thessaloniki, Grecia, 2001.
[10] N. Masuda and K. Aihara: "Cryptosystems with discretized chaotic maps", IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications, Vol 49, No. 1, January 2002, pp. 28-40
[11] S. Tsekeridou, V. Solachidis, N. Nikolaidis, a. Nikolaidis, A. Tefas and I. Pitas: "Theoretic investigation on the use of watermark signals derived from Bernoulli chaotic sequences", Department of Informatic in Aristotle University of Thessaloniki, Grecia, 2003.
[12] Z. Li, K. Li, C. Wen and Y. Chai Soh: "A new chaotic secure communication system", IEEE Transactions on Communications, Vol 51, No. 8, August 2003, pp. 1306-1312.
[13] N. Sajeeth Philip and K. Babu Joshep: "Chaos for stream cipher", Cochin University of Science and Technology, Kochi, India, 2004
[14] J. Szczepanski, J. M. Amigó, T. Michalek and L. Kocarev: "Cryptographically secure substitutions based on the approximation of mixing maps", IEEE Transactions on Circuits and Systems.I: Regular papers, vol 52, no. 2, February 2005, pp. 443-453.
[15] A. Tsuneda: "Design of binary sequences with tunable exponential autocorrelations and run statistics based on one-dimensional chaotic maps", IEEE Transactions on Circuits and Systems- I: Regular papers, vol 52, no. 2, February 2005, pp. 454-462.
[16] Ali Kanso and Nejib Smaoui, "Logistic chaotic maps for binary numbers generations", Chaos, Solutions & Fractals, Volume 40, Issue 5, 15 June 2009, Pages 2557-2568.
[17] Beker H, Piper F., "Cipher systems: the protection of communications". New York: van Nostrand Reinhold; 1982.

VIII. BIOGRAFÍAS

César Enrique Rojas López. Nació en Tehuantepec, Oaxaca. Recibió el título de Licenciado en Informática en el año 2000, por el Instituto Tecnológico del Istmo. Es profesor de tiempo completo del Instituto Politécnico Nacional, adscrito a la Escuela Superior de Ingeniería Mecánica y Eléctrica, ESIME, Unidad Culhuacan. Actualmente se encuentra realizando el tercer semestre de la Maestría en Ciencias en Ingeniería en Microelectrónica con especialidad en Seguridad Informática, en la Sección de Estudios de Posgrado e Investigación de la propia ESIME Culhuacan. Sus áreas de interés son la criptografía, la Seguridad Informática, Computación y las Comunicaciones. (crojas@ipn.mx).

Carlos Cortés Bazán. Egresado del Tecnológico de Estudios Superiores del Oriente del Estado de México como Ingeniero en sistemas Computacionales con Mención Honorífica (2006). Participó en el desarrollo de diversos sistemas cliente/servidor, así como sistemas web para el Área Central del Instituto Politécnico Nacional. Es Profesor de tiempo completo del Instituto Politécnico Nacional, adscrito a la Escuela Superior de Ingeniería Mecánica y Eléctrica, ESIME, Unidad Culhuacan. Sus áreas de interés son el desarrollo de sistemas, cómputo forense, la criptografía y el análisis de algoritmos. Actualmente se encuentra preparando su ingreso a la Maestría en Seguridad Informática y tecnologías de la Información en la Sección de Estudios de Posgrado e Investigación en la ESIME Culhuacan. (ccortesb@ipn.mx).

Jorge Alberto Martínez Ñonthe. Recibió el título de Ingeniero en Comunicaciones y Electrónica por el Instituto Politécnico Nacional (2005). Obtuvo el Grado de Maestro en Ciencias de Ingeniería en Microelectrónica en la Sección de Estudios de Posgrado e Investigación en la ESIME Unidad Culhuacan del I.P.N (2008), actualmente cursa el tercer semestre del Doctorado en Comunicaciones y Electrónica de la misma sección. Desde 2005 ha trabajado en infraestructura de comunicaciones e informática. Sus áreas de interés son la Seguridad Informática, Computación, Comunicaciones y Auditoria informática. (jmartinezn9800@ipn.mx).

Rubén Vázquez Medina, Member, IEEE . Nació en la Ciudad de México en 1966. Recibió el título de Ingeniero en Electrónica especialidad en Comunicaciones en 1988 en la Universidad Autónoma Metropolitana Unidad Iztapalapa y el grado de Maestro en Ciencias especialidad en Ingeniería Eléctrica opción en Telecomunicaciones en Septiembre de 1991 en el Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional. Obtuvo el grado de Doctor en Ciencias en la Universidad Autónoma Metropolitana Unidad Iztapalapa en Octubre de 2008. Fue jefe de la Sección de Estudios de Posgrado e Investigación de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhucan del 28 de Marzo del 2003 al 17 de agosto del 2006. Actualmente se desempeña como profesor en la Sección de Estudios de Posgrado de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacan del IPN, dentro de los programas de posgrado en Maestría en Ciencias de Ingeniería en Microelectrónica, Maestría en Ingeniería en Seguridad Informática y Tecnologías de la Información y en el Doctorado en Comunicaciones y Electrónica. Sus áreas de interés son la criptografía, la esteganografía, la informática forense y la forensia digital. (ruvazquez@ipn.mx).

CP-38
PON 86

Cifrador caótico de bloques Tent con criterio de autenticación y precisión de 16 bits

C. E. Rojas-López, J. A. Martínez-Ñonthe, R. Vázquez-Medina *Member, IEEE*

Instituto Politécnico Nacional
SEPI ESIME Culhuacan
Av. Santa Anna No. 1000 Col. San Francisco Culhuacan Delegación Coyoacan,
04430 México D. F.

Email: {crojas, jmartinezn9800,ruvazquez} @ipn.mx

1. Resumen

Se presenta la construcción y evaluación de un criptosistema de bloques basado en la transformación caótica unidimensional "tent". El algoritmo propuesto opera como cifrador de 64 bits, con 4 bloques y cada uno de estos con una precisión de 16 bits; la finalidad de cifrar con 16 bits es porque se aproxima mejor a la transformación original, mejorando de esta forma la densidad y distribución. En este trabajo se incorporó un criterio de autenticación que sirve como módulo de seguridad y ayuda a verificar la autenticación de la clave de descifrado, así como la autenticación del número de rondas utilizadas en el cifrado. El criptosistema se ha evaluado y comparado empleando conceptos de la teoría de la información como son la entropía como medida de difusión en el proceso de cifrado, y la información mutua como medida de la relación que existe entre la entrada y la salida del cifrador. La aleatoriedad de la salida del criptosistema fue evaluada con la batería de pruebas del NIST.

2. Introducción

En este trabajo se plantea el uso de una transformación caótica unidimensional denominada "tent" que sirve como función de transformación al momento de cifrar un archivo y que se incorpora dentro de la estructura expuesta por Ljupco Kocarev y Goce Jakimoski[1].

ROC&C'2010 – **CP-38** PONENCIA RECOMENDADA POR EL **COMITÉ DE COMPUTACION** DEL **IEEE SECCIÓN MÉXICO** Y PRESENTADA EN LA **REUNIÓN DE OTOÑO, ROC&C'2010**, ACAPULCO, GRO., DEL 28 DE NOVIEMBRE AL 4 DE DICIEMBRE DEL 2010.

El desarrollo de este trabajo contempla utilizar un modulo de seguridad que nos permita validar la autenticidad del proceso de cifrado y descifrado de nuestro cifrador, en este punto se realiza una concatenación de la clave de cifrado y el número de rondas de cifrado al mensaje cifrado y una verificación de la clave de cifrado y el número de rondas de cifrado antes de proceder al descifrado del mensaje. Además se realiza un ajuste al momento de trabajar los bloques del mensaje original de 64 bits a cifrar y descifrar utilizando una longitud de bloque con precisión de 16 bits. Con este ajuste es necesario realizar un escalamiento y discretización adecuado definido en el intervalo de (0,65535) de los enteros. La evaluación de nuestro cifrador usa los conceptos de entropía e información mutua, de acuerdo a lo que sugiere la teoría de la información [2]. En esta comparación se emplean archivos de referencia de distintos formatos y de diversos contenidos, los cuales son cifrados. En cada caso, la salida se espera que sea muy parecida a una señal de ruido con una distribución estadística muy parecida a la uniforme. Por ello, se determina para cada caso que tan cercana está la entropía del mensaje cifrado del valor de entropía máxima esperado (8 para este caso), y que tan cercano está el valor de la información mutua del criptosistema del valor de información mutua mínima esperado (cero en este caso). Adicionalmente, se mide la aleatoriedad de los mensajes cifrados haciendo uso de la batería de pruebas de aleatoriedad NIST [3] y con ello, determinar si cumple con el criterio de criptosistema seguro de Shannon.

3. Transformación Tent

La transformación caótica "tent" [4], está definida por la siguiente ecuación (1).

$$x_{n+1} = f(x_n) = r\left(1 - 2\left|x_n - \frac{1}{2}\right|\right) \qquad (1)$$

Donde r es el parámetro de control, y x_n tiene valores dentro del intervalo $[0,1]$. Para la definición anterior y para nuestros fines en el cifrador caótico "tent" utiliza la siguiente función, definida por la ecuación (2).

$$f:[0,1]\rightarrow[(1-\mu)/2,(1+\mu)/2] \quad (2)$$

con $0<\mu\leq 1$ en general. Para la implementación de esta transformación en el cifrador, se requiere de una forma equivalente que nos permita representar a la transformación "tent" para nuestros fines y para poder expresarla nos apoyamos de la ecuación (3).

$$f(x)=\begin{Bmatrix} 2\mu x_n+\dfrac{(1-\mu)}{2};.....0\leq x_n\leq\dfrac{1}{2} \\ 2\mu(1-x_n)+\dfrac{(1-\mu)}{2};...\dfrac{1}{2}<x_n\leq 1 \end{Bmatrix} \quad (3)$$

3.1 Escalamiento y discretización

Para realizar el escalamiento y discretización partimos de la ecuación (3), esta parte se resume a continuación:

La trasformación "tent" es escalada de tal manera que los valores de entrada y salida de la trasformación se encuentra definida en el intervalo $[0,65535]$ de los reales y no en un intervalo (0,1) como originalmente se encuentra definido, ver ecuación (4).

$$x_n\in(0,1)\rightarrow x_n\in[0,65535]\rightarrow R \quad (4)$$

Posteriormente, la trasformación "tent" escalada es ahora discretizada, de tal manera que los valores de la trasformación estén en el intervalo [0,65535] de los enteros. Ver ecuación (5).

$$x_n^{\cdot}\in[0,65535]\rightarrow \mathrm{I} \quad (5)$$

Ahora de esta forma la transformación "tent" escalada y discretizada queda definida por la ecuación (6).

$$f(x)=\begin{cases} floor\left[2\mu x_n+\left(65535*\left(\dfrac{1-\mu}{2}\right)\right)\right] \\ \\ floor\left[-\mu\left(2\left(x_n-65535\right)\right)+\left(65535*\left(\dfrac{1-\mu}{2}\right)\right)\right] \end{cases} \quad (6)$$

3.2 Precisión de 16 bits

Cuando nosotros referenciamos precisión de 16 bits es porque el algoritmo propuesto funcionara como cifrador de 64 bits, pero con 4 bloques y cada uno de estos con una precisión de 16 bits; La finalidad de cifrar con 16 bits es porque con esta precisión se aproxima mejor a la transformación original, mejorando la densidad y distribución. Por lo que la transformación caótica "tent" se discretizó y normalizó al intervalo de los enteros en $[0,65535]$. Pero con valores de $2^{16}=65536$. Esto hace que sea más precisa. Cuando se compara con el valor de $2^8=256$ es menos denso y tiene pocos valores de ocurrencia, ver figura 1.

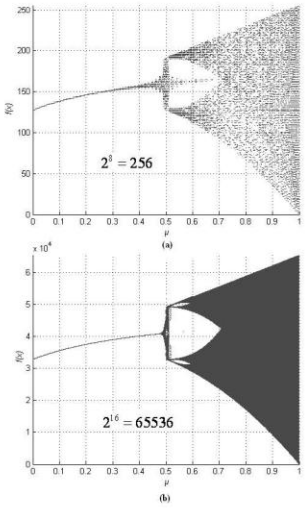


Figura 1. Comparativa del diagrama de bifurcación (a) cuando utilizamos una precisión de 8 bits y (b) cuando utilizamos una precisión de 16 bits.

4. Desarrollo

4.1 Algoritmo propuesto

El desarrollo del algoritmo planteado en este articulo toma como base lo expuesto por Ljupco Kocarev y Goce Jakimoski en 2001 [1]. En este algoritmo se define que el tamaño de bloque del mensaje original es de 64 bits de longitud (L= 16 bytes). Los bloques de datos quedan representados por $B_i=x_{i,0}, \ldots x_{i,4}$. De modo que, el cifrado consiste de r rondas junto con la transformación caótica "tent" y la clave de cifrado aplicadas sobre el bloque

del mensaje original, obteniéndose $B_{i+1}=x_{i+1,0}$, ... $x_{i+1,4}$ como el bloque correspondiente de texto cifrado (Ver Figura 2). De igual manera, el descifrado consiste de r rondas con la aplicación inversa junto con la transformación caótica "tent" y la clave de cifrado para obtener B_i a partir de B_{i+1} (Ver Figura 3). Así, la función de cifrado está dada por la ecuación (7):

$$x_{i,k+1} = x_{i-1,k} \oplus f_{k-1}\left(x_{i-1,1}, \cdots, x_{i-1,k-1}, z_{i-1,k-1}\right) \qquad (7)$$

Cada ronda se compone de una red desbalanceada, que utiliza una subclave de 64 bits. Esta ronda se puede definir como un conjunto de transformaciones. Las funciones de cifrado quedan definidas con la siguiente ecuación (8):

$$\begin{aligned} x_{i+1,2} &= x_{i,1} \oplus f_0 \\ x_{i+1,3} &= x_{i,2} \oplus f_1(x_{i,1} \oplus z_{i-1,1}) \\ x_{i+1,0} &= x_{i,3} \oplus f_2(x_{i,1} \oplus x_{i,2} \oplus z_{i-1,2}) \\ x_{i+1,1} &= x_{i,0} \oplus f_3(x_{i,1} \oplus x_{i,2} \oplus x_{i,3} \oplus z_{i-1,3}) \\ &\vdots \\ x_{i,k+1} &= x_{i-1,k} \oplus f_{k-1}(x_{i-1,1}, \ldots, x_{i-1,k-1}, z_{i-1,k-1}) \end{aligned} \qquad (8)$$

Los valores $x_{i,0}$, $x_{i,1}$, $x_{i,2}$, ... $x_{i,4}$ representan los bits del bloque de texto claro. Los valores $x_{i+1,0}$, $x_{i+1,1}$, $x_{i+1,2}$, ... $x_{j+1,4}$ representan los bits del bloque de texto cifrado, obtenidos de la transformación expresada por la figura 1. Es importante mencionar que la transformación caótica unidimensional "tent" es caótica en su intervalo de definición (a, b) en los números reales. Sin embargo, para su aplicación en criptografía esta función debe escalarse y discretizarse, para que puedan utilizarse en el intervalo (0, 65535) en los números enteros.

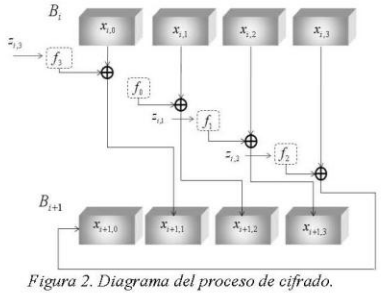


Figura 2. Diagrama del proceso de cifrado.

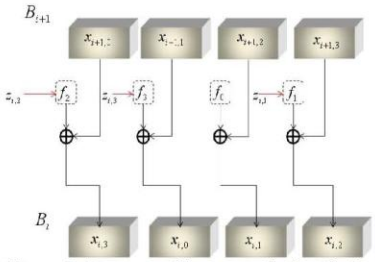


Figura 3. Diagrama del proceso de descifrado.

4.2 Proceso de generación de subclaves

Este proceso se utiliza para generar las subclaves. Resulta útil esta implementación para generar mayor confusión y densidad para el mensaje a cifrar, ya que al ir cifrando cada bloque la clave estará cambiando y de esta forma no permanecerá la clave inicial. A continuación describimos dicho proceso:

A partir de la clave principal se le aplica la función SHA-512, la cual devuelve una palabra en hexadecimal de 128 caracteres (512bits), esta palabra se convierte en un equivalente decimal y se divide en bloques de 64 bits que representa una subclave a utilizar en determinada ronda del cifrado o descifrado. Para generar el siguiente grupo de subclave se utiliza el bloque resultante cifrado junto con el bloque a cifrar, ha estos 2 bloques se le aplica nuevamente la función SHA-512 a la palabra resultante del anterior grupo de subclaves. El proceso se muestra de forma sencilla en la figura 4.

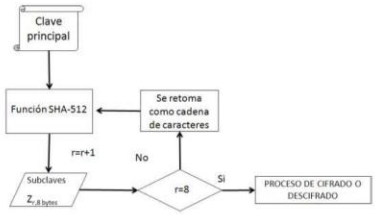


Figura 4. Diagrama del proceso de generación de subclaves.

5. Criterio de autenticación

Además de utilizar un proceso de generación de subclaves en cada ronda. Se procedió a realizar un módulo de seguridad que nos ayuda a verificar la autenticación de la clave de descifrado, así como la autenticación del número de rondas utilizadas en el cifrado, y en caso de que no sean similares cualquiera de las dos, salir del proceso de descifrado sin realizar éste, ya que originalmente se estaba

realizando el mencionado proceso, y aunque el resultado del descifrado no era en texto claro, un criptoanalista haciendo un análisis de los resultados, podría llegar a obtener el texto plano. Para realizar el módulo de seguridad, se añadió en el mensaje cifrado C, la clave de cifrado y el número de rondas, dando, evidentemente, un archivo más grande que el original.

5.1 Concatenación de la clave de cifrado y el número de rondas de cifrado al mensaje cifrado

Para realizar este proceso, una vez obtenido el mensaje cifrado, se procede a aplicarle a éste la función hash SHA-256, h(C); así mismo, se le aplica la misma función hash a la clave de cifrado, h(k). Al obtener los dos valores, éstos se concatenan, para aplicar, a este nuevo valor, la función hash SHA-256. Al resultado, que es hexadecimal, se convierte a decimal, y se procede a cifrarlo con la transformación unidimensional tent, previamente explicada, para posteriormente, el resultado de este cifrado, concatenarlo al mensaje cifrado. Se hace lo mismo para el número de rondas, cambiando h(k), por h(r). Ver figura 5.

Figura 5. Diagrama del proceso de concatenación de la clave de cifrado y el número de rondas de cifrado, al mensaje cifrado.

5.2 Verificación de la clave de cifrado y el número de rondas de cifrado antes de proceder al descifrado del mensaje

Para realizar el proceso de verificación de la clave de descifrado y el número de rondas, contra las que fue cifrado el archivo, es necesario descomponer el mensaje cifrado en sus tres partes, es decir, el texto cifrado original, y las concatenaciones de la clave de cifrado y el número de rondas; una vez que se cuenta con esto, a la parte correspondiente al texto cifrado se le aplica la función hash SHA-256, h(C), así como a la clave de descifrado h(k), y una vez con estos resultados, se concatenan, y se le aplica la función hash SHA-256. El resultado lo comparamos contra el valor concatenado de la clave de cifrado que se obtiene al descifrar la parte correspondiente a la concatenación de la clave de cifrado; si son iguales, se repite el proceso, pero ahora con los datos del número de rondas k(r). Si no son iguales, se aborta el descifrado, sin que se entregué ningún resultado, para prever que éste pueda ser objeto de un criptoanálisis. Ver figura 6.

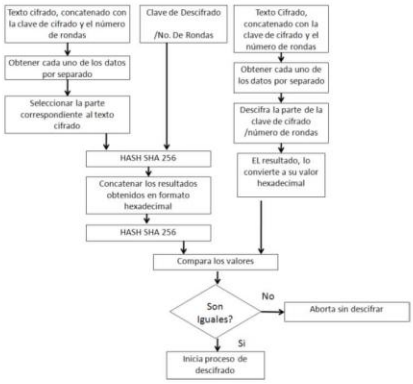


Figura 6. Diagrama del proceso de verificación de la clave de descifrado y el número de rondas contra las que fue cifrado el archivo.

6. Evaluación del cifrador propuesto

La evaluación del algoritmo se ha realizado usando fundamentalmente tres conceptos: la entropía de los mensajes cifrados, la información mutua del algoritmo de cifrado y la aleatoriedad de los mensajes cifrados [4].

A. Resultados de la Entropía

Desde el punto de vista de la Teoría de la Información, la entropía es la cantidad de información promedio que contiene un mensaje. La entropía se utiliza para estimar que tan lejos está el mensaje cifrado de ser un sistema equiprobable. De manera que, si la entropía del mensaje cifrado es máxima significa que el mensaje cifrado puede tener una distribución estadística uniforme. Esta condición busca una alta difusión y confusión y se considera que el sistema es llevado al estado de mayor incertidumbre. El valor más alto que puede tener la entropía máxima esperada es de $H(X)=8.0$,

cuando se trabaja con un alfabeto de 256 caracteres. En este caso, el alfabeto es de 256 caracteres por que se está empleando el código ASCII extendido. La tabla 1 muestra los resultados obtenidos para la entropía del cifrador caótico tent, AES, DES, 3-DES, BLOWFISH, IDEA y SKIPJACK. Todos ellos exhiben una entropía muy cercana a la entropía máxima. En la primera columna de la tabla se coloca el nombre del archivo, la segunda corresponde a la entropía del archivo original y la tercera a la entropía del mensaje cifrado con la transformación tent. Las demás columnas contienen la entropía de los mensajes cifrados con los otros cifradores.

Tabla 1. Valores de Entropía del cifrador caótico de bloques "tent"

ENTROPIA DE ARCHIVOS								
ARCHIVO	ORIGINAL	CIFRADOR CAOTICO "TENT" CON PGSC	AES	DES	TRIPLE DES	BLOWFISH	IDEA	SKIPJACK
TXT	5.0802508	**7.999173637**	7.9989214	7.9990928	7.9990560	7.9992366	7.9992333	7.9992205
DOC	5.8174680	**7.999426499**	7.9994851	7.9994666	7.9994770	7.9994579	7.9995290	7.9995004
RTF	3.5727489	**7.999939016**	7.9999467	7.9999425	7.9999544	7.9999393	7.9999444	7.9999375
PPT	6.6895915	**7.999671958**	7.9995576	7.9996359	7.9995855	7.9996457	7.9996626	7.9996461
XML	5.4487835	**7.999824941**	7.9998163	7.9997969	7.9998340	7.9998275	7.9998091	7.9998166
DOCPDF	7.5195080	**7.999229928**	7.9992848	7.9992379	7.9992725	7.9993313	7.9991953	7.9992437
RTFPDF	7.5197135	**7.999185492**	7.9992350	7.9993302	7.9992484	7.9991685	7.9991760	7.9992468
PPTPDF	7.6155933	**7.999516563**	7.9995539	7.9994796	7.9995109	7.9995085	7.9995314	7.9995583
XMLPDF	7.5188643	**7.999227461**	7.9993010	7.9992697	7.9992382	7.9993030	7.9992340	7.9993279
JPG	7.9432769	**7.997262142**	7.9971172	7.9972515	7.9973558	7.9970720	7.9973665	7.9970975
TIF	7.7325211	**7.999901287**	7.9998840	7.9998947	7.9998959	7.9998992	7.9998929	7.9998980
PNG	7.9724726	**7.999842464**	7.9998350	7.9998164	7.9998131	7.9998473	7.9998296	7.9998319
BMP	7.3018177	**7.999925329**	7.9999181	7.9999206	7.9999222	7.9999270	7.9999277	7.9999119
MP3	7.9634176	**7.999948501**	7.9999450	7.9999450	7.9999524	7.9999541	7.9999474	7.9999430
MP3-64Kb	7.9265755	**7.999493868**	7.9996116	7.9995648	7.9995169	7.9995737	7.9995396	7.9995913
MP3-128Kb	7.8660824	**7.999790245**	7.9997397	7.9997831	7.9997767	7.9997859	7.9997783	7.9997890
MP3-256Kb	7.9282939	**7.999882543**	7.9998766	7.9998785	7.9998983	7.9998759	7.9998935	7.9999031
WAV	4.5080543	**7.999781582**	7.9997824	7.9998120	7.9997926	7.9998005	7.9997777	7.9997695
WMV	7.9626926	**7.999950882**	7.9999484	7.9999550	7.9999553	7.9999597	7.9999574	7.9999558

B. Resultados de Información Mutua

La información mutua mide la cantidad de información que aporta sobre una variable el conocimiento de otra. Se dice que un criptosistema es seguro si la cantidad de información que aporta el hecho de conocer el criptograma C sobre la entropía del texto plano M vale cero. La siguiente ecuación (9) define el concepto de Información mutua.

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

Se busca que:

$$I(C,M) = 0 \qquad (9)$$

La tabla 2, muestra la información mutua, la primera columna muestra el tipo de archivo, la segunda es la información mutua con el criptosistema cuando se utiliza el esquema de generación de claves que usa la función HASH SHA-512 descrita con anterioridad, las demás columnas son la información mutua con los criptosistemas AES, DES, 3DES, BLOWFISH, IDEA y SKIPJACK. De acuerdo a la definición dada por Shannon para un criptosistema seguro, el valor ideal de la información mutua vale cero.

Tabla 2. Valores de Información Mutua del cifrador caótico de bloques "tent"

INFORMACION MUTUA DE ARCHIVOS							
ARCHIVO	CIFRADOR CAOTICO "TENT" CON PGSC	AES	DES	TRIPLE DES	BLOWFISH	IDEA	SKIPJACK
TXT	**0.069618783**	0.0710524	0.0700894	0.0699289	0.0718758	0.0695141	0.0714406
DOC	**0.144927532**	0.1434934	0.1440087	0.1438502	0.1437636	0.1449023	0.1438782
RTF	**0.004785542**	0.0047777	0.0048416	0.0047844	0.0047694	0.0047549	0.0048623
PPT	**0.100683255**	0.1013163	0.1018438	0.1015368	0.1004735	0.1014454	0.1017251
XML	**0.017484572**	0.0177723	0.0177288	0.0178956	0.0178747	0.0177751	0.0178305
DOCPDF	**0.215586135**	0.2165742	0.2177017	0.2156614	0.2160382	0.2175431	0.2174396
RTFPDF	**0.217109438**	0.2173077	0.2196499	0.2180281	0.2164405	0.2179679	0.2155744
PPTPDF	**0.131626563**	0.1301556	0.1300224	0.1286948	0.1295876	0.1303589	0.1316981
XMLPDF	**0.216491576**	0.2172430	0.2177842	0.2178913	0.2169651	0.2171931	0.2165054
JPG	**0.799014321**	0.7997339	0.7968874	0.7966479	0.8034175	0.7949750	0.8002418
TIF	**0.025882716**	0.0259519	0.0256978	0.0257062	0.0258542	0.0259238	0.0259979
PNG	**0.044587319**	0.0448896	0.0446199	0.0446353	0.0447225	0.0446728	0.0446759
BMP	**0.020409852**	0.0201125	0.0200896	0.0202369	0.0200715	0.0200860	0.0203488
MP3	**0.013867125**	0.0136527	0.0139307	0.0137321	0.0138242	0.0138376	0.0137762
MP3-64Kb	**0.117246956**	0.1169477	0.1158613	0.1164297	0.1163459	0.1163869	0.1159267
MP3-128Kb	**0.061664256**	0.0569216	0.0572494	0.0572891	0.0564386	0.0575616	0.0564292
MP3-256Kb	**0.028269175**	0.0285086	0.0283171	0.0281878	0.0284937	0.0282826	0.0281313
WAV	**0.045198564**	0.0451589	0.0451134	0.0451638	0.0452104	0.0451912	0.0449093
WMV	**0.011697787**	0.0115693	0.0116329	0.0117380	0.0117136	0.0117131	0.0116312

C. Pruebas del NIST

Por último se muestra los resultados de aplicar el conjunto de pruebas estadísticas del NIST a un archivo, el cual se cifró con cada criptosistema antes mencionado. Para poder aplicar las pruebas, los archivos cifrados se binarizarón obteniendo así una cadena de 100 millones de bits. La tabla 3 muestra los P-valores obtenidos para la transformación tent así como un comparativo para los diferentes P-valores de otras transformaciones.

Tabla 3. P-valores de la transformación "tent"

PRUEBAS DE PSEUDOALEATORIEDAD DEL NIST						
No.	PRUEBA	P-VALOR				RESULTADO
		MAPLES TENT	MAPLES TENT MODIFIC	MAPLES LOGISTICO	AES	
1	Frecuency	**0.5955**	0.6371	0.2826	0.9403	PASA
2	Block Frecuency	**0.3669**	0.1620	0.4992	0.4559	PASA
3	Cumulative-sums Forward	**0.9357**	09420	0.8816	0.9340	PASA
	Cumulative-sums Reverse	**0.7197**	0.8103	0.2248	0.7791	PASA
4	Runs	**0.9357**	0.1472	0.4969	0.9558	PASA
5	Longest-Runs of Ones	**0.6579**	0.1597	0.9491	0.7597	PASA
6	Rank	**0.3503**	0.1372	0.4070	0.3304	PASA
7	Spectral ft	**0.0428**	0.0969	0.4599	0.0797	PASA
8	Overlapping-templates	**0.7792**	0.7197	0.5941	0.9858	PASA
9	Non-Overlapping-template	**0.4461**	0.6163	0.7981	0.5749	PASA
10	Universal	**0.3345**	0.1597	0.2133	0.6786	PASA
11	Aproximate-Entropy	**0.4012**	0.3545	0.7177	0.7597	PASA
12	Random-Excursions	**0.5047**	0.7197	0.9642	0.02818	PASA
13	Random-Excursions-Variant	**0.3789**	0.7597	0.9114	0.1005	PASA
14	Serial	**0.86669**	0.5463	0.6392	0.6371	PASA
15	Linear complexity	**0.5955**	0.8904	0.9642	0.9558	PASA

7. Conclusiones

En este trabajo se muestra la implementación general y el comportamiento de un cifrador caótico de bloques con la transformación caótica "tent". Podemos destacar que al utilizar esta transformación caótica es una buena alternativa para nuestro cifrador ya que es una transformación lineal a tramos y no posee islas de estabilidad problema que nos llevaría a señales periódicas. Esta última condición es indeseable dentro de un cifrador.

La aportación de este trabajo radica en estudiar el comportamiento de la transformación "tent" como función no lineal dentro de la estructura de Kocarev. Además se determinó que escalando y discretizando

dicha transformación y usando una precisión de 16 bits en los números que las alimentan es posible alcanzar el comportamiento caótico que produce señales de ruido con una buena distribución estadística.

Se determino un criterio de Autenticación que nos permite darle seguridad a nuestra cifrador caótico de bloques "tent", considerando un proceso de generación de claves hasta la comparación de la clave y el número de rondas que se utilizaron para cifrar y de esta forma comparar los valores para proceder al descifrado del mensaje.

Para evaluar el cifrador caótico "tent" se usaron conceptos de Teoría de la información, y de los resultados se puede concluir que la entropía de los mensajes cifrados está altamente relacionada con su distribución estadística. De manera que, si la entropía es cercana a la entropía máxima, la distribución estadística del mensaje cifrado es muy parecida a una distribución estadística uniforme.

Se valoró la información mutua del cifrador concluyendo que la información mutua es muy cercana a cero, buscando apegarse al criterio de criptosistema seguro de Shannon. Lo que significa que el mensaje cifrado no posee evidencia del mensaje original después del proceso de cifrado.

8. Bibliografía

[1] L. Kocarev, Goce Jakimoski "Chaos and Cryptography: "Block Encryption Ciphers Based on Chaotic Maps". IEEE Trans. On Circuits and Systems, 2001. Vol. 48(2)

[2] R.A. Rueppel: "Analysis and Design of Stream Ciphers", Springer Verlag, (1986).

[3] NIST. Federal Information Processing Standards Publication (FIPS 140-1). Security requirements for cryptographic modules; 1994.

[4] "Chaos and Nonlinear Dynamics An Introduction for Scientists and Engineers, Robert C. Hilborn, Department of Physics, Amherst College, Second Edition, Oxford University Press, Pág. 185.

Curriculum Vitae

Cesar E. Rojas López

Nació en Tehuantepec, Oaxaca. Recibió el título de Licenciado en Informática en el año 2000, por el Instituto Tecnológico del Istmo. Es profesor de tiempo completo del Instituto Politécnico Nacional, adscrito a la Escuela Superior de Ingeniería Mecánica y Eléctrica, ESIME, Unidad Culhuacan. Actualmente se encuentra realizando el tercer semestre de la Maestría en Ciencias en Ingeniería en Microelectrónica con especialidad en Seguridad Informática, en la Sección de Estudios de Posgrado e Investigación de la propia ESIME Culhuacan. Sus áreas de interés son la criptografía, la Seguridad Informática, Computación y las Comunicaciones. (crojas@ipn.mx).

Jorge Alberto Martínez Ñonthe

Recibió el título de Ingeniero en Comunicaciones y Electrónica por el Instituto Politécnico Nacional (2005). Obtuvo el Grado de Maestro en Ciencias de Ingeniería en Microelectrónica en la Sección de Estudios de Posgrado e Investigación en la ESIME Unidad Culhuacan del I.P.N (2008), actualmente cursa el primer semestre del Doctorado en Comunicaciones y Electrónica de la misma sección. Desde 2005 ha trabajado en infraestructura de comunicaciones e informática. Sus áreas de interés son la seguridad informática, Computación, comunicaciones y Auditoria informática. (jmartinezn9800@ipn.mx)

Rubén Vázquez Medina, ***Member, IEEE***

Nació en la Ciudad de México en 1966. Recibió el título de Ingeniero en Electrónica especialidad en Comunicaciones en 1988 en la Universidad Autónoma Metropolitana Unidad Iztapalapa y el grado de Maestro en Ciencias especialidad en Ingeniería Eléctrica opción en Telecomunicaciones en Septiembre de 1991 en el Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional. Obtuvo el grado de Doctor en Ciencias en la Universidad Autónoma Metropolitana Unidad Iztapalapa en Octubre de 2008. Fue jefe de la Sección de Estudios de Posgrado e Investigación de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhucan del 28 de Marzo del 2003 al 17 de agosto del 2006. Actualmente se desempeña como profesor en la Sección de Estudios de Posgrado de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacan del IPN, dentro de los programas de posgrado en Maestría en Ciencias de Ingeniería en Microelectrónica, Maestría en Ingeniería en Seguridad Informática y Tecnologías de la Información y en el Doctorado en Comunicaciones y Electrónica. Sus áreas de interés son la criptografía, la esteganografía, la informática forense y la forensia digital. (ruvazquez@ipn.mx).

CP-40
PON 89

Análisis comparativo de criptosistemas caóticos: analógicos y digitales

L. Palacios-Luengas, C. E. Rojas-López, J. A. Martínez-Ñonthe, A. Castañeda-Solís, R. Vázquez-Medina

Instituto Politécnico Nacional
SEPI ESIME Culhuacan
Av. Santa Ana No. 1000 Col. San Francisco Culhuacan Delegación Coyoacán,
04430 México D. F.
Email: lpluengas@msn.com, {crojas, jmartinezn9800, acastanedas, ruvazquez} @ipn.mx

1. Resumen

En este artículo se examinan algunos sistemas criptográficos en relación con el uso y aplicación de transformaciones caóticas para procesos de protección de información. Se definen aquellos sistemas caóticos que pueden aplicarse en criptografía y se establecen las ventajas y desventajas de sus versiones analógicas y digitales. Este artículo tiene como finalidad preparar el estado del arte y el punto de partida de una investigación más ampliada relacionada al diseño, implementación y finalmente el desarrollo de algoritmos criptográficos, los cuales incluyan transformaciones caóticas de diversos tipos.

2. Introducción

El caos puede definirse como la tendencia de los sistemas iterativos, simples, determinísticos a ser sensibles a las condiciones iniciales y a ser altamente no predecibles. El elemento central de los sistemas caóticos aquí considerados es el concepto de iteración. El estado actual del sistema es una función determinística del estado o valor anterior. Formalmente una correspondencia caótica se especifica por medio de la expresión: $x_{n+1}=f(x_n)$ donde $f(x_n)$ es una función no lineal.

ROC&C'2010 – CP-40 PONENCIA RECOMENDADA POR EL **COMITÉ DE COMPUTACION** DEL **IEEE SECCIÓN MÉXICO** Y PRESENTADA EN LA **REUNIÓN DE OTOÑO, ROC&C'2010**, ACAPULCO, GRO., DEL 28 DE NOVIEMBRE AL 4 DE DICIEMBRE DEL 2010.

En general, la teoría del caos surge de las matemáticas de modelización de mecanismos físicos tales como la predicción del tiempo atmosférico, la evolución de la población, la dinámica de fluidos, la teoría de gases, etc. Todos los ejemplos anteriores son sistemas iterativos por naturaleza. Por ejemplo, la población de los próximos años es una función de la de este año. Se ha observado que los modelos más simples producen un comportamiento altamente no lineal; de modo, que variando las condiciones iniciales, una cantidad pequeña, puede tener unos efectos completamente impredecibles después de varias iteraciones. Todo sistema caótico es muy sensible a las condiciones iniciales y genera un comportamiento aparentemente aleatorio pero a la vez completamente determinístico.

Estas propiedades del caos proporcionan un potencial para aplicaciones en criptografía, ya que las predicciones a largo plazo de los sistemas caóticos son muy difíciles. El hecho de que sea determinístico significa que se puede obtener el mismo conjunto de valores siempre que se disponga de la misma función de correspondencia caótica y de su valor inicial.

Puesto que las funciones caóticas son muy sensibles a las condiciones iniciales, cualquier pequeña diferencia en el valor inicial empleado, significará que la salida producida será muy diferente. Si estos sistemas se usan en criptografía, esta condición significa que el sistema será robusto contra ataques por fuerza bruta. Esto se debe a que el número de posibles claves puede llegar a ser muy grande, y dependerá de la precisión de los valores iniciales, la cual a su vez estará en función del hardware utilizado.

3. Clasificación de criptosistemas caóticos

Desde el punto de vista del tipo de señales que se procesan, se puede considerar que hay dos clases de cifradores caóticos: los cifradores analógicos, basados en técnicas de sincronización e implementados con circuitos analógicos, y los cifradores digitales, basados en caos e implementados con circuitos digitales.

Los cifradores analógicos o continuos utilizan transformaciones caóticas contínuas como el de Lorenz [1], circuito de Chua [2] y sistema Rössler [3]. Para estos, el proceso de cifrado se realiza sumando la señal del mensaje analógica a la salida de la transformación caótica. El proceso de descifrado se realiza restando la señal cifrada de la salida de la transformación caótica sincronizada. La señal cifrada toma el aspecto de ruido, ya que se ha mezclado con una señal caótica.

En los cifradores digitales o discretos se utilizan transformaciones caóticas discretos como el de la transformación Logística [4], Bernoulli [5] y Senoidal [6], entre otras, las cuales se utilizan como la función generadora de ruido. Cuando se usan las versiones discretizadas y escaladas de estas funciones, para el procesamiento de archivos, la información manejada puede considerarse una cadena de números enteros en el intervalo cerrado [0, 255]. El primer valor puede utilizarse como la condición inicial en la transformación caótica. El proceso de cifrado iterará la condición inicial un número predeterminado de veces especificado por la clave, la salida obtenida será el texto cifrado. El proceso de descifrado consiste en realizar sobre el texto cifrado el mismo número de iteraciones inversas utilizando la misma transformación caótica y la salida resultante es el texto en claro.

Otra forma de clasificar a los criptosistemas caóticos es atendiendo a como se utiliza la transformación caótica. Un caso es cuando se utiliza como ruido aleatorio que se envía a la salida. Por ejemplo, en sistemas simétricos OTP ("*One Time Pad*") [7]. Otro tipo es cuando la transformación caótica se puede utilizar como función de ruido sobre el texto en claro, por ejemplo en sistemas asimétricos.

4. Sistemas criptográficos caóticos analógicos

Los sistemas criptográficos analógicos, destinados principalmente a enmascarar la señal de voz habían caído en desuso debido a la facilidad de realizar la recuperación total o parcial de la información transmitida. Así, las técnicas analógicas están siendo sustituidas por técnicas digitales. Sin embargo, a partir de los planteamientos que hizo Lewis y Carroll [8] se ha despertado un nuevo interés por los criptosistemas analógicos. Estos planteamientos mostraron algunos ejemplos sobre el comportamiento caótico imprevisibles que inicialmente evolucionan sobre trayectorias diferentes y pueden unirse en una sola trayectoria en común siempre y cuando se acoplen adecuadamente. A partir de los trabajos de Lewis y Carroll, los sistemas criptográficos se basan en el concepto de sincronización caótica, lo cual significa que la información puede transmitirse por la señal caótica en varias formas. A continuación se describe cada una de ellas.

4.1. Enmascaramiento caótico

El enmascaramiento caótico "*Chaos Masking*" (CM) [9-13] se basa en utilizar una señal caótica como enmascarante de una señal de información, luego por el canal transmitir la suma de estas dos señales para ocultar la señal transmitida. El diagrama a bloques del enmascaramiento caótico se muestra en la figura 4.1. Básicamente una señal portadora de información *S (t)* se agrega a la salida *X (t)* del sistema caótico del transmisor en el lado del receptor un sistema caótico idéntico trata de sincronizarse con *X (t)*. Desde el punto de vista de la señal de información *X (t)* es una perturbación, y la sincronización se da solo aproximadamente. Aun así, si el error de sincronización es pequeño con respecto a *S'(t)*, esta última puede ser recuperada por sustracción.

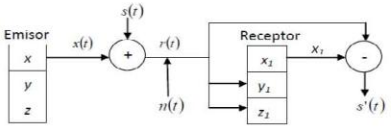


Figura4.1. Diagrama a bloques del enmascaramiento caótico.

4.2. Conmutación caótica

La idea principal de la conmutación caótica "*chaotic switching or chaos shift keying*" (CSK) [14-15] es modular uno de los coeficientes del sistema transmisor con la información de una determinada forma de onda y transmitir la señal caótica. En otras palabras, se modula algún parámetro de la señal caótica para la transmisión de información. El esquema del enmascaramiento mediante CSK

(figura4.2) consiste en una señal de información *S (t)* binaria, la cual controla un conmutador que cambia los valores de los parámetros del sistema caótico. De esta forma, el valor de *S (t)* en un instante *t*, el sistema caótico tiene el vector de parámetros *p* o en su caso *'p*. La salida *Y (t)* se transmite en dos copias del sistema caótico, vector de parámetros *p* y *'p*. Si la posición del interruptor en el transmisor se encuentra en la posición *p*, entonces el sistema con el vector *p* en el receptor se sincronizará; por otro lado, si la posición se encuentra en *'p* se des-sincronizará. De esta forma, la señal de error *e (t)* convergerá a cero, mientras que *e' (t)* tendrá una forma irregular con amplitud diferente de cero. Por lo tanto, la señal *S (t)* puede recuperarse a partir de las señales de error *e (t)* y *e' (t)*.

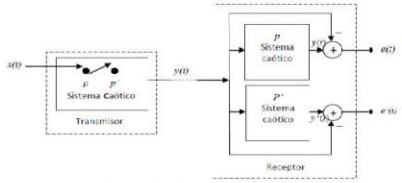


Figura 4.2. Transmisión CSK.

4.3. Modulación caótica

La modulación caótica "*chaotic modulation*" [16-19] tiene dos formas para modular la señal de mensaje dentro de portadoras caóticas. El primer método es llamado modulación de parámetros caóticos mostrado en la figura 4.3a usa una señal de mensaje para cambiar parámetros del transmisor caótico. El segundo método llamado modulación caótica no autónoma mostrada en la figura 4.3b usando la señal de mensaje para cambiar el espacio de fase del transmisor caótico.

4.3.1. Modulación de parámetros caóticos

En la figura 4.3a la señal de mensaje *m (t)* se utiliza para modular algunos parámetros del sistema caótico en el transmisor, de manera que cambian sus trayectorias distintas a atractores caóticos [20]. Dado que el espacio de bifurcación es muy complejo, es muy difícil averiguar la forma en que cambian los parámetros, incluso si algún intruso tiene el conocimiento parcial del sistema caótico en el transmisor. En el extremo el receptor un control adaptativo se utiliza para ajustar los parámetros del sistema caótico, de tal forma que el error de sincronización se aproxima a cero. De esta manera, la salida del controlador adaptable puede recobrar la señal de mensaje.

4.3.2. Modulación caótica no autónoma

En lugar de cambiar los parámetros del transmisor caótico, la modulación caótica no autónoma de la figura 4.3b utiliza la señal de mensaje para perturbar el atractor caótico directamente en el espacio de fase. A diferencia de la modulación de parámetros caóticos, donde el transmisor cambia entre las diferentes trayectorias en diferentes atractores caóticos, el transmisor en la modulación caótica no autónoma cambia entre las diferentes trayectorias en el mismo atractor caótico. Teóricamente, modulación caótica no autónoma es un error de esquema libre ("*error free scheme*"). La segunda generación mejoró el grado de seguridad hasta cierto punto, pero se encuentra todavía insuficiente.

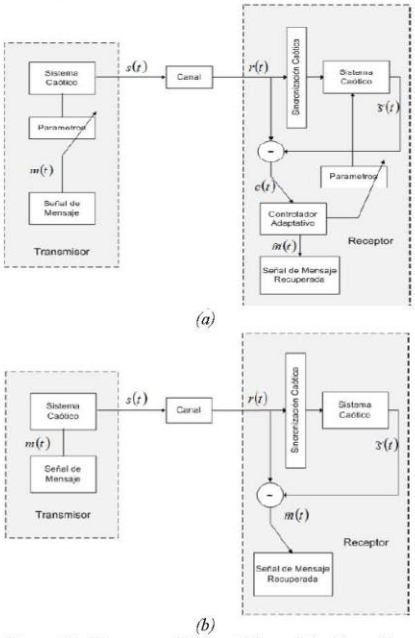


Figura 4.3. Diagrama a bloques de la modulación caótica a) Modulación de parámetros caóticos, b) Modulación caótica no autónoma.

4.4. Criptosistema caótico

En el criptosistema caótico "chaos control" [21-23] la señal de texto plano *p (t)* es cifrada por una regla

con una señal clave $k(t)$, la cual es generada por el sistema caótico en el transmisor. La señal mezclada se utiliza para manejar el sistema caótico, de tal manera que la dinámica caótica cambia al sistema caótico en el transmisor, la cual se transmite a través del canal público el cual puede acceder el intruso. Dado que el intruso no puede acceder a la clave de hardware caótica, es muy difícil encontrar $p(t)$ a partir de $S(t)$. En el receptor la señal recibida $r(t) = S(t) + n(t)$, donde $n(t)$ es el ruido del canal, es utilizada para sincronizar los dos sistemas caóticos en el transmisor y el receptor. Después de que la sincronización caótica se ha logrado, la señal $k(t)$ y $y(t)$ se pueden recuperar en el receptor con algunos ruidos denotados por $\hat{k}(t)$ y $\hat{y}(t)$. Para la alimentación $\hat{k}(t)$ y $\hat{y}(t)$ en la regla de descifrado en el receptor, la señal de texto plano se puede recuperar con algunos ruidos como $\hat{p}(t)$.

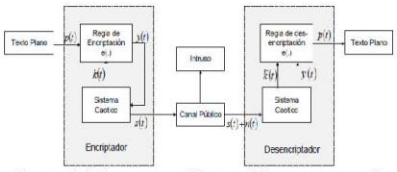


Figura 4.4. Diagrama a bloques del criptosistema de comunicaciones caóticas seguras.

Para finalizar el análisis de los sistemas analógicos podemos decir que existe la preocupación de que los esquemas de comunicación no puedan ser lo suficientemente seguros. Para superar este inconveniente, una manera de abordar el problema es estudiar él hipercaos [24] basado en sistemas de comunicaciones seguras, pero estos sistemas pueden presentar más dificultades para la sincronización. Por otro lado, podemos mejorar la seguridad del caos a baja dimensionalidad ("*low-dimensional*") basado en esquemas de comunicaciones seguras mediante la combinación de sistemas criptográficos convencionales con un sistema caótico.

5. Sistemas criptográficos caóticos digitales

El segundo acercamiento al diseño de sistemas criptográficos basados en caos, consiste en utilizar sistemas digitales (tales como computadoras, microcontroladores, DSPs, FPGAs, etc.) para recorrer una transformación caótica y la máscara del mensaje binario en un número de formas [25-30]. Estos cifradores no dependen de la sincronización. En su lugar, por lo general utilizan uno o más transformaciones caóticas, donde la condición inicial x_0 y el valor del parámetro desempeñan el papel de la clave. Los sistemas criptográficos pertenecientes a estos sistemas se dividen en dos categorías:

a) Aquellos que utilizan alguna función caótica como PRNG ("*Pseudo Random Number Generador*") [31], para la generación de la clave aleatoria, o para usarla como fuente de números aleatorios para un criptosistema denominado simétrico de flujo OTP ("*One Time Pad*"). En estas aplicaciones, la clave es el estado inicial del sistema.
b) Aquellos que hacen corresponder el texto en claro al estado inicial del sistema caótico, y a continuación hacen pasar al sistema por un ciclo de iteraciones, siendo que el estado resultante es el texto cifrado. En estas aplicaciones, se usa una clave y debe de ser:
    i. El algoritmo de correspondencia.
    ii. Los detalles de la función que representan el sistema.
    iii. El número de iteraciones.
    iv. Cualquier combinación de las tres anteriores.

Un ejemplo de este tipo de aplicación de correspondencia es un sistema que segmenta números reales en un número de partes igual al tamaño del alfabeto, e itera estos valores, utilizando la función caótica durante muchos ciclos, para obtener el texto cifrado. Los siguientes criterios corresponden a los de las funciones caóticas ideales, que debe cumplir cualquier sistema caótico que pretenda utilizarse en criptografía:

a) Semillas o condiciones iniciales muy similares deben producir secuencias muy diferentes de valores. Para las aplicaciones de correspondencia, claves similares deberían cifrar el texto en claro dando lugar a texto cifrado muy diferente.
b) Cada secuencia debería ser aleatoria, aperiódica, para cualquier longitud de mensaje concebible. Las aplicaciones PRNG deberían ser no periódicas para prevenir coincidencia y ataques de inferencia. Las aplicaciones de correspondencia deberían no tener patrones para ocultar cualquier similitud entre texto en claro y cifrado, y se no periódicas, para asegurar que cada texto cifrado se descifra a un único texto en claro. La ausencia de periodicidad es crítica, si un texto en claro se itera a cualquier valor, que es un elemento de un periodo, entonces puede ser indistinguible de otros elementos.

c) Para aplicaciones PRNG el conocimiento de una sucesión de elementos de la secuencia no debería permitir predecir los elementos anteriores o posteriores. Para aplicaciones de correspondencias, la función no debería ser fácilmente reversible sin la clave.
d) Debería existir un número de claves viable mayor que el número más grande concebible de sesiones de comunicaciones que tengan lugar durante el ciclo de vida de la función caótica.
e) La progresión del sistema, de un estado al siguiente, debería ser determinístico y reproducible. El grado de cumplimiento de cada criterio anterior varía. Por ello, debe realizarse un análisis de cada implementación. Se deben tener en cuenta aspectos como:
   i. Las funciones caóticas son muy sensibles a las condiciones iniciales, con un pequeño cambio en las condiciones iniciales se produce cambios importantes en la secuencia de valores generada.
   ii. Una función caótica, por definición, presenta un comportamiento no lineal que puede mejorarse. Es posible optimizar la no linealidad de una función dada y controlar la aleatoriedad.
   iii. Si se revela toda la información sobre el estado del sistema todos los estados siguientes se podrán calcular. Es importante construir PRNG que sólo utilicen información de estado parcial como salida.
   iv. Debe examinarse el sistema respecto al número de claves posibles y si existen claves débiles.
   v. Cuando se utilice matemática punto flotante es crítico que todas las partes tengan igual precisión, ya que cualquier redondeo inconsistente puede dar lugar a texto cifrado no reconocible.

5.1 Cifrador de flujo basado en PRNGs caóticos

Este tipo de criptosistemas utilizan transformaciones caóticas para generar un tipo de secuencia pseudoaleatoria con la que se mezcle el mensaje en claro. La mezcla de la secuencia cifrante con el mensaje se realiza bit a bit, haciendo uso de una función XOR. En la figura 5.1 se muestra la arquitectura general de un cifrador de flujo, el cual es un dispositivo con memoria interna, que transforma el dígito de la cadena del texto claro en el dígito *c* de la cadena del texto cifrado, por medio de una función que depende de una llave secreta *K* y del estado interno del cifrador de flujo en el tiempo *j*.

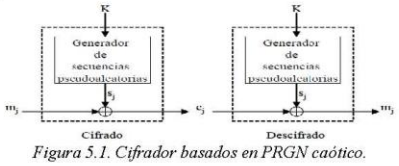


Figura 5.1. Cifrador basados en PRGN caótico.

5.2 Cifrador de Bloque

El cifrado de bloque es un cifrado de sustitución simple y debe hacerse a grandes porciones del texto claro para prevenir ataques por fuerza bruta. El nombre de bloque es usado para limitar el tamaño de la entidad de texto claro a cifrar.

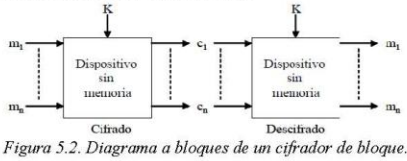


Figura 5.2. Diagrama a bloques de un cifrador de bloque.

En la figura 5.2 se observa que un criptosistema de bloques es un dispositivo sin memoria que, bajo el control de una llave *K* en un bloque de texto cifrado, transforma cada bloque del texto claro *m*. El alfabeto del texto claro y el alfabeto del texto cifrado usualmente es el mismo. Generalmente, este tipo de criptosistema se basa en las redes de Feistel [32].

En las redes de Feistel balanceadas, un bloque de tamaño N bits comúnmente N=64 ó 128 bits se divide en dos semibloques de tamaño N/2, denominados por A y B. A partir de aquí, comienza el proceso de cifrado y consiste en aplicar una función unidireccional (muy difícil de invertir) a un semibloque B y a una sub-llave *k1* generada a partir de la llave secreta. Posteriormente, se mezcla el semibloque A con el resultado de la función mediante un XOR. Luego, se permutan los semibloques y se repite el proceso n veces. Finalmente, se unen los dos semibloques en un bloque único. Como se ilustra en la figura 5.3.

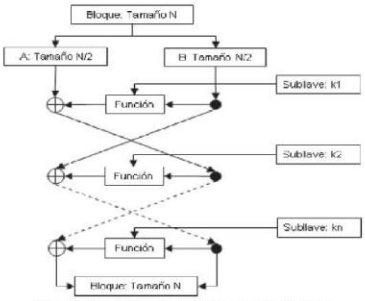


Figura 5.3. Cifrado por bloques de Feistel.

6. Análisis de comparación de criptosistemas: analógicos y digital

Los sistemas de mayor difusión y amplio uso para cifrar información se basan en criptografía digital [33-35]. Dicha masificación se justifica por la facilidad de implementación en arquitecturas digitales y que ofrecen una recuperación casi perfecta de la información. Sin embargo, hay debilidades que se han hecho presentes en algunos sistemas que, por su naturaleza, han permitido que los sistemas analógicos sigan usándose.

Uno de los problemas que deben considerarse, tanto en la implementación de sistemas criptográficos analógicos como digitales es el ruido. Cabe mencionar que los sistemas analógicos son más susceptibles que los digitales, y en ambos casos deberá ser ampliamente estudiado su efecto en la implementación.

En los sistemas criptográficos caóticos digitales el mensaje sólo se puede codificar en un campo, Z_n, de números finitos. Esto significa que en vez de tener un conjunto de números continuo (como el conjunto de los números reales R), para operar, se limita a un subconjunto del infinito. Ahora bien, si no se tiene el debido cuidado, se provocaría que la función caótica genere órbitas periódicas, lo cual no ocurre con las órbitas caóticas del dominio analógico. Por ello, cuando se diseñan circuitos electrónicos que habrán de operar como criptosistemas digitales, se debe considerar el uso de técnicas que permitan generar secuencias tan largas como la tecnología lo permita.

Otro aspecto que se debe destacar es que actualmente existe la posibilidad de cifrar en hardware en tiempo real a elevada velocidad, haciendo uso de sistemas y herramientas actuales, las cuales permiten implementaciones muy eficientes haciendo uso de circuitos analógicos.

Por último, en la tabla 6.1 se muestras comparaciones más relevantes de ambas tecnologías.

Tabla 6.1 Análisis de comparación cifradores: analógicos y digitales.

Sistemas dinamicos continous	Sistemas dinamicos discretos
Supcetibles al ruido	Se puede recuperar la señal en un 100%
Presentan algunos poblemas de implementación en Hardware	Su implementación es realtivamente sencillo
Difícil de implementar para sistemas que usan transmisión digital	Relativamente Facil en implementación
Se requiere una señal de sincronización que se envía por el canal	No requiere una señal de sincronización

7. Conclusiones

Se presentó un análisis de los sistemas basados en transformaciones caóticas para implantaciones meramente criptográficas. Considerando métodos analíticos de tecnología continua y discreta es posible realizar implementaciones que cumplen con las propiedades de difusión y confusión de Shannon.

Los sistemas criptográficos analógicas y digitales brindan soluciones de protección de datos, pero es necesario realizar el análisis para conocer el área en donde serán enfocados y, finalmente, como serán implementados. Las dos tecnologías planteadas en este artículo ofrecen soluciones muy interesantes dependiendo de las necesidades a cubrir. Por una parte, se tiene la tecnología analógica que no tiene muchas ventajas y probablemente el campo de aplicación es limitado por las cuestiones comentadas. Sin embargo, la utilización sigue siendo un factor importante en ciertas áreas un ejemplo práctico es en la necesidad de enviar información cifrada en los radios inalámbricos de corto alcance.

Los criptosistemas digitales permiten soluciones mucho muy interesantes debido a la facilidad de realizarlas y es posible la implementación de algoritmos mucho muy complejos para obtener sistemas robustos. Es importante considerar el rápido desarrollo en cuanto a la tecnología de hardware programable y configurable que permite realizar estas implementaciones.

Agradecimientos

Este artículo se enmarca dentro de los apoyos otorgados por CONACyT y por el apoyo en el proyecto SIP-20101510 del Instituto Politécnico Nacional.

8. Referencias

[1]. R. Nuñez P., "Implementación y Prueba de un Comunicador Caótico Bidireccional de Información Oculta, Basado en Dos Circuitos Sincronizados de Lorenz". Informe Técnico: CTETT20005, DET-CICESE (2000).

[2]. H. Dedieu; M. P. Kennedy y M. Hasler, "Chaos Shift Keying: Modulation and Demodulation of a Chaotic Carrier Using Self-Synchronizing Chua's Circuits". IEEE Transactions on Circuits and Systems II, Vol. 40, No. 10, pp. 634-642, (1993).

[3]. C. Aguilar Ibáñez, J. Sánchez H., M.S. Suárez C. y F. Flores A. "Identificación del sistema de Rössler: enfoque algebraico y algoritmos genéticos". CIC-IPN (2005).

[4]. R. Bedient and M. Frame, "Carrying surfaces for return maps of averaged logistic maps". Computers & Graphics, Volume 31, Issue 6, December 2007, Pages 887-895.

[5]. William C. Saphir and Hiroshi H. Hasegawa, "Spectral representations of the Bernoulli map". Physics Letters A Volume 171, Issues (5-6), 14 December 1992, Pages 317-322.

[6]. L. M. Benítez, "Mapeo Caótico Senoidal aplicado al cifrador de bloques". SEPI-Culhucan, Instituto Politécnico Nacional, (2010).

[7]. S. Patil, M. Devare & Ajay Kumar," Modified One Time Pad Data Security Scheme: Random Key Generation Approach". International Journal of Computer Science and Security (IJCSS), Volume (3): Issue (2), Pages 138 – 141.

[8]. L.M. Pecora and T.L. Carroll, "Synchronization in chaotic systems".

[9]. L. Kocarev, K. S. Halle, K. Eckert, L. O. Chua, and U. Parlitz, "Experimental demonstration of secure communications via chaotic synchronization". Int. J. Bifurc. Chaos, 2:709–713, (1992).

[10]. C. W. Wu and L. O. Chua, "A simple way to synchronize chaotic systems with applications to secure communications systems". Int. J. Bifurc. Chaos, 3:1619–1627, (1993).

[11]. K. M. Cuomo, A. V. Openheim, and S. H. Strogatz. "Synchronization of lorenz-based chaotic circuits with applications to communications". IEEE Trans. Circuits Syst – II, 40:626–633, (1993).

[12]. O. Morgul and M. Feki. "A chaotic masking scheme by using synchronized chaotic systems". Phys. Lett. A, 251:169–176, (1999).

[13]. S. M. Shahruz, A. K. Pradeep, and R. Gurumoorthy. "Design of a novel cryptosystem based on chaotic oscillators and feedback inversion". J. Sound and Vibration, 250:762–771, (2002).

[14]. H. Dedieu, M. P. Kennedy, and M. Hasler. "Chaos shift keying: Modulation and demodulation of a chaotic carrier using self-synchronizing". IEEE Trans. Circuits and Systems – II, 40:634–641, 1993.

[15]. U. Parlitz, L. O. Chua, L. Kocarev, K. S. Halle, and A. Shang. "Transmission of digital signals by chaotic synchronization". Int. J. Bifurcation. Chaos, 2:973–977, 1992.

[16]. K. S. Halle, C. W. Wu, M. Itoh, and L. O. Chua. "Spread spectrum communication through modulation of chaos". Int. J. Bifurcation. Chaos, 3:469–477, 1993.

[17]. K.M. Cuomo and A. V. Openheim. "Circuit implementation of synchronized chaos with applications to communications". Phys. Rev. Lett., 71:65–68, 1993.

[18]. M. Feki. "An adaptive chaos synchronization scheme applied to secure communication". Chaos, Solitons and Fractals, 18:141–148, 2003.

[19]. J. Y. Chen, K. W. Wong, L. M. Cheng, and J. W. Shuai. "A secure communication scheme based on the phase synchronization of chaotic systems". Chaos, 13:508–514, 2003.

[20]. Lorenz, Edward, "The Essence of Chaos". University Of Washington Press, Seattle, (1995).

[21]. S. Hayes, C. Grebogi, and E. Ott. "Communicating with chaos". Phys. Rev.Lett., 70:3031–3034, 1993.

[22]. S. Hayes, C. Grebogi, E. Ott, and A. Mark. "Experimental control of chaos for communication". Phys. Rev. Lett., 73:1781–1784, 1994.

[23]. C. Grebogi, Y. Lai, and E. Bolt. "Communicating with chaos using two-dimensional symbolic dynamics". Phys. Lett. A, 255:75–81, 1999.

[24]. T. Kapitaniak, "Chaos synchronizations and hiperchaos", Conference Series 23 (2005) 317–324.

[25]. M. S. Baptista, "Cryptography with chaos". Phys. Lett. A, 240:50–54, 1998.

[26]. J. Fridrich. "Symmetric ciphers based on two-dimensional chaotic maps". Int. J. Bifurc. Chaos, 8:1259–1284, 1998.

[27]. N. K. Pareek, V. Patidar, and K. K. Sud. "Discrete chaotic cryptography using external key". Phys. Lett. A, 309:75–82, 2003.

[28]. W. Wong, L. Lee, and K. Wong. "A modified chaotic cryptographic method". Computer Physics Communications, 138:234–236, 2001.

[29]. E. Álvarez, A. Fernández, P. García, J. Jiménez, and A. Marcano. "New approach to chaotic encryption". Phys. Lett. A, 263:373–375, 1999.

[30]. P. García and J. Jiménez. "Communication through chaotic map systems". Phys. Lett. A, 298:35–40, 2002.

[31]. Ali Kanso and Nejib Smaoui, "Logistic chaotic maps for binary numbers generations", Chaos, Solitons & Fractals, Volume 40, Issue 5, 15 June 2009, Pages 2557-2568.

[32]. Knudsen Lars R., "The security of Feistel ciphers with six rounds or less". Journal of cryptology, 2002, vol. 15, no3, pp. 207-222 (31 ref).

[33]. E. Lopez," Implementaci´on Eficiente en FPGA del Modo CCM usando AES", sección de computación CINVESTAV-IPN, (2005).

[34]. M. A. Ajo, G. Fericean, M. Borda, and V. Rodellar, "An IP design of the IDEA Cryptographic Algorithm", Facultad de Informática, Universidad Politécnica de Madrid.

[35]. F.I. Peralta C, "Diseño de Arquitecturas Digitales para Criptografía", SEPI-Culhucan, Instituto Politécnico Nacional, (2005).

9. Biografías

Leonardo Palacios Luengas. Recibió el título de Ingeniero en Comunicaciones y Electrónica, con especialidad en Comunicaciones en el año 2003, por el Instituto Politécnico Nacional. Trabajo como Diseñador Electrónico en el Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional en el proyecto MARINA-2002-CO1-3199 del fondo sectorial CONACyT-Secretaria de Marina. Actualmente se encuentra realizando el primer semestre de la maestría en Ciencias de la Ingeniera en Microelectrónica en el área de seguridad informática en la Sección de Estudios de posgrado e Investigación de la propia ESIME Culhuacan. Su experiencia se basa en las áreas de: Diseño Electrónico Digital y Programación de Sistemas Embebidos. Sus áreas de interés son, la Criptografía Electrónica y Comunicaciones. (lpluengas@msn.com).

César Enrique Rojas López. Nació en Tehuantepec, Oaxaca. Recibió el título de Licenciado en Informática en el año 2000, por el Instituto Tecnológico del Istmo. Es profesor de tiempo completo del Instituto Politécnico Nacional, adscrito a la Escuela Superior de Ingeniería Mecánica y Eléctrica, ESIME, Unidad Culhuacan. Actualmente se encuentra realizando el tercer semestre de la Maestría en Ciencias en Ingeniería en Microelectrónica con especialidad en Seguridad Informática, en la Sección de Estudios de Posgrado e Investigación de la propia ESIME Culhuacan. Sus áreas de interés son la criptografía, la Seguridad Informática, Computación y las Comunicaciones. (crojas@ipn.mx).

Jorge Alberto Martínez Ñonthe. Recibió el título de Ingeniero en Comunicaciones y Electrónica por el Instituto Politécnico Nacional (2005). Obtuvo el Grado de Maestro en Ciencias de Ingeniería en Microelectrónica en la Sección de Estudios de Posgrado e Investigación en la ESIME Unidad Culhuacan del I.P.N (2008), actualmente cursa el tercer semestre del Doctorado en Comunicaciones y Electrónica de la misma sección. Desde 2005 ha trabajado en infraestructura de comunicaciones e informática. Sus áreas de interés son la Seguridad Informática, Computación, Comunicaciones y Auditoria informática. (jmartinezn9800@ipn.mx).

Antonio Castañeda Solis

Nació en México D.F. Obtuvo el grado de Ingeniero de comunicaciones y Electrónica del IPN, (titulación por créditos de posgrado).Obtuvo el grado de Maestría en Ciencias en Ingeniería de Telecomunicaciones, por parte del IPN con el tema de tesis: Soluciones fundamentales cuaterniónicas para las ecuaciones de Maxwell y sus aplicaciones en la solución numérica de problemas con valores de frontera. Obtuvo el Doctorado en Comunicaciones y Electrónica, IPN. Tema de tesis: Métodos pseudoanalíticos aplicados a modelos de la teoría de campos. Es Profesor en el área de Seguridad en redes de cómputo como parte de la Maestría en Seguridad de la Información impartida en la SEPI-Culhuacan. Es profesor en el Centro de estudios Superiores Navales de la materia de Seguridad en redes de cómputo como parte de la Maestría en Seguridad de la Información. Fue Director del área de TI Planeación y coordinación de proyectos de TI relacionados con "modelos de gobierno de TI. Es Miembro del Sistema Nacional de Investigadores del CONACyT (SNI).Sus áreas de interés son la seguridad en redes y la la criptografía. (acastanedas@ipn.mx).

Rubén Vázquez Medina, Member, IEEE . Nació en la Ciudad de México en 1966. Recibió el título de Ingeniero en Electrónica especialidad en Comunicaciones en 1988 en la Universidad Autónoma Metropolitana Unidad Iztapalapa y el grado de Maestro en Ciencias especialidad en Ingeniería Eléctrica opción en Telecomunicaciones en Septiembre de 1991 en el Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional. Obtuvo el grado de Doctor en Ciencias en la Universidad Autónoma Metropolitana Unidad Iztapalapa en Octubre de 2008. Fue jefe de la Sección de Estudios de Posgrado e Investigación de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhucan del 28 de Marzo del 2003 al 17 de agosto del 2006. Actualmente se desempeña como profesor en la Sección de Estudios de Posgrado de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacan del IPN, dentro de los programas de posgrado en Maestría en Ciencias de Ingeniería en Microelectrónica, Maestría en Ingeniería en Seguridad Informática y Tecnologías de la Información y en el Doctorado en Comunicaciones y Electrónica. Sus áreas de interés son la criptografía, la esteganografía, la informática forense y la forensia digital. (ruvazquez@ipn.mx).

8

LIII Congreso Nacional de Física
Boca del Río Veracruz, del 25 al 29 de octubre de 2010

Criptosistema caótico de bloques usando la transformación tent

J. A. Martínez-Ñonthe*, C. E. Rojas-López, J.L. del Río Correa, J. A. Díaz-Méndez, R. Vázquez-Medina
*e-mail:jmartinezn9800@ipn.mx

RESUMEN; Se presenta la construcción y evaluación de un criptosistema de bloques basado en la transformación caótica unidimensional "tent". El algoritmo propuesto opera como cifrador de 64 bits y se ha evaluado empleando conceptos de la teoría de la información como son la entropía y la información mutua. Los resultados muestran un adecuado comportamiento del criptosistema para otorgar confidencialidad a los archivos de cualquier formato. La aleatoriedad de la salida del criptosistema fue evaluada con la batería de pruebas del NIST. Para precisar y analizar el comportamiento de la función de combinación, como generadora de ruido en este criptosistema, se han usado las herramientas de la mecánica estadística como son el diagrama de bifurcación, el exponente de Lyapunov, la distribución invariante y el Teorema Ergódico.

TRANSFORMACIÓN TENT

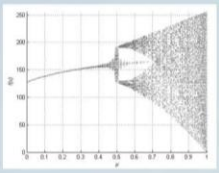
Función Original

$$f(x)=\begin{cases}2\mu x_n+\dfrac{(1-\mu)}{2};\ldots\ldots\ldots\ldots 0\le x_n\le\dfrac{1}{2}\\ 2\mu(1-x_n)+\dfrac{(1-\mu)}{2};\ldots\ldots\dfrac{1}{2}<x_n\le 1\end{cases}$$

Función escalada y discretizada

$$f(x)=\begin{cases}floor\left[2\mu x_n+\left(255*\left(\dfrac{1-\mu}{2}\right)\right)\right]\\ floor\left[-\mu\left(2\left(x_n-255\right)\right)+\left(255*\left(\dfrac{1-\mu}{2}\right)\right)\right]\end{cases}$$

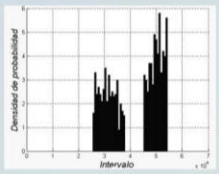
DIAGRAMA DE BIFURCACIÓN



DISTRIBUCIÓN ESTADÍSTICA

Para obtener el histograma asociado a la órbita se hace uso del **Teorema Ergódico de Birkhoff**:

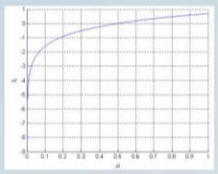
$$\overline{b(x_0)}=\langle b(x)\rangle=\int b(x)\rho_{est}(x)dx$$



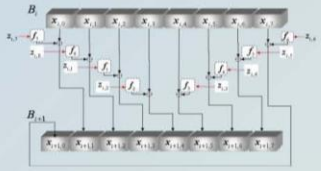
EXPONENTE DE LYAPUNOV

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

Agradezco por el apoyo otorgado al IPN como alumno PIFI, Proyecto 20101510 y al CONACYT para la realización y presentación de este trabajo



ALGORITMO DESARROLLADO



Esquema de Cifrado

Esquema de Descifrado

RESULTADOS DE ENTROPÍA

$$H=-\sum_{i=1}^{n}P(x_i)\log_2\left[\frac{1}{p(x_i)}\right]$$

INFORMACIÓN MUTUA

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

Se busca que:

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

RESULTADOS DEL NIST

CONCLUSIONES

Se implementó un algoritmo de cifrado por bloques caóticos, usando el mapeo caótico tent. El algoritmo fue evaluado usando conceptos fundamentales de la Mecánica Estadística. Así como la Teoría de la información (Entropía y Información Mutua). La entropía de la señal de salida está altamente relacionada con su distribución estadística, de manera que si la entropía es cercana a la entropía máxima, la distribución estadística de la salida es muy parecida a una distribución estadística uniforme. Se ha trabajado con transformaciones generalizados y discretizados que permitan tener como dominio de definición el universo de los símbolos ASCII usados en una computadora.

Bibliografía

1. Nieto de Alba, Ubaldo., [2008] "Predicción y Caos en Economía" Universidad Complutense.
2. A.L. Goldberg. [2008] "Caos y fractales en la fisiología Humana," Investigación y Ciencia, Vol. 163.
3. 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. Facultad de ciencias, Departamento de Físico Matemáticas Universidad Autónoma de San Luis Potosí.
4. Devaney, R. L. [1989] An introduction to Chaotic Dynamical Systems (Addison-Wesley, Redwood City, California, USA).
5. L. Kocarev, G. J. [2001] logistic map as a block Encryption Algorith, Vol. 289.

CP-07
P20

ESTRATEGIAS DE CONSTRUCCIÓN DE CLAVES PARA CIFRADORES DE BLOQUES

L. M. Benítez-Barrón*, J. A. Martínez-Ñonthe, C. E. Rojas-López, R. Vázquez-Medina

INSTITUTO POLITÉCNICO NACIONAL
SEPI, ESIME Culhuacan,
Av. Santa Ana 1000, Col. San Francisco Culhuacan,
Del. Coyoacán, 04430 México, D.F., MÉXICO.
e-mail: lbenitezb0401@ipn.mx*, jmartinezn9800@ipn.mx, crojas@ipn.mx, ruvazquez@ipn.mx

1. Resumen

En este artículo se hace una revisión de las estrategias usadas por los algoritmos DES (Data Encryption Standard) y AES (Advanced Encryption Standard) para generar las claves de cifrado en cada ronda del proceso. Además, se propone una implementación en Matlab® para generar claves de cifrado basada en las estrategias revisadas y se muestra su efecto en un cifrador caótico de bloques.

Palabras clave: Algoritmo, Cifrador caótico de bloques, Mapeo Senoidal, AES, DES, Clave.

2.- Introducción

Uno de los principios de la criptografía establece que la seguridad de un proceso criptográfico solo debe depender de la secrecía de la clave utilizada. Este principio fue establecido por Auguste Kerchoff en el siglo 19, y dice que "Un criptosistema debe ser seguro aun si todo sobre el sistema, excepto la clave, es de conocimiento público"[1]. En teoría, cualquier proceso criptográfico puede ser roto probando todas las posibles claves. A este proceder se le conoce como ataque por fuerza bruta. Si la única opción es probar todas las claves, el tiempo de computación requerido aumenta exponencialmente con la longitud de la clave. Así, un sistema con las claves de 56 bits (como DES) toma un esfuerzo sustancial, pero es fácilmente rompible con hardware especial, como el FEP DES cracker (apodado "Deep crack"), el cual es una máquina construida por la Electronic Frontier Foundation (EFF) en 1998 para realizar una búsqueda de fuerza bruta de la clave del cifrador DES [2]. Sin embargo, existen estrategias que permiten romper un proceso de cifrado, sin necesariamente probar todas las posibilidades de clave.

ROC&C'2009 – CP-07 PONENCIA RECOMENDADA POR EL **COMITÉ DE COMPUTACIÓN** DEL **IEEE SECCIÓN MÉXICO** Y PRESENTADA EN LA **REUNIÓN DE OTOÑO, ROC&C'2009**, ACAPULCO, GRO., DEL 29 DE NOVIEMBRE AL 5 DE DICIEMBRE DEL 2009.

Estas estrategias son: (a) Ataque del cumpleaños, el cual se basa en la matemática detrás de la paradoja del cumpleaños, haciendo uso de una situación de compromiso espacio-tiempo informática [3]; (b) Ataque de hombre en el medio (man in the middle attack) el cual es un ataque en el que el enemigo adquiere la capacidad de leer, insertar y modificar a voluntad mensajes entre dos partes sin que ninguna de ellas conozca que el enlace entre ellos ha sido violado [4]. (c) Ataque de encuentro en el medio (meet in the middle attack) el cual es un ataque similar al ataque de cumpleaños, que utiliza un compromiso entre tiempo y espacio [5]. Por ello, no es conveniente basar la seguridad de un proceso solo en la longitud de la clave. En general, es muy difícil diseñar cifradores que no puedan romperse usando otros métodos más eficaces como los sistemas dinámicos [6].

3. Clave en el algoritmo DES

El DES es un cifrador de bloques de 64 bits que emplea cajas de permutación (p-boxes) y sustitución (s-boxes), de expansión y reducción. Usa una clave de 64 bits, de los que 8 son paridad.

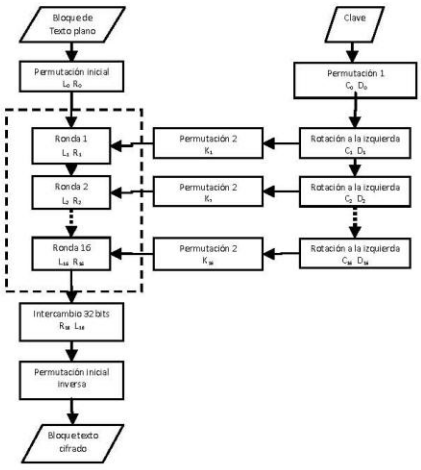


Fig. 1 Esquema general del algoritmo DES

DES tiene 19 etapas diferentes, la primera es una permutación inicial del texto plano. La última etapa es la inversa de la primera. La penúltima etapa intercambia los 32 bits de la izquierda y los 32 bits de la derecha. Las 16 etapas restantes son una cada una, una Red balanceada de Feistel. En cada una de las 16 iteraciones se emplea un valor K_i, obtenida a partir de la clave de 56 bits y distinto en cada iteración. A continuación se describe como se genera y usa la clave en DES. Se realiza una permutación inicial sobre la clave, y luego la clave obtenida se divide en dos mitades de 28 bits, cada una de las cuales se rota a la izquierda un número de bits determinado (depende de C(i-1) y D(i-1) para obtener C(i) y D(i), respectivamente.). K_i se deriva de la elección permutada de 48 de los 56 bits de estas dos mitades rotadas, en la figura 2 se puede observar claramente su estructura.

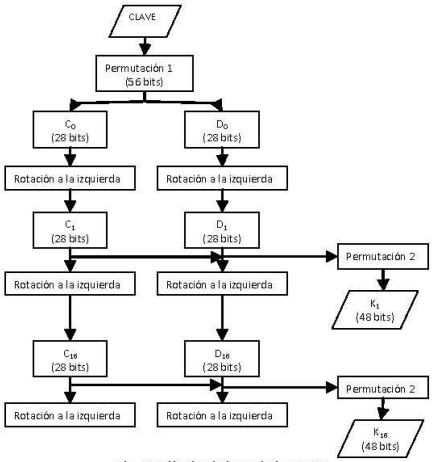


Fig. 2 Cálculo de las subclaves, K_i

Su proceso de descifrado es el mismo algoritmo empleando las K_i en orden inverso. Todo esto está descrito oficialmente en el documento FUP 46-3 [7].

4.- Clave en el algoritmo AES

El AES es un cifrador de bloques de 128 bits y longitudes de claves de 128, 192 y 256 bits, con 10, 12 y 14 vueltas respectivamente (N_r), cada vuelta consiste en la aplicación de una ronda estándar, que consiste de 4 funciones matemáticas diferentes e invertibles. La ronda inicial y final consiste de 1 y 3 funciones matemáticas respectivamente [8]. La información generada por cada función es un resultado intermedio, que se conoce como *Estado*.

El algoritmo representa el *Estado* como una matriz rectangular de bytes, que posee 4 filas y N_b columnas. Siendo el número de columnas N_b en función del tamaño del bloque.

$$N_b = \frac{Tamaño\ del\ bloque\ en\ bits}{32} \quad \ldots(1)$$

La clave principal se representa mediante una matriz de bytes de 4 filas y N_k columnas. Siendo el número de columnas en función del tamaño de la clave.

$$N_k = \frac{Tamaño\ de\ la\ clave\ en\ bits}{32} \quad \ldots(2)$$

Como se observa en la figura 3 el algoritmo AES tiene 4 funciones básicas:

SubBytes.- En este paso se realiza una sustitución no lineal donde cada byte es reemplazado con otro de acuerdo a la S-BOX de Rijndael.

ShiftRows.- En este paso se realiza una transposición donde cada fila del *Estado* es rotada de manera cíclica un número determinado de veces.

MixColumns.- Operación de mezclado que opera en las columnas del *Estado*, combinando los cuatro bytes en cada columna usando una transformación lineal.

AddRoundKey.- Cada byte del *Estado* es combinado con la subclave; cada subclave se deriva de la clave de cifrado usando una iteración de la clave.

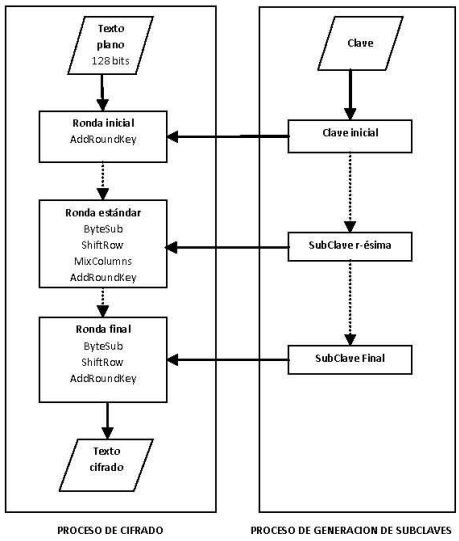


Fig.3 Esquema general del algoritmo AES

Lo principal de esta función son las subclaves, que se generan mediante bytes que se derivan de la clave principal *K*. Para ello, el proceso de generación de subclaves utiliza 2 funciones auxiliares:

Selección de claves.- Toma consecutivamente de la secuencia obtenida por la función de expansión de clave bytes que va asignado a cada subclave K_i, para formar

bloques del mismo tamaño que la matriz *Estado*; es decir, toma N_b*4 bytes para cada vuelta.

Expansión de clave.- Permite generar bytes útiles como subclaves a partir de la clave principal *K*. Se describe como un arreglo lineal, denominado *W*, de palabras de 4 bytes y con una longitud de:

$$[W_i], 0 \le i \le N_b(N_r + 1) \quad \ldots(3)$$

Las primeras N_k palabras de este arreglo contienen la clave principal, ya que se mapea tal cual al arreglo *W*, mientras que el resto de las palabras se van generando a partir de estas primeras N_k palabras.
Para la generación de las demás *W* tenemos tres funciones que realizan este trabajo:
RotWord() toma una palabra de 4 bytes $[a_0,a_1,a_2,a_3]$ y realiza una permutación cíclica como $[a_1,a_2,a_3,a_4]$.
SubWord() toma una palabra de 4 bytes y aplica la S-BOX de Rinjndael a cada una de los 4 bytes.
Rcon[i] representa una constante que contiene los valores dados por

$$[R(j),\{00\},\{00\},\{00\}] \quad \ldots(4)$$

donde *R(j)* es el elemento de los campos finitos GF(2^8) correspondientes al valor x^{i-1}.

Desde un punto más esquemático la función de expansión se puede ver como:

Para $N_k \le 6$
Para todo valor de *i* que no sea múltiplo de N_k , las palabras subclaves se calculan como

$$W(i) = W(i - N_k) \oplus W(i-1) \quad \ldots(5)$$

Para todo valor de *i* que sea múltiplo de N_k

$$W(i) = W(i - N_k) \oplus \left[SubWord(RotWord[w(i-1)]) \oplus Rcon\left(\frac{i}{N_k}\right)\right] \quad \ldots(6)$$

Para $N_k \ge 6$
El funcionamiento es igual que para $N_k \le 6$ salvo cuando el valor de la variable *i* satisface que $i \bmod N_k = 4$, en este caso las palabras de subclaves se calculan como

$$W(i) = W(i - N_k) \oplus SubWord(W[i-1]) \quad \ldots(7)$$

5.- Implementación propuesta para el cifrador caótico de bloques

En el algoritmo de cifrado de bloques propuesto en [9], se trabaja dos funciones caóticas, el mapeo logístico y el modificado de Bernoulli. En esta ocasión se trabajará con la misma estructura del cifrador (una red de Feistel desbalanceada), pero con una función diferente la cual es la transformación caótica senoidal.

La transformación senoidal se define como:

$$x_{n+1} = \mu sen(\pi x_n) \quad \ldots(8)$$
$$\mu \in (0,1) \quad x_0 \in (0,1)$$

Cuando μ=0.87 La función senoidal exhibe un comportamiento caótico; y por lo tanto, la propiedad de sensibilidad a las condiciones iníciales. Este comportamiento se puede observar en la figura 4.

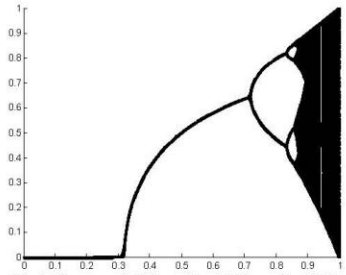


Fig 4. Diagrama de Bifurcación del Mapeo Senoidal.

El proceso de cifrado trabaja con el universo del los caracteres ASCII, por la cual, el mapeo se discretizo y normalizo en un intervalo de (0,255), quedando definido de la siguiente manera.

$$f(x) = \left\{floor\left[(\mu * 255)\sin\left(\left(\frac{x}{255}\right)\pi\right)\right]\right\} \quad \ldots(9)$$

5.1 Descripción del cifrador caótico

El cifrador caótico propuesto en [9], toma bloques de texto plano de 64 bits con 8 rondas, donde el bloque de salida de una ronda es la entrada en la siguiente ronda, excepto en la última, la cual es el bloque de texto cifrado. Es importante mencionar que el tamaño del bloque cifrado es de 64 bits (8 bytes) y es del mismo tamaño que el bloque de texto claro (Figura 5). Cada ronda se compone de una red de Feistel desbalanceada que utiliza una subclave de 64 bits.

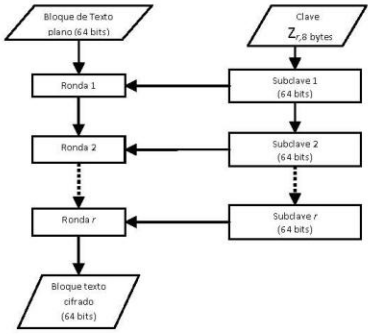


Fig. 5 Esquema general del cifrador caótico

Cada ronda se compone de una red de Feistel desbalanceada (tomando como referencia el esquema en [10]), que utiliza una subclave de 64 bits y lo podemos observar en la figura 6.

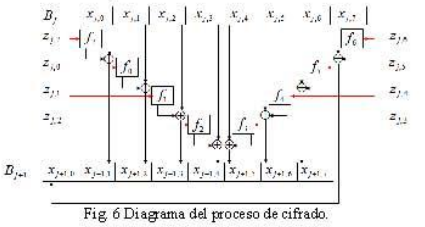


Fig. 6 Diagrama del proceso de cifrado.

5.2 Estrategia de construcción de la clave

Como lo mencionamos anteriormente, lo principal del los algoritmos de cifrado es la creación de las subclaves. Una estrategia para probar la eficiencia del algoritmo propuesto es la creación de las subclaves en forma aleatoria. En este caso, el cifrador está compuesto de 8 rondas lo cual determina se necesita una clave para cada una de las de 8 ronda y con un tamaño de 8 bytes cada una, lo que da 512 bits por cada bloque cifrado.

Un método rápido y exacto para generar una clave es usando funciones HASH, que por sus propiedades, con las cuales se obtiene la misma clave cuando se usa la misma condición inicial. Ya que se necesita 512 bits para las subclaves la más idónea de las funciones HASH es la SHA-256 [11].

Bajo este esquema de uso de claves surge un conflicto, ya que es posible la repetición de algunos de los bloques de texto plano, y si se usa la misma clave, daría como resultado el mismo bloque cifrado. Esto provoca redundancia en el texto cifrado y le quita pseudoaleatoriedad. Para enfrentarlo y resolverlo se usa la siguiente estrategia. Se toma el resultado de aplicar la SHA-256 a la clave principal y se vuelve a pasar por la SHA-256, como se muestra en la figura 7.

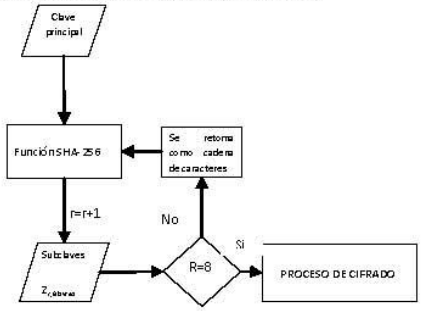


Fig. 7 Estructura de la clave

La limitante de este proceso es que la Función SHA-256 maneja hexadecimales, por lo cual la clave propuesta solo maneja un universo de 16 caracteres.

5.3 Herramientas de evaluación

Para evaluar el cifrador caótico se toma como base los conceptos referentes al criptosistema seguro de Shannon [12]. Se emplea la entropía y la información mutua figuras de mérito de los procesos criptográficos.

La entropía es la cantidad de información promedio del mensaje y se utiliza para medir que tan lejos está el criptosistema evaluado de un criptosistema seguro (sistema equiproblema). La entropía máxima es aquella cuya huella estadística es uniforme. Lo que se busca al tratar de tener un sistema equiprobable es llevar al mensaje cifrado al estado de mayor incertidumbre. La entropía se define como:

$$H(C) = -\sum_{i=1}^{n} p_i \log p_i \quad \ldots(10)$$

La información mutua mide la cantidad de información que aporta sobre una variable el conocimiento de otra. Se dice que un criptosistema es seguro si la cantidad de información que aporta el hecho de conocer el mensaje cifrado C sobre la entropía del texto plano M vale cero. Es decir:

$$I(C,M) = 0 \quad \ldots(12)$$

Por último, el cifrador caótico es evaluado con la suite de pruebas estadísticas de pseudoaleatoriedad del NIST (National Institute of Standards and Technology) [13], la cual consiste de 15 pruebas que son consideradas internacionalmente para evaluar secuencias criptográficas. Cada prueba de la suite del NIST arroja un P-valor que se encuentra en un intervalo de [0,1], donde el valor 1 representaría una secuencia totalmente aleatoria (solo teóricamente) y el valor cero representa que la secuencia no es nada aleatoria. El mínimo P-valor para considerar que el archivo es pseudoaleatorio es P-valor≥0.001.

Con esto si un criptosistema tiene una información mutua muy cercana a cero y la entropía del texto es cercana a 8, entonces se espera que su huella estadística sea muy parecida a la uniforme y por lo tanto el mensaje debe parecer ruido. Por lo tanto, se espera que al evaluar el texto cifrado con las pruebas del NIST dicho texto cifrado debe parecer una secuencia de ruido.

6 Resultados

Las siguientes tablas muestran los resultados obtenidos en las tres pruebas mencionadas, las cuales son comparadas con dos de los criptosistemas mas conocidos, el AES y el DES.

Las pruebas se realizaron a 16 archivos de diferentes tipos. Primero se muestra los resultados de la entropía (Tabla 1), la primera columna muestra el tipo de archivo, la segunda es la entropía del archivo original, la tercera es la entropía del cifrador con la misma clave para todos los bloques, la cuarta es el cifrador con la estructura

propuesta de la clave, la quinta y la sexta son las entropías de los cifradores AES y DES.

Tabla 1 Entropía

ARCHIVO	ORIGINAL	CIFRADO			
		CLAVE ESTATICA	CLAVE ALEATORIA	AES	DES
TXT	5.0775	7.9970	7.9992	7.9991	7.9992
DOC	5.7806	7.8859	7.9995	7.9995	7.9995
RTF	5.4569	7.4751	7.9999	7.9999	7.9999
PDF	7.5589	7.9917	7.9992	7.9993	7.9993
XLS	5.8964	7.2171	7.9927	7.9934	7.9923
BMP	7.7959	7.9999	7.9999	8.0000	7.9999
TIF	7.5952	7.9999	7.9999	7.9999	8.0000
JPG	7.9176	7.9938	7.9989	7.9990	7.9990
GIF	7.9806	7.9995	7.9996	7.9995	7.9996
ZIP	7.9918	7.9979	7.9983	7.9982	7.9987
PDFZIP	7.9987	7.9991	7.9991	7.9990	7.9990
WAV	7.4144	7.9998	7.9999	7.9999	7.9998
MP3-64K	7.6117	7.9986	7.9992	7.9993	7.9992
MP3-128K	7.8720	7.9902	7.9996	7.9996	7.9996
MP3-256K	7.9393	7.9994	7.9998	7.9998	7.9998
MP3-320K	7.8913	7.9976	7.9998	7.9998	7.9998

La tabla 2 muestra la información mutua.

Tabla 2 Información Mutua

Archivo	CLAVE ESTATICA	CLAVE ALEATORIA	AES	DES
TXT	0.1802	0.0697	0.0693	0.0700
DOC	0.7343	0.1424	0.1428	0.1436
RTF	0.8787	0.0074	0.0076	0.0076
PDF	0.6711	0.2038	0.2051	0.2067
XLS	1.3746	0.8051	0.7931	0.7966
BMP	0.0221	0.0210	0.0129	0.0128
TIF	0.0135	0.0124	0.0123	0.0124
JPG	0.3788	0.3052	0.3054	0.3030
GIF	0.1183	0.1179	0.1173	0.1180
ZIP	0.4960	0.4946	0.4962	0.4913
PDFZIP	0.2687	0.2715	0.2672	0.2080
WAV	0.0373	0.0370	0.0372	0.0369
MP3-64K	0.2573	0.2412	0.2411	0.2408
MP3-128K	0.2006	0.1136	0.1132	0.1127
MP3-256K	0.0686	0.0548	0.0550	0.0550
MP3-320K	0.0817	0.0443	0.0443	0.0443

Por último, se muestra los resultados de aplicar la suite de pruebas estadísticas del NITS a un archivo (Tabla 3), el cual se cifró con cada cifrador antes mencionado. Para poder aplicar las pruebas, los archivos cifrados se binarizaron obteniendo así una cadena 100 millones de bits.

Tabla 3 Pruebas de pseudoaleatoriedad del NIST

No.	Prueba	P-valor			
		CLAVE ESTATICA	CLAVE ALEATORIA	AES	DES
1	Frequency	0.739918	0.637119	0.946308	0.015598
2	Block Frequency	0.851383	0.162606	0.455937	0.739918
3	Cumulative-sum Forward	0.262249	0.924076	0.924076	0.350485
	Cumulative-sum Reverse	0.153763	0.816537	0.779188	0.075719
4	Runs	0.017912	0.137282	0.955835	0.236810
5	Long Runs of Ones	0.213309	0.153763	0.759756	0.637119
6	Rank	0.000170	0.137282	0.350485	0.319084
7	Spectral DFT	0.334538	0.096578	0.075719	0.289667
8	Non-Overlapping Templates	0.181557	0.616305	0.574903	0.383827
9	Overlapping Templates	0.554420	0.719747	0.383827	0.419021
10	Universal	0.455937	0.153763	0.678686	0.924076
11	Approximate Entropy	0.014550	0.334538	0.759756	0.350485
12	Random Excursions	0.723129	0.719747	0.028181	0.249284
13	Random Excursions Variant	0.723129	0.759756	0.100508	0.145326
14	Linear Complexity	0.249284	0.350485	0.554420	0.213309
15	Serial	0.739918	0.946308	0.637119	0.015598

7 Conclusiones

Se observa que con la nueva estructura de la clave el cifrador propuesto mejora considerablemente la entropía y la información mutua. Se observa además que los resultados arrojados con los cifradores AES y DES son parecidos a los del cifrador propuesto.
Como trabajo a futuro se tratará de expandir el universo de los caracteres de la clave y se piensa que con esto los resultados serán aun mejores.

8 Bibliografía

[1] A. Kerckhoffs, "La cryptographie militaire", *Journal des sciences militaires*, vol. IX, pp. 5–83, 1883 Enero.

[2] Electronic Frontier Foundation, "Cracking DES - Secrets of Encryption Research", Wiretap Politics & Chip Design. 1998, Oreilly & Associates Inc..

[3] M. Bellare, T. Kohno: "Hash Function Balance and Its Impact on Birthday Attacks." EUROCRYPT 2004: pp401-418

[4] G. Coretez. "Bypassing Secure Web Transaction via DNS Corruption. A Man-in-The-Middle-Attack." 1999 April, Endeavor Information Systems, Inc.

[5] W. Diffie and M. E. Hellman. "Exhaustive Cryptanalysis of the NBS Data Encryption Standard." Junio 1977, Computer 10 (6): 74-84.

[6] http://cnx.org/content/m20332/latest/

[7] National Institute of Standards and Technology, "Federal information Processing standards publication 46-3." 1999 Octubre 25.

[8] NIST, "Federal Information Processing Standards Publication 197." 2001 Noviembre 26.

[9] Gutierrez Flores S., et al, "Algoritmo cifrador de bloques usando mapeos caóticos" ROC&C'2008, Acapulco, Gro., del 30 de noviembre al 6 de diciembre del 2008.

[10] L. Kocarev and Goce Jakimoski. "Logistic Map as a Block Encryption Algorithm"

[11]NIST, "Federal Information Processing Standards Publication 180-2." 2002 Agosto 1.

[12] C. E. Shannon: "A mathematical theory of communication". Bell System Technical Journal, vol. 27, pp. 379–423 and 623–656, 1948 Julio y Octubre.

[13] NIST, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications" Special Publication 800-22 Revision 1, 2008 Agosto.

9 Curriculum Vitae

Luz María Benítez Barrón, Nació en México D.F. en 1986 y egreso de la carrera de Ingeniería en Computación por el Instituto Politécnico Nacional unidad ESIME Culhuacan (2008), estudia el posgrado de la Maestría en Ciencias de la Ingeniería en Microelectrónica con especialidad en Seguridad Informática de la Sección de Estudios de Posgrado e Investigación ESIME Culhuacan

Jorge Alberto Martínez Ñonthe, Nació en la Ciudad de México en 1982 y es Ingeniero en Comunicaciones y Electrónica por el Instituto Politécnico Nacional (2005). Obtuvo el Grado de Maestro en Ciencias de Ingeniería en Microelectrónica en la Sección de Estudios de Posgrado e Investigación en la ESIME Unidad Culhuacan del I.P.N (2008), actualmente cursa el primer semestre del Doctorado en Comunicaciones y Electrónica de la misma sección. Desde 2005 ha trabajado en infraestructura de comunicaciones e informática. Sus áreas de interés son la seguridad informática, Computación, comunicaciones y Auditoria informática.

Cesar Enrique Rojas Lopez, Profesor de tiempo completo del Instituto Politécnico Nacional, adscrito a la Escuela Superior de Ingeniería Mecánica y Eléctrica, ESIME, Unidad Culhuacan. Realizando el primer semestre de la Maestría en Ciencias en Microelectrónica con especialidad en Seguridad Informática, en la Sección de Estudios de Posgrado e Investigación de la propia ESIME Culhuacan.

Rubén Vázquez Medina, ***Member, IEEE*** Recibió el título de Ingeniero en Electrónica especialidad en Comunicaciones de Septiembre de 1984 a Octubre de 1988 en la Universidad Autónoma Metropolitana y el grado de Maestro en Ciencias especialidad en Ingeniería Eléctrica opción en Telecomunicaciones de Septiembre de 1989 a Septiembre de 1991 en el CINVESTA- IPN, Obtuvo el grado de Doctor en Ciencias en la Universidad Autónoma Metropolitana (2008), fue jefe de la Sección de Estudios de Posgrado e Investigación del 28 de Marzo del 2003 al 17 de agosto del 2006.. Puesto actual profesor de la Sección de Estudios de Posgrado de ESIME Culhuacan.

CM-06
P22

Cifrador caótico de bloques logístico

J. A. Martínez-Ñonthe, L. M. Benítez-Barrón, C. E. Rojas-López, M. Cruz-Irisson, R Vázquez-Medina

Instituto Politécnico Nacional
SEPI ESIME Culhuacan
Av. Santa Ana No. 1000 Col. San Francisco Culhuacan Delegación Coyoacan,
04430 México D. F.

Email: nonthealbert1@hotmail.com,{ lbenitezb0401, crojas, ruvazquez} @ipn.mx

1. Resumen.

En este artículo 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 el diseño y evaluación de un algoritmo de cifrado de bloques basado en el modelo propuesto por Ljupco Kocarev. Este algoritmo se construye a partir del Escalamiento y Discretización de la trasformación logística [1], y la red desbalanceada de Feistel [2]. El algoritmo opera como cifrador de bloques, con tamaño de bloque y longitud de llave de 64 bits. Para medir la difusión en el proceso de cifrado se calcula la entropía de archivos de diferentes formatos a la entrada y se compara con la entropía de los archivos respectivos a la salida. Finalmente, usando el criterio de seguridad de Shannon, se compara la fortaleza del algoritmo propuesto con la de otros algoritmos de cifrado de bloques como el DES, TRIPLE DES, AES y BLOWFISH.

2. Introducción.

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, ha sido ampliamente estudiado y ha encontrado un sin número de aplicaciones en diferentes áreas de la ciencia. Por ejemplo, en la economía, al estudiar el comportamiento de los mercados financieros [3], la medicina, en el estudio del sistema inmunitario humano [4], la biología, en el estudio de encimas y hormonas sujetas a la dinámica caótica [5], etc.

ROC&C'2009 – CM-06 PONENCIA RECOMENDADA POR EL **COMITÉ DE COMUNICACIONES** DEL **IEEE SECCIÓN MÉXICO** Y PRESENTADA EN LA **REUNIÓN DE OTOÑO, ROC&C'2009**, ACAPULCO, GRO., DEL 29 DE NOVIEMBRE AL 5 DE DICIEMBRE DEL 2009.

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 [6]. El comportamiento caótico de un sistema no lineal 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. La definición dada por Devaney [7] para un sistema caótico es la siguiente: Sea X un espacio métrico. Un mapeo continuo $f: X \to X$ se dice ser caótico en X sí:

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

3. Transformación Logística

La trasformación logística se define como un sistema dinámico discreto determinista y unidimensional que puede iterarse, de manera que se elige un número cualquiera como dato de entrada de la función y el resultado obtenido se utiliza como dato de entrada en la misma función en la siguiente aplicación (iteración). La transformación Logística puede obtenerse de ecuaciones no lineales simples [1] [8]. Fue popularizada en 1976 [9]. La función logística está dada por la ecuación (1):

$$x_{n+1} = \mu x_n (1 - x_n) \qquad (1)$$

Donde μ es el parámetro de esta función, que en general toma un valor dentro del intervalo $[0,4]$. En términos técnicos, representa una razón de crecimiento. x_n representa el número de individuos de la población en la n-enésima generación. 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.

El comportamiento de la trasformación logística se define como la familia de curvas parabólicas este comportamiento se muestra en la figura 1, como una función de x_n [0,1] y para diferentes valores del parámetro μ [0,4]. Esta familia de curvas se pueden obtener usando la ecuación (2) con $x \in (0,1)$ y $\mu \in (0,1)$.

$$\tau_\mu(x) = \mu x(1-x) \qquad (2)$$

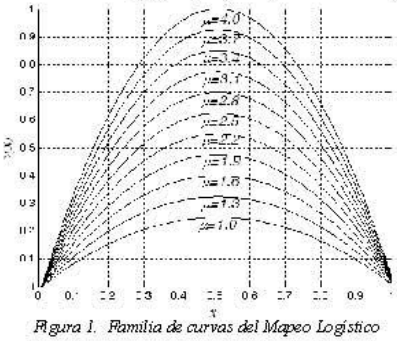


Figura 1. Familia de curvas del Mapeo Logístico

3.1 Diagrama de Trayectorias

Una manera más de ver el proceso de dispersión generado por estas trasformaciones unidimensionales, es considerar el diagrama de trayectorias, que muestra como gradualmente el mapeo cubre el área del cuadrante. Para valores de $0 \le \mu \le 4$ en la trasformación logística la altura de la parábola estará en el intervalo [0, 1]. Si se itera esta función se observa la dinámica discreta de la población que modela la función.

Para iterar una función se necesita un valor inicial y el resultado será la siguiente entrada de la función. Por lo tanto ver ecuación (3):

$$\begin{aligned} x_1 &= f(x_0) \\ x_2 &= f(x_1) = f^2(x_0) \\ &\cdots \\ x_n &= f(x_{n-1}) = f^n(x_0) \end{aligned} \qquad (3)$$

Donde x_n es la n-enésima 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 $f(x)$. Ver figura 2.

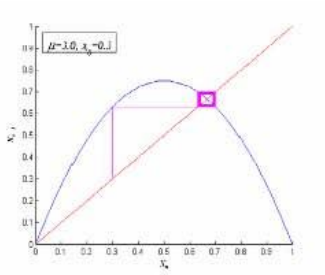


Figura 2. Diagrama de Trayectorias para el punto atractor (1-1/μ)

3.2 Diagrama de Bifurcación

Los diagramas de bifurcación son herramientas muy útiles para observar el comportamiento de la transformación. Es posible determinar las regiones periódicas y aperiódicas de la transformación. Básicamente, se grafica el valor de μ contra los puntos donde la dinámica se ha concentrado después de algunas iteraciones iniciales, esto es, se grafican los puntos fijos atractores. En la figura 3 se puede observar que para $0 < \mu < 1$, 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,....

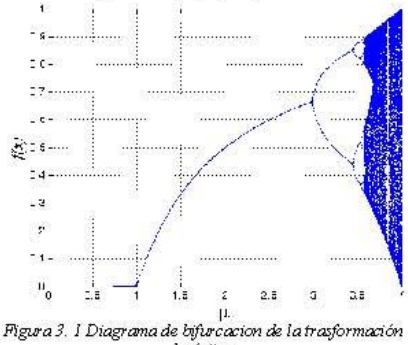


Figura 3. 1 Diagrama de bifurcacion de la trasformación logística

Nótese que en el diagrama de bifurcación se observa algunas regiones donde hay otros periodos llamados islas de estabilidad en las cuales la transformación tiene un comportamiento periódico, a pesar de que ya se encuentra en la región caótica también llamada inestable. En la figura 4 nótese una gran isla de

estabilidad de periodo 3 cerca de $\mu = 3.83$. Estos periodos también se duplican a 6, 12, 24,..., periodos y nuevamente alcanzan una región caótica. Las islas de estabilidad son regiones donde el sistema retorna a un comportamiento periódico

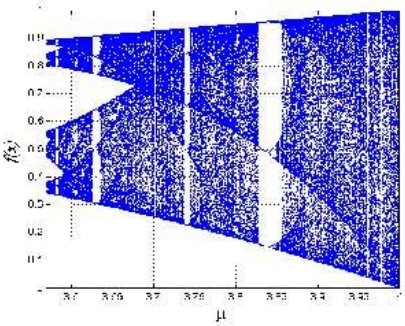


Figura 4. Isla de periodo 3.

A partir del diagrama de bifurcación es posible notar que esta transformación caótica tiene un comportamiento fractal. Esto es, si nosotros realizamos un acercamiento en regiones especificas y haciendo rotaciones y cambios de escala se reproduce nuevamente el diagrama de bifurcación original. Ver figura 5 donde se realiza un acercamiento para un valor $\mu = 3.56$.

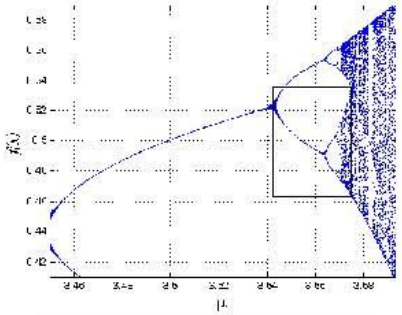


Figura 5. Acercamientos en el diagrama de bifurcación para apreciar el comportamiento auto similar

Nótese que se ha realizado una ampliación de la sección que se indica con rectángulo en negro de la figura 5 y en la figura 6 se ha graficado en orden inverso al eje vertical.

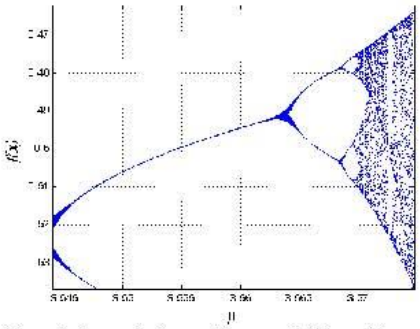


Figura 6. Acercamientos en el diagrama de bifurcación para apreciar el comportamiento auto similar en orden inverso al eje vertical

Se dice que el diagrama de bifurcación es un fractal también llamado autosimilar debido a que si se hace un acercamiento (zoom) a alguna de las regiones del diagrama original, se encontrará un diagrama con un comportamiento semejante al que tiene el diagrama original. El diagrama de bifurcación se conoce también como diagrama de Feigenbaum debido a que está muy relacionado con el trabajo de Mitchell Feigenbaum [10].

3.3 Distribución Estadística

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. En la figura 6. Se puede apreciar que la densidad de probabilidad es muy parecida a la uniforme, exceptuando en los extremos límite del intervalo.

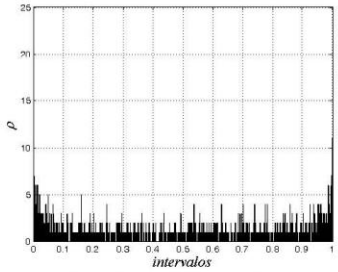


Figura 6. Mapeo Logistico para μ=4.0, Densidad de probabilidad

Los histogramas que describen la función de densidad de probabilidad para los mapeos se calculan aprovechando la propiedad de que la Transformación logística es Ergódica, lo que significa que el comportamiento a largo plazo de una sola órbita, que se obtiene al dar una condición inicial e iterar la trasformación, un número grande de veces es igual al comportamiento estadístico de un ensamble de condiciones iniciales. Si f es la trasformación entonces,

$$\overline{f} = \langle f \rangle \tag{4}$$

3.4 Escalamiento y Discretización

El proceso de escalamiento y discretización obedece a la definición del intervalo o universo de interés. Para este caso particular, ese universo corresponde al código ASCII extendido, definido en el intervalo [0,255]. Este intervalo se conoce como intervalo de definición de la transformación. Ahora bien partiendo de la ecuación (1).

$$x_{n+1} = \mu x_n (1 - x_n) \tag{1}$$

Donde: $x_n = 1/2$, Validado x_{n+1} tenemos:

$x_{n+1} = \mu \frac{1}{2}(1 - 1/2) = \mu . \frac{1}{2}(1/2) = \frac{\mu}{4}$ Obtenemos un valor de: $\frac{\mu}{4}$

La figura 7 muestra el valor de la parábola cuando tiene un valor de $\frac{\mu}{4}$ esto cuando nuestro intervalo esta definido en $x_n \in (0,1); \mu \in (0,4)$

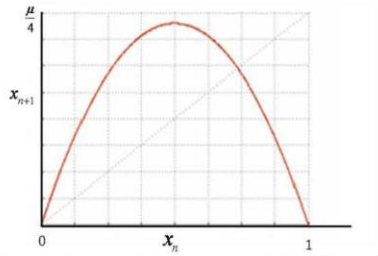


Figura 7 Curva escalada en un intervalo de (0,1) cuando $x_{n+1} \in (0,1)$

La función de transformación se obtiene a partir de la ecuación de transformación logística en dos pasos:

- La trasformación logística es escalado de tal manera que los valores de entrada y salida de la trasformación estén en el intervalo [0,255] de los reales y no en un intervalo (0,1) ver ecuación (5).

$$x_n \in (0,1) \rightarrow x_n \in [0,255] \rightarrow \mathbf{R} \tag{5}$$

- La trasformación logística escalada es discretizada de tal manera que los valores de la trasformación estén en el intervalo [0,255] de los enteros ver ecuación (6).

$$x_n' \in [0,255] \rightarrow \mathrm{I} \tag{6}$$

Teniendo en cuenta lo anterior, en la figura 2 antes citada se muestran las parábolas de trasformación, del universo de caracteres ASCII.

3.5 Red de Feistel

Una red de Feistel se puede analizar como un cifrado de flujo que procesa datos de longitud pequeña, a los más del tamaño del bloque de datos. En este pequeño cifrador de flujo la señal de ruido se genera a partir de una función NO lineal cuyos parámetros son una clave y parte del bloque de datos a cifrar. En esta red de Feistel un bloque de datos se divide en dos partes, la parte izquierda (L) y la parte derecha (R). Cuando el tamaño de L es igual al tamaño de R, se dice que la red es balanceada y cuando son diferentes se dice que es desbalanceada.

La figura 8 muestra ente aspecto de la res de Feistel como cifrador de flujo.

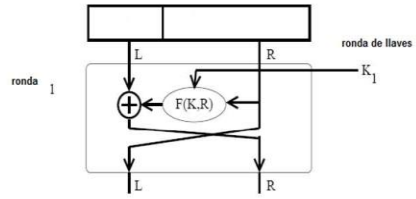


Figura 8. Red de Feistel como cifrador de flujo.

Un cifrador de bloques emplea varias rondas de una red de Feistel y emplea en cada ronda una clave distinta derivada de la clave original. Ver figura 9.

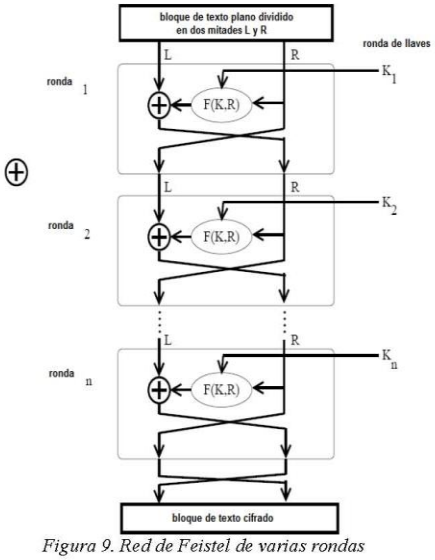


Figura 9. Red de Feistel de varias rondas

4. Desarrollo

4.1 Algoritmo propuesto.

El algoritmo propuesto se deriva de aquel que propone Ljupco Kocarev y puede ser expresado por las figuras 10 y 11.

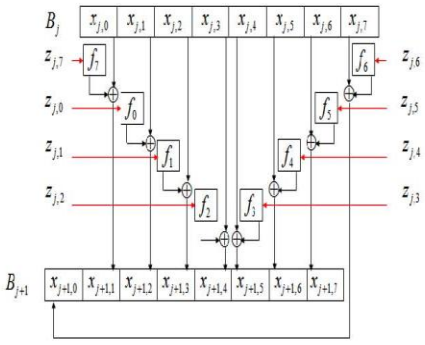


Figura 10. Diagrama del proceso de cifrado

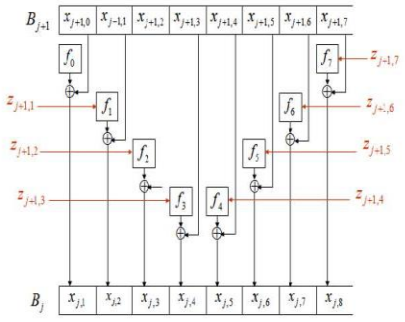


Fig. 11 Diagrama del proceso de descifrado

Se toma un bloque B_0 de texto claro de 64 bits, de la forma siguiente:

$$B_0 = \langle x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3}, x_{0,4}, x_{0,5}, x_{0,6}, x_{0,7} \rangle \quad (7)$$

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

La función de transformación está dada por la siguiente expresión: Podemos observar que el bloque cifrado consiste de r rondas de transformaciones similares aplicadas en una secuencia al bloque de texto plano. La transformación de cifrado esta dada por:

$$x_{i,k+1} = x_{i-1,k} \oplus f_{k-1}\left(x_{i-1,1}, \ldots, x_{i-1,k-1}, z_{i-1,k-1}\right) \quad (8)$$

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

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

...
....

$x_{j+1,7}$

$x_{j+1,0}$

$x_{j+1,1}$ (9)

Los valores $x_{j,1}, x_{j,2}, ...x_{j,n}$ representan los bits del bloque de texto claro, y los valores $x_{j+1,1}, x_{j+1,2}, ...x_{j+1,n}$ representan los bits del bloque de texto cifrado. Para calcular el valor del bloque de texto cifrado se toma el valor anterior. 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 logistica.

Una correcta programación de la función logística como función de transformación del algoritmo, depende de un adecuado proceso de escalamiento y discretización, ya que el intervalo sobre el cual varía ahora la función es de $[0,255]$ abarcando así todos los caracteres del código ASCII extendido.

4.2 Evaluación del Cifrador propuesto

A continuación 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, 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.

La tabla 1. 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.

Tabla 1. Valores de Entropía del cifrador caótico de bloques Logístico VS AES, DES y Triple-DES

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. La tabla 2. 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.

Tabla 2. Valores de Información Mutua del cifrador caótico de bloques Logístico VS AES, DES, Triple-DES, GOST y Skipjack

Archivo	ALGORITMO					
	AES	Logístico	DES	Triple-DES	GOST	Skipjack
TXT	0.0693	0.1777	0.0689	0.0688	0.0689	0.0688
DOC	0.1432	0.7356	0.1437	0.1443	0.1451	0.1425
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	0.6968	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	0.1150	0.0118	0.0118	0.0119	0.0118
ZIP	0.0111	0.6301	0.0112	0.0112	0.0119	0.0117
PDFZIP	0.0255	0.2728	0.0288	0.0258	0.0295	0.0286
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	0.1099	0.1114
MP3-256K	0.0291	0.0656	0.0297	0.0257	0.0298	0.0296
MP3-320K	0.0257	0.0810	0.0259	0.0258	0.0284	0.0259

Tabla 3. 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.

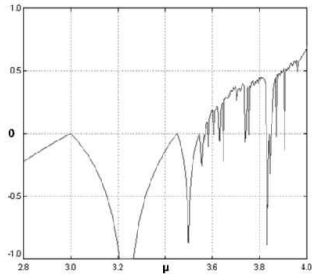


Figura 12. Exponente de Lyapunov del Mapeo Logístico.

En la figura 12 se presume la forma de interpretar el exponente de Lyapunov sobre el Mapeo Logístico este 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.

5. Conclusiones

En este trabajo propusimos implementar un algoritmo de cifrado por bloques caóticos, usando los mapeos caóticos logístico. 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. Se ha trabajado con mapeos generalizados y discretizados que permitan tener como dominio de definición el universo de los símbolos ASCII usados en una computadora.

6. Bibliografía

[1]. A. González, "Dynamical behaviour arising in the adaptive control of the generalized logistic map", Chaos, Solitons & Fractals, Elsevier, Vol. 8, Issue 9, pp. 1485-1488, Sept. 1997.

[2]. M. 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.

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

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

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

[6] 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. Facultad de ciencias, Departamento de Físico Matemáticas Universidad Autónoma de San Luis Potosí.

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

[8]. E. Ott, "Chaos in Dynamical Systems",Cambridge University Press, 2002.

[9]R. M. May and G. F. Oster, "Bifurcation and Dynamic Complexity in Simple Ecological Models", American Naturalist, Vol. 110, Issue 974, Jul-Aug, 1976, pp. 573-599.

[10] Feigenbaum, M. J. "Quantitative Universality for a Class of Non-LinearTransformations." J. Stat. Phys. 19, 25-52, 1978

Curriculum Vitae

Jorge Alberto Martínez Ñonthe

Recibió el título de Ingeniero en Comunicaciones y Electrónica por el Instituto Politécnico Nacional (2005). Obtuvo el Grado de Maestro en Ciencias de Ingeniería en Microelectrónica en la Sección de Estudios de Posgrado e Investigación en la ESIME Unidad Culhuacan del I.P.N (2008), actualmente cursa el primer semestre del Doctorado en Comunicaciones y Electrónica de la misma sección. Desde 2005 ha trabajado en infraestructura de comunicaciones e informática. Sus áreas de interés son la seguridad informática, Computación, comunicaciones y Auditoria informática.

Rubén Vázquez Medina, *Member, IEEE*

Recibió el título de Ingeniero en Electrónica especialidad en Comunicaciones de Septiembre de 1984 a Octubre de 1988 en la Universidad Autónoma Metropolitana y el grado de Maestro en Ciencias especialidad en Ingeniería Eléctrica opción en Telecomunicaciones de Septiembre de 1989 a Septiembre de 1991 en el CINVESTA- IPN, Obtuvo el grado de Doctor en Ciencias en la Universidad Autónoma Metropolitana (2008), fue jefe de la Sección de Estudios de Posgrado e Investigación del 28 de Marzo del 2003 al 17 de agosto del 2006.. Puesto actual profesor de la Sección de Estudios de Posgrado de ESIME Culhuacan.

Cesar E. Rojas López

Profesor de tiempo completo del Instituto Politécnico Nacional, adscrito a la Escuela Superior de Ingeniería Mecánica y Eléctrica, ESIME, Unidad Culhuacan. Realizando el primer semestre de la Maestría en Ciencias en Microelectrónica con especialidad en Seguridad Informática, en la Sección de Estudios de Posgrado e Investigación de la propia ESIME Culhuacan.

Luz M. Benítez Barrón

Nació en México D.F. en 1986 y egreso de la carrera de Ingeniería en Computación por el Instituto Politécnico Nacional unidad ESIME Culhuacan (2008), estudia el posgrado de la Maestría en Ciencias de la Ingeniería en Microelectrónica con especialidad en Seguridad Informática de la Sección de Estudios de Posgrado e Investigación ESIME Culhuacan

APÉNDICE B

Tipos de Cifradores de Flujo

Los cifradores en flujo se dividen normalmente en dos tipos: los síncronos y los autosincronizantes. A continuación se describen ambos tipos.

A) Cifradores de flujo síncronos

Un generador de secuencias pseudo-aleatorias se puede dividir en una parte conductora *F* que es la que cambia de estado según una determinada regla, y una función de estado no lineal f que hace que la secuencia producida por la parte conductora sea más impredictible, tal como se puede ver en la figura B.1.

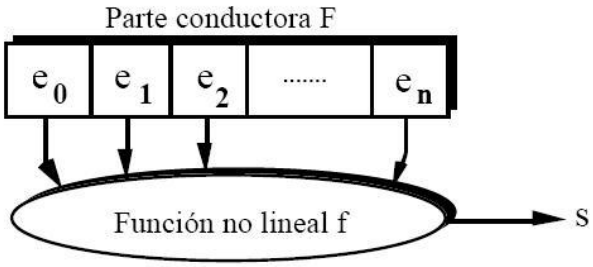


Figura B.1 Estructura de un generador de secuencias pseudo-aleatorias

Por tanto, el proceso de generación del flujo de bits de clave se puede representar como $G = (F, f)$. Puesto en notación abreviada, donde se establece la relación funcional entre la clave k y el flujo de claves s^P:

$$G : k \to s^P \qquad s^P = G(k) \tag{33}$$

donde k representa una clave escogida de un espacio de claves K, y E denota el espacio de estados internos del generador (sus distintos estados internos son los elementos: e_0, e_1, e_2,..., donde e_0 es el estado inicial).

En un cifrador de flujo síncrono, el flujo de claves (*keystream*) se genera independientemente del flujo de caracteres del mensaje (y de la secuencia de texto cifrado), tal como se muestra en la figura B.2.

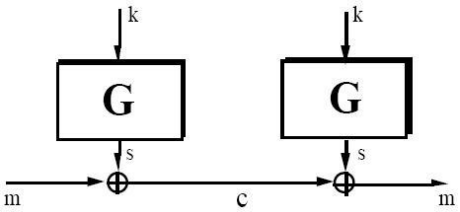


Figura B.2 Cifrador en flujo síncrono.

La generación de la secuencia de claves está gobernada por dos reglas:

$$e_{i+1} = F(k, e_i)$$
$$s_i = f(k, e_i)$$

Los cifradores de flujo síncronos se pueden dividir a su vez de acuerdo con el modo que operan:

A1) Modo contador

$$e_{i+1} = F(e_i)$$
$$s_i = f(k, e_i)$$

En este caso el siguiente estado de la parte conductora no depende de la clave pero está garantizado que pasará a través de todos (o la mayoría) de los estados del espacio de estados. Ejemplos de tales funciones conductoras son los contadores y los LFSR de longitud máxima. La fortaleza criptográfica reside, sin embargo, en la función de estado no lineal de salida *f*.

A2) Modo de realimentación de salida o de realimentación interna

En este modo se cumple que:

$$e_{i+1} = F(k, e_i)$$
$$s_i = f(k, e_i)$$

Ahora la función de estado no lineal f no depende de la clave. Frecuentemente este método consiste simplemente en usar un bit del estado actual como predicado para la realimentación (por ejemplo, el bit menos significativo o de paridad).

A veces se usa una variante de este modo donde la clave k determina sólo el estado inicial:

$$e_0 = k$$
$$e_{i+1} = F(e_i)$$
$$s_i = f(e_i)$$

En los sistemas de flujo síncronos, si un carácter del texto cifrado se pierde, tanto el emisor como el receptor deben resincronizar sus generadores de claves. Si se pierde la sincronización, debe recuperarse inicializando todo el sistema, pues no se recupera por sí sola. Esto garantiza que ante cualquier ataque se desincronizará y no se podrá alterar el contenido del mensaje.

Por tanto, los cifradores en flujo síncronos son inmunes a ataques activos de inserción, eliminación y repetición de parte de la secuencia del texto cifrado, garantizando así la integridad de la información.

B) Cifradores de flujo autosincronizantes

A diferencia de los anteriores, en los que cada nuevo símbolo de salida del generador de secuencias pseudo-aleatorias venía determinado sólo en función del estado interno y de la clave, en los cifradores autosincronizantes el nuevo estado (y, por tanto, la salida del generador) depende además de N símbolos previos del texto cifrado, tal como se ve en la figura B.3. El modo más común de cifrador de flujo autosincronizante es el de realimentación de texto cifrado, que es el que muestra la figura B.3, y que cumple que:

$$e_i = F\left(c_{i-1}, c_{i-2}, \ldots, c_{i-N}\right)$$
$$s_i = F\left(k, e_i\right)$$

La fortaleza criptográfica reside en la función de estado no lineal f. Hay que notar que la entrada (los símbolos del texto cifrado) y la salida (el flujo de clave) de la función de estado no lineal f son conocidas por el criptoanalista en un ataque con texto en claro conocido.

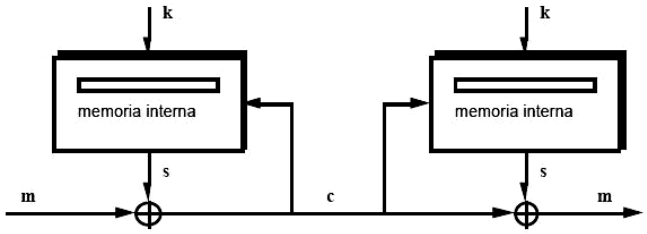


Figura B.3 Cifrador en flujo autosincronizante.

Con este tipo de sistemas, si se altera o se pierde un bit (o carácter) del texto cifrado antes de que éste llegue al receptor, el error se propagará durante los N bits (o caracteres) descifrados siguientes, pero el cifrador se resincronizará una vez reciba de nuevo N bits (o caracteres) correctos del texto cifrado. Una propiedad muy importante de este tipo de cifradores es que la secuencia de salida no es periódica debido a la dependencia que tiene el texto cifrado con los caracteres precedentes (y, por tanto, con el mensaje, el cual será con toda seguridad no periódico). Las ventajas del método autosincronizante son: capaz de recuperarse de los errores que se hayan producido en el canal debido a que el tamaño de su memoria interna es limitado, y

protege al mensaje contra ataques basados en archivos de cifrado, por lo que estará protegido contra búsquedas. Sus desventajas son la dificultad que presentan para ser analizados, incluso por el propio diseñador (ya que todas sus características dependen del mensaje que se transmite), y el hecho de que no son capaces de proteger contra posibles ataques activos de inserción, eliminación y repetición de textos. Siempre será posible obtener un sistema autosincronizante a partir de otro síncrono, simplemente modificando el punto de realimentación del registro interno del sistema.

APÉNDICE C

Modos de operación de los cifradores de bloques

En criptografía, un cifrador por bloques opera en bloques de tamaño fijo, a menudo de 64 o 128 bits. Para cifrar mensajes de mayor tamaño se usan diferentes modos de operación. Los primeros modos descritos, como ECB, CBC, OFB y CFB, aseguraban la confidencialidad, pero no aseguraban la integridad del mensaje. Otros modos han sido diseñados para asegurar la confidencialidad y la integridad del mensaje, como modo CCM, modo EAX y modo OCB.

Modo ECB (Electronic codebook)

El método más simple de modo de cifrado es el llamado ECB (Electronic codebook), en el cual el mensaje se divide en bloques, cada uno de los cuales es cifrado de manera separada. La desventaja de este método es que bloques idénticos de mensaje sin cifrar producirán idénticos textos cifrados. Por esto, no proporciona una auténtica confidencialidad y no es recomendado para protocolos criptográficos.

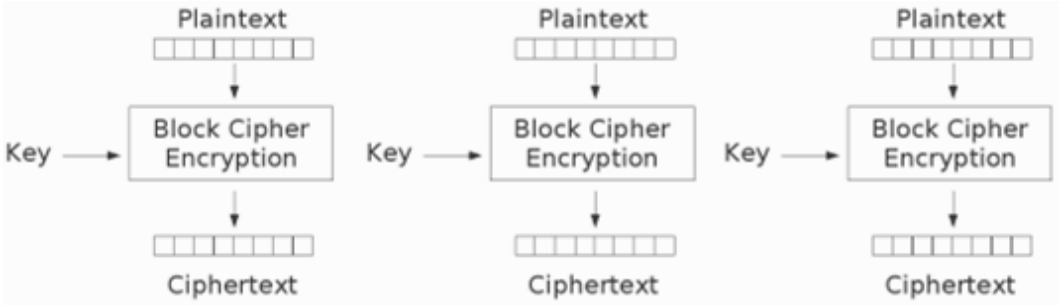


Figura C.1 Modo de cifrado ECB.

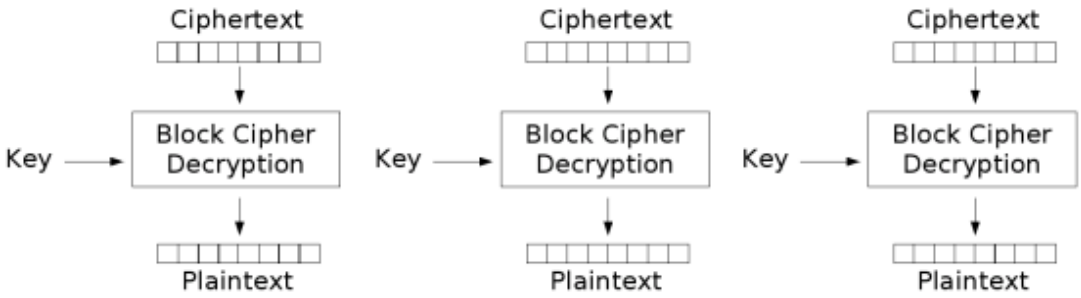


Figura C.2 Modo de descifrado ECB.

Modo CBC (Cipher-block chaining)

En el modo CBC (Cipher-block chaining), antes de ser cifrado, a cada bloque de texto se le aplica una operación XOR con el previo bloque ya cifrado. De este modo, cada bloque es dependiente de todos los bloques de texto planos hasta ese punto. Además, para hacer cada mensaje único se puede usar un vector de inicialización.

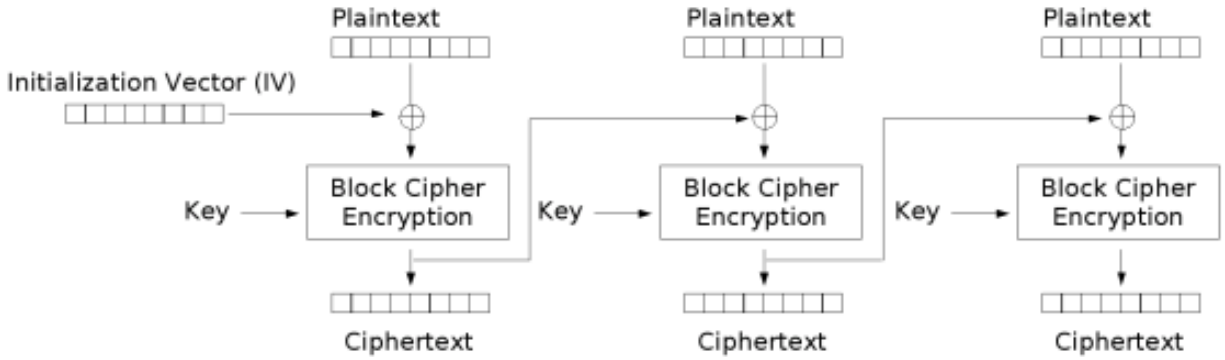


Figura C.3 Modo de cifrado CBC.

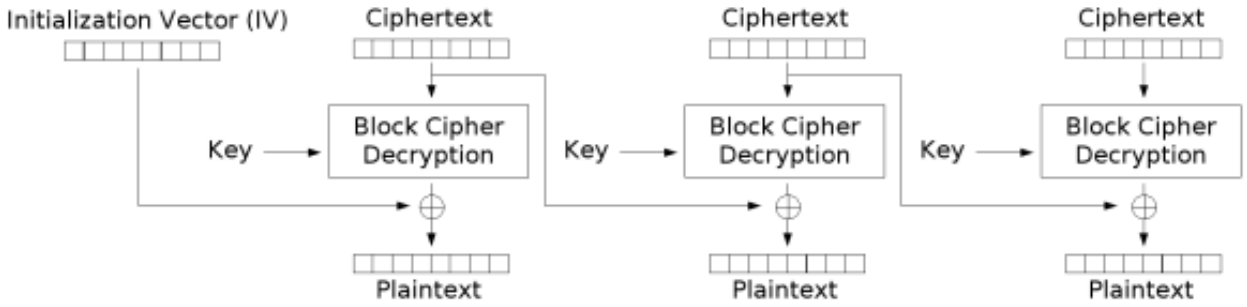


Figura C.4 Modo de descifrado CBC.

CBC es el modo usado más a menudo. Su principal contrapartida es que es secuencial y no puede funcionar en paralelo.

Modo CFB (Cipher FeedBack)

Es posible convertir cualquier cifrador de bloque en un cifrador de flujo usando el modo de realimentación de cifrado (CFB). Uncifrador de flujo elimina la necesidad de completar el último bloque de un mensaje para que tenga la longitud requerida. También puede operar en tiempo real. Así, si se

está transmitiendo una rista[7] de caracteres, se puede cifrar y transmitir inmediatamente cada carácter usando un cifrador de flujo orientado al mismo.

Una propiedad deseable de un cifrador de ristra es que el texto cifrado sea del mismo tamaño que el texto claro. Así, si se están transmitiendo caracteres de 8 bits, los caracteres deberían cifrarse usando 8 bits. Si se usaran más de 8 bits se estaría malgastando capacidad de transmisión.

La figura C.5 muestra el esquema del CFB. Se asume que la unidad de transmisión es s bits; un valor común es s=8. Con el CBC las unidades de texto claro se encadenan juntas, de manera que el texto cifrado de cualquier unidad de teto claro es función de todos los textos claros anteriores.

La entrada a la función de cifrado es un registro de desplazamiento de 64 bits que al principio se inicializa con un vector (IV). A los s bits más a la izquierda (más significativos) de la salida de la función de cifrado se les aplica un XOR con la primera unidad de texto claro P_i para producir la primera unidad de texto cifrado C_i, que luego se transmite. Además los contenidos del registro de desplazamiento se mueven s bits a la izquierda y se coloca C_i en los s bits más a la derecha (menos significativo) del registro de desplazamiento. Este proceso continúa hasta que se hayan cifrado todas las unidades del texto claro.

Para descifrar se usa el mismo esquema, excepto que la unidad de texto cifrado recibido se aplica el XOR con la salida de la función de cifrado para producir la unidad de texto claro. Nótese que se usa la función de *cifrado* no la descifrado.

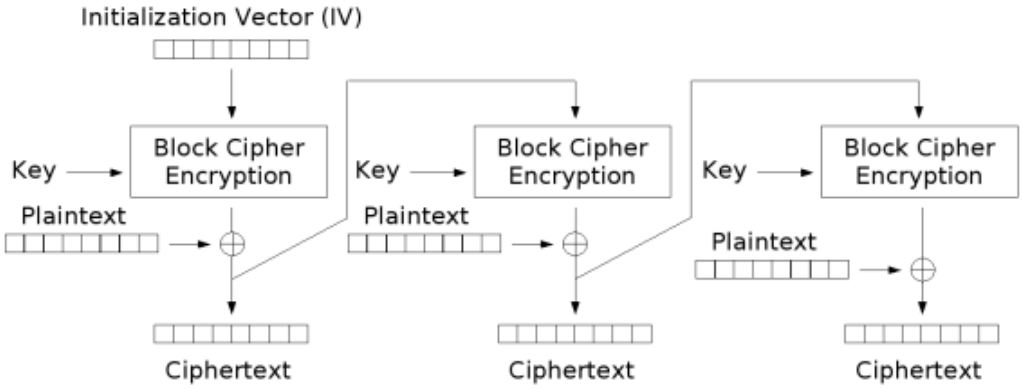


Figura C.5 Modo de cifrado CFB.

[7] f. coloq. Conjunto de ciertas cosas colocadas unas tras otras. RAE.

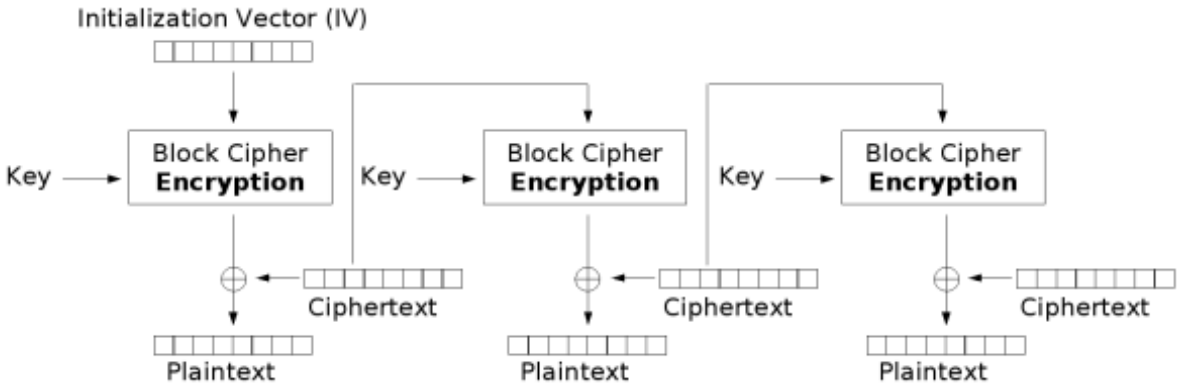


Figura C.6 Modo de descifrado CFB.

Modo OFB (Output FeedBack)

Este modo es similar al modo CFB, véase figura C.7 con la diferencia que la realimentación de la señal se realiza antes de la operación XOR. En este modo la propagación de un error afecta solo a un byte, el que se realimenta en el registro de desplazamiento, mientras que en el modo CBF, la propagación de un error abarca todo el bloque actual de cifrado.

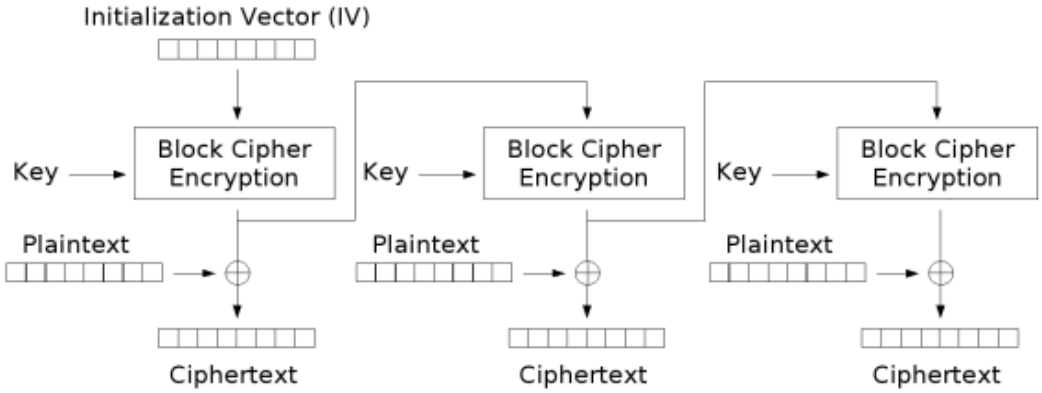


Figura C.7 Modo de cifrado OFB.

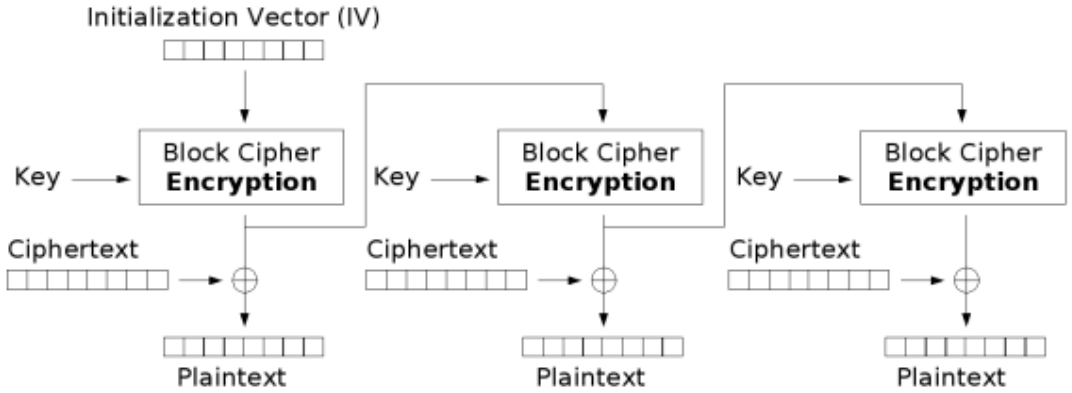


Figura C.8 Modo de descifrado OFB.