Universidad Autónoma Metropolitana
Unidad Azcapotzalco

División de Ciencias Básicas e Ingenierías

Licenciatura en Ingeniería en Computación

Modelación del cifrado de una imagen a color usando un autómata celular bidimensional

Modalidad: Proyecto tecnológico

Mauricio Moreno Escalera
Matrícula: 207204398

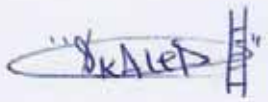
M. en C. Germán Téllez Castillo
Departamento de Sistemas
Profesor asociado

Trimestre 2018 Otoño
Fecha de entrega: 17 de diciembre de 2018

Declaratoria

Yo, Germán Téllez Castillo, declaro que aprobe el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Yo, Mauricio Moreno Escalera, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, asi como en el Repositorio Institucional de UAM Azcapotzalco.



RESUMEN

En este proyecto se propone un esquema de cifrado de imágenes a color basado en la combinación de un autómata celular reversible (ACR) combinado con un algoritmo caótico. El protocolo de cifrado consta de dos fases iterativas: La fase de confusión y la fase de difusión. La primera etapa permuta los pixeles en la imagen usando un mapa caótico discreto conocido como el mapeo del gato (Cat Map), mientras que en la segunda etapa, los valores del pixel (las propiedades de color de cada pixel) son modificados secuencialmente mediante un autómata celular de memoria reversible. Dada la simplicidad de las reglas de autómatas celulares reversibles y de su capacidad de generar desorden o desinformación es posible desarrollar una dinámica reversible que permite el cifrado de imágenes digitales. El esquema propuesto pertenece a la clase de sistemas simétricos. La descripción anterior se codificó en el lenguaje de programación Java y se realizaron diversas pruebas que demuestran la calidad del cifrado ofrecido.

Índice

1. Introducción ... 3
2. Antecedentes ... 5
2.1 Trabajos relacionados ... 5
3. Justificación ... 6
4. Objetivos ... 8
4.1 Objetivo General ... 8
4.2 Objetivos Específicos ... 8
5. Marco Teórico ... 8
5.1 Criptografía ... 8
5.2 Conceptos ... 9
5.2.1 Criptografía moderna ... 10
5.2.2 Criptosistema ... 11
5.2.3. Mapas discretos caóticos ... 13
5.2.4 Autómatas celulares ... 14
5.2.5 Autómatas celulares bidimensionales ... 18
5.2.6 Autómatas celulares reversibles ... 19
6. Desarrollo del proyecto ... 23
6.1 Descripción de la solución propuesta ... 23
6.2 Fase de Cifrado ... 23
6.2.1 Primera Fase del cifrado – Confusión (Cat map). ... 23
6.2.2 Segunda fase del cifrado – Difusión (ACR) ... 25
6.3 Dinámica de cifrado ... 32
6.4 Fase de descifrado ... 34
6.4.1 Ciclo de descifrado ... 35
6.4.2 La clave del descifrado ... 37
7. Resultados ... 40
8. Análisis y discusión de resultados ... 44
8.1 Análisis de simulaciones ... 48

9. Conclusiones 51

10. Bibliografía 52

Ilustración 1. Pasos que realiza Cat Map en cada iteración. 14
Ilustración 2. Tipos de vecindades más utilizadas en autómatas celulares a) Von Neumann y b) Moore 18
Ilustración 3. Partición del espacio celular en la vecindad de Margolus 20
Ilustración 4. Imagen muestra de una imagen a color de tamaño 124 x 124 píxeles iterada mediante el algoritmo Cat Map desarrollado. 24
Ilustración 5. Combinación de píxeles al aplicar regla 57 27
Ilustración 6. Combinación de píxeles al aplicar regla 27. 27
Ilustración 7. Ejemplo de cómo se mueven las filas impares al aplicar las reglas 180 y 225 alternadamente. 28
Ilustración 8. Ejemplo de cómo se mueven las filas impares al aplicar las reglas 108 y 198 alternadamente 29
Ilustración 9. Diagrama del proceso de cifrado. 34
Ilustración 10. Diagrama del proceso de descifrado. 39

1. Introducción

Con la llegada de las computadoras personales e Internet, una gran cantidad de datos digitales visuales son almacenados en diferentes medios y se intercambian en varios tipos de redes abiertas hoy en día. Usualmente, esos datos visuales contienen información confidencial o privada. Como consecuencia y teniendo en cuenta este nuevo entorno, existen varios problemas de seguridad asociados con el procesamiento y transmisión de imágenes digitales: es necesario asegurar la confidencialidad, la integridad y la autenticidad de la transmisión de imágenes digitales. Debido a la gran cantidad de datos involucrados en la transmisión de imágenes digitales sólo pueden ser usados protocolos de cifrado simétricos. Para enfrentar estos desafíos, una amplia variedad de protocolos criptográficos han aparecido en la literatura científica.

Los sistemas de datos tradicionales muestran algunos inconvenientes y debilidades en el cifrado de imágenes digitales (por ejemplo, eficiencia de bajo nivel cuando la imagen es de gran tamaño); consecuentemente, estos sistemas tradicionales no son adecuados para el cifrado de imágenes. Al respecto, los mapas caóticos bidimensionales son naturalmente empleados a cualquier imagen digital que pueda ser representada como un arreglo bidimensional de píxeles.

Sin embargo, los algoritmos basados en sistemas dinámicos y basados en el caos, han mostrado tener un rendimiento superior: tienen muchas propiedades tales como la sensibilidad dependiendo de las condiciones iniciales y parámetros del sistema, propiedades pseudoaleatorias, ergodicidad, no periodicidad y transitividad topológica. La mayoría de las propiedades cumplen con algunos requisitos tales como que son sensibles a las claves, la difusión y mezcla en el sentido de la criptografía.

En [1] se sugiere que los protocolos de cifrado de imágenes basados en mapas caóticos deberían componerse de 2 etapas iterativas: Confusión caótica

y la etapa de difusión de píxeles. La confusión permuta los píxeles de la imagen sin cambiar los valores (el color del píxel) usando un adecuado mapa caótico bidimensional tal como el mapeo Baker, mapeo del gato o un mapa caótico estándar.

En la primera etapa, los parámetros del mapa caótico sirven como clave de confusión. En la etapa de difusión, los valores del píxel son modificados secuencialmente de modo que un pequeño cambio en el valor de solamente un píxel se extiende a muchos píxeles (efecto avalancha). El valor inicial o el parámetro de control de la función de difusión sirven como la llave de difusión.

La ronda de confusión-difusión se repite varias veces para lograr un resultado satisfactorio nivel de seguridad. En este proyecto la etapa de confusión se llevará a cabo mediante el uso del mapeo del gato (Cat map), que muestra buenas propiedades criptográficas y en la etapa de difusión el uso de un adecuado autómata celular reversible (ACR) con buenas propiedades de difusión.

Los autómatas celulares son máquinas de estados finitos formadas por una colección de n unidades de memoria llamadas células. El estado de una célula en particular es actualizado sincrónicamente de acuerdo a la regla especificada cuyas variables son los estados de las celdas vecinas en el paso de tiempo anterior.

Máquinas de estados finitos (y, consecuentemente, autómatas celulares) son ubicuos en el modelado del comportamiento de sistemas integrados de un solo hilo, de tal manera que capturen los estados en que se encarnan los sistemas basados en computadora y las transiciones que realizan. Las máquinas de estados finitos se han utilizado en varias metodologías de ingeniería de software, herramientas y desarrollo y siguen siendo fundamentales para estándares de modelado de software ampliamente soportados. En este sentido, varias aplicaciones para criptografía han aparecido en los últimos años.

2. Antecedentes

La criptografía es un área en constante desarrollo, la seguridad total no existe, lo que hoy es un criptosistema seguro con el transcurrir del tiempo se vuelve inseguro, por esta razón nuevos paradigmas de autentificación y cifrado actualmente son desarrollados. En este apartado revisaremos algunos proyectos desarrollados que tienen relación con este trabajo.

2.1 Trabajos relacionados

Los proyectos tecnológicos relacionados al tema de Criptografía son los siguientes:

- *Transmisión de mensajes en archivos de imágenes y texto usando esteganografía en imágenes GIF [8].* El proyecto pretende manipular texto e imágenes con la finalidad de transmitir un mensaje de texto oculto dentro de una imagen. La relación que tiene este proyecto con el que se pretende desarrollar es que los dos intentan esconder algo, la estenografía pretende ocultar un archivo o mensaje dentro de otro pero, uno de los archivos conserva casi la totalidad de su estructura en este caso la imagen, en cambio el cifrado descompone una imagen y solo teniendo el algoritmo reversible del que cifro la imagen se puede tener el mensaje original.
- *Esteganografía de archivos usando redundancia cíclica en imágenes JPEG [9].* En este trabajo se implementa un software que permite ocultar texto dentro de una imagen aplicando un algoritmo criptográfico junto con un proceso esteganográfico. Al igual que el proyecto anterior éste utiliza la esteganografía para ocultar un mensaje dentro de una imagen de formato JPEG que es el formato que se utilizará, este proyecto se basa en un algoritmo de redundancia cíclica que es un

método diferente pero que logra su propósito, el proyecto que se pretende desarrollar cifra la imagen y el citado solo esconde información.

- *Implementación en software-hardware de aritmética sobre campos finitos binarios F_2^m en curvas elípticas para aplicaciones criptográficas de llave pública [10].* Combinando software, diseño de hardware y matemáticas se utilizan operaciones aritméticas para encriptar mensajes a través de un canal entre dos dispositivos que envían y reciben información. Aunque este proyecto citado no es aplicado a imágenes, cifra información que es lo que el proyecto que se pretende desarrollar intenta, su aplicación será para proteger la información que se envía por cualquier canal de comunicación.

3. Justificación

Las redes públicas como internet no proporcionan un medio de comunicación segura entre entidades. La comunicación en esas redes es susceptible a que terceras personas, sin autorización tengan acceso a ella o la modifiquen. Las grandes industrias necesitan mantener su información segura, los bancos nacionales e internacionales deben asegurar que los canales de comunicación al efectuar transacciones sean seguras, la milicia debe asegurar la integridad de la información y la comunicación.

Debido a la amplia difusión de las cámaras digitales y a su incorporación a los dispositivos móviles, las imágenes digitales constituyen un tipo de data de tránsito común en la Internet. En una imagen digital a color, un conjunto de píxeles dispuestos según una distribución particular sobre una matriz bidimensional representan una imagen. La correlación entre los píxeles vecinos es uno de los indicadores fundamentales de la información contenida en la imagen. Por lo que un método de cifrado robusto aplicado a imágenes digitales debe construir una imagen cifrada en la cual se minimiza la correlación entre los píxeles vecinos, ocultando la información de la imagen original.

Hoy en día gran cantidad de personas cuentan con dispositivos electrónicos como computadoras, teléfonos inteligentes, tabletas electrónicas, etc., que les permiten enviar y recibir información a través de internet pero, no siempre se le da importancia al tema de seguridad y tampoco se piensa quién podría tener acceso a los mensajes, conversaciones, archivos, imágenes, música, etc., sin nuestro consentimiento.

De lo anterior surgió la idea de crear un sistema criptográfico haciendo uso de autómatas celulares aplicados especialmente al tipo de archivo imagen. El sistema tiene la finalidad de que a través del proceso de cifrado, una imagen pueda ser manipulada antes de ser enviada a través de algún canal de comunicación y que una vez recibida la imagen cifrada, esta pueda ser descifrada y se obtenga una imagen idéntica a la original es decir, que no haya pérdida y el archivo pueda ser visto solamente por el receptor al que iba dirigido tal y como el emisor quería que fuese vista.

El cifrado convencional, también denominado encriptamiento simétrico o encriptamiento de clave simple, fue el único tipo de cifrado usado antes del desarrollo del encriptamiento de clave pública. De estos dos tipos principales de cifrado simétrico continúa siendo el más utilizado.

El amplio campo de aplicación del cifrado simétrico ha orientado los esfuerzos de numerosos investigadores impulsando la creación de esquemas de cifrado simétrico basados en los más diversos fundamentos.

Existen sistemas de encriptación que realizan procedimientos similares pero tienen un costo, no solamente en calidad de imagen o pérdidas parciales de información sino que son programas por los que se tiene que pagar cierta cantidad para poder utilizarlo. La ventaja del sistema que se propone es que se desarrollará con software libre y no tendrá costo además de que si se considera que la imagen resultante no tendrá perdidas después del cifrado y transmisión lo hace una buena opción.

4. Objetivos

4.1 Objetivo General

- Diseñar e implementar un autómata celular bidimensional que permita cifrar y descifrar una imagen cuadrada nxn en formato JPEG a color.

4.2 Objetivos Específicos

I. Diseñar e implementar un módulo que manipulará la imagen a nivel de píxeles.
II. Diseñar e implementar un módulo que permita ejecutar la permutación de píxeles en la imagen.
III. Diseñar e implementar un módulo que permita ejecutar una función de proyección de píxeles sobre una matriz.
IV. Diseñar e implementar un módulo que permita hacer la composición de las funciones de permutación y proyección de píxeles.
V. Diseñar e implementar las funciones que regirán la evolución del autómata celular.
VI. Diseñar e implementar las funciones de evolución inversa para el autómata celular.

5. Marco Teórico

5.1 Criptografía

La criptografía (de los vocablos griegos *kriptos*: "ocultar" y *grafos*: "escribir", literalmente "escritura oculta") es el arte o ciencia de cifrar y descifrar información utilizando técnicas matemáticas que hacen posible el

intercambio de mensajes de manera que sólo pueden ser leídos por las personas a quienes van dirigidos.

Con más precisión, cuando se habla de esta área de conocimiento como ciencia, se debería hablar de criptología, término que engloba tanto las técnicas de cifrado, la criptografía propiamente dicha, como las técnicas complementarias de criptoanálisis, que estudian los métodos que se utilizan para romper textos cifrados con objeto de recuperar la información original en ausencia de la clave.

La finalidad de la criptografía es, en primer lugar, garantizar el secreto en la comunicación entre dos entidades (personas, organizaciones, etc.) y, en segundo lugar, asegurar que la información que se envía es auténtica en un doble sentido, que el remitente sea realmente quien dice ser y que el contenido del mensaje enviado, habitualmente denominado criptosistema, no haya sido modificado en su tránsito.

5.2 Conceptos

En la jerga de la criptografía, la información original que debe protegerse se denomina texto plano. El cifrado es el proceso de convertir el texto plano en un galimatías irreconocible, denominado texto cifrado o criptograma. Por lo general, la aplicación concreta del algoritmo de cifrado (también llamado cifra) se basa en la existencia de una clave: información secreta que adapta el algoritmo de cifrado para cada uso diferente.

Las dos técnicas más básicas de cifrado en la criptografía clásica son la sustitución (que supone el cambio de significado de los elementos básicos del mensaje –las letras, los dígitos o los símbolos-) y la transposición (que supone una reordenación de las mismas); la mayoría de los cifrados clásicos son combinaciones de estas dos operaciones básicas. El descifrado es el proceso inverso que recupera el texto plano a partir del criptosistema y la clave.

El protocolo criptográfico especifica los detalles de cómo se utilizan los algoritmos de cifrado y las claves (y otras operaciones primitivas) para conseguir el efecto deseado. El conjunto de protocolos, algoritmos de cifrado, procesos de gestión de claves y actuaciones de los usuarios, en su globalidad es lo que constituyen un criptosistema, que es con lo que el usuario final trabaja e interactúa.

Con frecuencia los procesos de cifrado y descifrado se encuentran en la literatura como encriptado y desencriptado, aunque ambos son neologismos todavía sin reconocimiento académico. Hay quien hace distinción entre "cifrado/descifrado" y "encriptado/desencriptado" según esté hablando de criptografía simétrica o asimétrica, pero la mayoría de los expertos en el mundo académico prefiere evitar ambos neologismos.

5.2.1 Criptografía moderna

La criptografía moderna nace al mismo tiempo que las computadoras, durante la segunda guerra mundial, en un lugar llamado Bletchley Park. Allí un grupo de científicos entre los que se encontraba Alan Turing, trabajaban en el proyecto "ULTRA" tratando de descifrar los mensajes enviados por el ejército alemán, cifrados con los más sofisticados ingenios de codificación ideados hasta ese entonces, la máquina "ENIGMA" y el cifrado Lorenz. Este grupo de científicos diseñó y utilizó el primer computador de la historia, denominado "COLOSSUS", aunque esta información permaneció en secreto hasta mediados de los años setenta.

Desde entonces hasta hoy ha habido un crecimiento espectacular de la tecnología criptográfica, si bien la mayor parte de estos avances se mantenían y se siguen manteniendo en secreto. Financiadas principalmente por la NSA (siglas en inglés de la Agencia Nacional de Seguridad de los EE. UU.), la mayor parte de las investigaciones hasta hace relativamente poco tiempo han sido tratadas como secretos militares.

Esta dualidad civil-militar ha dado lugar a una curiosa doble historia de la criptografía, en la que los mismos algoritmos eran descubiertos, con pocos años de diferencia, por equipos de anónimos militares y posteriormente por matemáticos civiles, alcanzando únicamente estos últimos el reconocimiento público por sus trabajos. Sin embargo, en los últimos años, investigaciones serias llevadas a cabo en universidades de todo el mundo han logrado que la criptografía sea una ciencia al alcance de todos y que se convierta en la piedra angular de asuntos tan importante como el comercio electrónico, la telefonía móvil, o las nuevas plataformas de distribución de contenidos multimedia.

5.2.2 Criptosistema

Se define un criptosistema como una quíntupla (M, C, K, E, D), donde:

- M representa el conjunto de todos los mensajes sin cifrar (lo que se denomina texto plano.
- C representa el conjunto de todos los posibles menajes cifrados o criptogramas.
- K representa el conjunto de claves que se pueden emplear en el criptosistema.
- E es el conjunto de transformaciones de cifrado o familia de funciones que se aplica a cada elemento de M para obtener un elemento de C. Existe una transformación diferente E_k para cada valor posible de la clave K.
- D es el conjunto de transformaciones de descifrado, análogo a E.

Todo criptosistema ha de cumplir la siguiente condición:

$$D_k\big(E_k(M)\big) = M$$

Es decir, que si un mensaje M, se cifrará empleando la clave K y luego se descifrará empleando la misma clave, se obtendría de nuevo el mensaje original M.

Existen dos tipos fundamentales de criptosistemas:

- ***Criptosistemas simétricos o de clave privada***. Son aquellos que emplean la misma clave K tanto para cifrar como para descifrar. Presentan el inconveniente de que para ser empleados en comunicaciones la clave K debe estar tanto en el emisor como en el receptor, lo cual lleva a preguntarse cómo transmitir la clave de forma segura.

- ***Criptosistemas asimétricos o de llave pública***. Emplean una doble clave (Kp, KP) en donde a Kp se conoce como clave privada y KP se conoce como clave pública. Una de ellas sirve para la transformación E de cifrado y la otra para la transformación D de descifrado. En muchos casos son intercambiables, es decir, si se emplea una para cifrar la otra sirve para descifrar y viceversa. Estos criptosistemas deben asegurar además que el conocimiento de la clave pública KP, no permita calcular la clave privada Kp ofreciendo un abanico superior de posibilidades, pudiendo emplearse para establecer comunicaciones seguras por canales inseguros puesto que únicamente viaja por el canal la clave pública.

5.2.3 Sistemas dinámicos

Es importante conocer las bases teóricas implícitas en un Sistema Dinámico y para ello hay que tener claro el concepto de Sistema. Para un físico, un Sistema es un objeto o conjunto de objetos reconocibles que pueden ser considerados como un todo, como por ejemplo una caja llena de átomos, un grupo de animales en su ambiente natural o equipos conectados en una red. Por su parte, un matemático diría usando una definición más precisa, que un Sistema es un conjunto de estados y un conjunto de reglas que actúan sobre esos estados.

Un Sistema Dinámico es un conjunto de estados con reglas que hacen que estos estados cambien en el tiempo. Se puede pensar en el tiempo como un

conjunto creciente de números enteros (valores discretos) o números reales (valores continuos).

5.2.3. Mapas discretos caóticos

El mapeo discreto del gato de Arnold (Arnold's Cat map)[2] es un sistema dinámico simple y discreto que se extiende y "pliega" las trayectorias en el espacio de fases, que es una característica típica de los procesos caóticos, específicamente, el mapeo del gato es el mejor ejemplo conocido de difeomorfismo de Anosov, que es una aplicación que posee aplicación inversa pero, son diferenciables. Éste es un mapeo caótico invertible dado por la siguiente transformación:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & a \\ b & ab+1 \end{pmatrix} \cdot \begin{pmatrix} x_n \\ y_n \end{pmatrix} \cdot (mod\ 1)$$

Donde a, b son parámetros de control y la notación $x(mod\ 1)$ representa la parte fraccionaria del número real x al restar o agregar un apropiado número entero. El mapeo del gato es no hamiltoniano, no analítico y mezclable. Como el determinante de su matriz de transformación lineal es igual a 1, también es conserva su área.

Propiedades del Cat map

- Es invertible porque su matriz determinante es 1 y por lo tanto su inverso tiene entradas enteras.
- Conserva su área original.

- Tiene un único punto hiperbólico (los vértices del cuadrado). La transformación lineal que define el mapa es hiperbólico; es un ejemplo bien conocido de un automorfísmo toral hiperbólico.

- Es ergódico y mezclable
- Es un difeomorfismo Anosov y en particular es estructuralmente estable.

Incluido a continuación se ilustra visualmente y a groso modo, los pasos básicos necesarios para que la imagen se modifique en cada iteración que realiza el algoritmo pudiendo después de determinado número de repeticiones ensamblarse nuevamente.

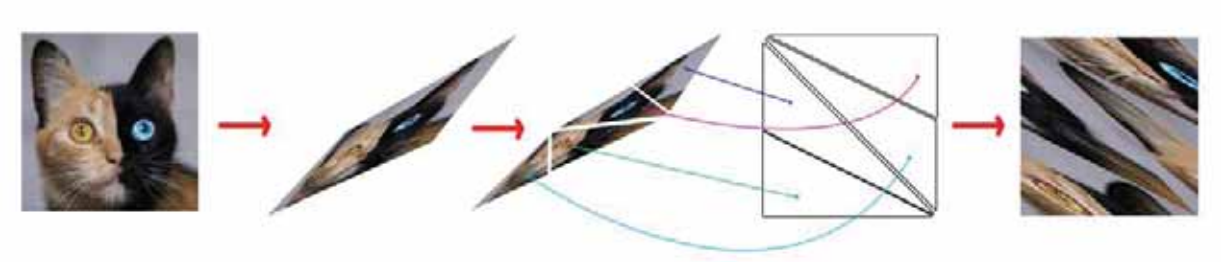


Ilustración 1. Pasos que realiza Cat Map en cada iteración.

5.2.4 Autómatas celulares

Los Autómatas Celulares pueden ser estructurados para el modelado de Sistemas Dinámicos donde el espacio, tiempo y estados son discretos. Estos están representados por un conjunto *n-dimensional* de celdas organizadas geométricamente según la conveniencia del problema. Cada una de las celdas tiene un estado específico en un momento dado, valor que viene determinado según el alfabeto del Autómata Celular. La dinámica de cambio del autómata es bastante simple, a cada celda del espacio celular se le aplica un proceso de transición a partir de su estado actual y del estado de sus celdas vecinas para ese mismo instante, el número de celdas vecinas y la forma como ellas son seleccionadas es parte de la configuración del autómata y se conoce con el nombre de vecindad.

Desde el punto de vista de sistemas de cómputo se suelen considerar las siguientes correspondencias entre un modelo de cómputo tradicional y el de un autómata celular:

- *Las reglas de transición*. Corresponderán con el "programa" que en este caso no deriva de algoritmo alguno.
- *La dinámica del autómata celular*. Es comparable al proceso de ejecución temporal del programa.
- *La configuración o estado inicial*. Es la data inicial o de entrada del programa.
- *Los estados o configuraciones sucesivas*. Son las sucesivas etapas de cómputo intermedio.
- *El estado final*. Generalmente un atractor de la dinámica, sería el cómputo objetivo.

De manera más precisa un Autómata Celular se puede concebir como una tupla o lista ordenada $\boldsymbol{\{G,\ V,\ Q,\ F\}}$ donde:

- $\boldsymbol{G}$ representa el espacio celular.
- $\boldsymbol{V}$ es alguna región local finita dentro del reticulado, definida por un patrón de vecindad a un sitio o nodo. Esta vecindad $\boldsymbol{\{i_1,\ i_2,\ldots,\ i_n\}}$ de sitios tiene, en general, el mismo tamaño y estructura para todos los nodos del espacio celular.
- $\boldsymbol{Q}$ denota un alfabeto finito sobre el cual toman valores los estados de los sitios.
- $\boldsymbol{F}$ es la función de transición local que asocia la configuración de la vecindad con cada estado de los sitios.

5.2.4.1 *Condiciones de frontera*

Por definición, un AC consta de un espacio celular infinito. Sin embargo, para fines prácticos (modelos de sistemas físicos, llevados a cabo en computadores de memoria finita), se requiere tomar ciertas consideraciones a la hora de implementar un AC en un sistema de cómputo. Es por ello que la definición original se modifica para dar cabida a espacios celulares finitos en los que las células del AC interactúen. Esto conlleva a la consideración extra de lo que debe suceder con aquellas células que se encuentren en los bordes del espacio celular. La implementación de una o varias consideraciones específicas es conocida como condición de frontera.

Dentro del ámbito de los AC, se pueden implementar numerosas condiciones de frontera, de acuerdo a lo que el problema real requiera para su modelado. Un AC puede exhibir las siguientes condiciones de frontera:

- *Frontera abierta*. Se considera que fuera del espacio celular residen células, todas con un valor fijo. En el caso particular del juego de la vida y otros AC con dos estados en su conjunto k, una frontera se dice fría si las células fuera de la frontera se consideran muertas y caliente si se consideran vivas.

- *Frontera periódica*. Se considera al espacio celular como si sus extremos se tocaran. En un espacio celular de dimensión 1, esto puede visualizarse en dos dimensiones como una circunferencia. En dimensión 2, el espacio celular podría visualizarse en tres dimensiones como un toroide.

- *Frontera reflectora*. Se considera que las células fuera del espacio celular reflejan los valores de aquellas dentro de la misma. Así, una célula que estuviera junto al borde del espacio celular (fuera de ella) tomaría como valor el de la célula que este junto al borde del espacio celular, dentro de ella.

- *Sin frontera*. Haciendo uso de implementaciones que hagan crecer dinámicamente el uso de memoria del espacio celular implementado, se puede asumir que cada vez que las células deben interactuar con células fuera del espacio celular, este se hace más grande para dar cabida a estas interacciones. Obviamente, existe un límite (impuesto por la memoria disponible) para esta condición. Es muy importante no confundir esta condición de frontera con la definición original de AC cuyo espacio celular es inicialmente infinito. En el caso de un AC sin frontera, el espacio celular comienza con un tamaño definido y finito, y conforme se requiera va creciendo en el tiempo, lo cual no lo hace necesariamente un modelo más cercano a la realidad, pues si se inicializara el espacio celular aleatoriamente, con esta condición sólo se pueden inicializar las células dentro del espacio inicial finito, mientras que en el caso de la definición

original, en teoría todas las células del espacio celular infinito deberían ser inicializadas.

5.2.4.1 *Estado*

Un autómata celular se construye por una serie de celdas, es decir un arreglo de elementos que se denominan células. Cada célula puede tener un número finito de valores, ya sea un valor entero, una letra o un color, lo que se quiera que represente cada una, a estos valores se les denomina estados, todas las células tienen el mismo número posible de estados.

5.2.4.1 *Función de transición*

La función de transición $\boldsymbol{F}$ se define como:

$$F : Q^n \rightarrow Q$$
$$\left(S_{1,}\, S_2, \dots ,\, S_n\right) \in Q^n \rightarrow F(S_1, S_2, \dots, S_n) \in Q$$

Donde n es la cardinalidad de V, es decir el número de nodos $|V|$, del reticulado, contenidos en la vecindad.

De esta forma la función e transición define una correspondencia desde el producto cartesiano de los estados de las celdas pertenecientes a la vecindad a uno de ellos, esta correspondencia es claramente de muchos a uno. Así el alfabeto α de entrada para el autómata celular consiste en el conjunto de posibles configuraciones de las células de la vecindad:

$$\alpha = Q^n$$

La cardinalidad de α es igual a la cardinalidad de Q elevada a la n:

$$|\alpha| = |Q|^n = k^n$$

Donde se denota por k la cardinalidad del alfabeto Q.

5.2.5 Autómatas celulares bidimensionales

Los autómatas celulares en dos dimensiones evolucionan en el plano cartesiano. Estos se encuentran determinados por dos tipos de vecindades fundamentales, la vecindad de Von Neumann y la vecindad de Moore:

- Vecindad de Moore. Está formada por una célula central y ocho células vecinas alrededor.
- Vecindad de Von Neumann. Suprime las células diagonales y conserva las células ortogonales con respecto a la vecindad de Moore.

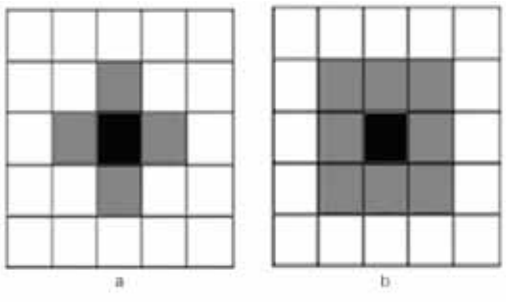


Ilustración 2. Tipos de vecindades más utilizadas en autómatas celulares a) Von Neumann y b) Moore

Es de notar que los autómatas celulares bidimensionales han sido utilizados en numerosas aplicaciones relacionadas con el estudio de sistemas dinámicos reales, tales como:

- Formación de estructuras cristalinas o de patrones específicos. Por ejemplo en reacciones químicas o propagación de fallas en materiales.
- Recurrencia de patrones como en la turbulencia de los fluidos.
- Variedad de sistemas autoreproductivos o evolutivos.
- Propagación de infecciones.
- Procesamiento de señales.

- Modelos económicos.

La complejidad implícita en esta clase de dispositivos es tan grande que su estudio sistemático es muy deficiente haciendo que exista poca literatura que defina una generalización, entre otras cosas esto se debe a que las herramientas que se emplean en una dimensión no son prácticas para aplicarse en dos y tres dimensiones.

En este trabajo se estudia un caso específico de aplicación de los autómatas celulares bidimensionales invertibles, debido a que por sus propiedades desordenan de manera compleja el estado inicial del espacio celular y lo restablecen al seguir la dinámica inversa, lo que resulta útil para construir un método de cifrado.

5.2.6 Autómatas celulares reversibles

Un autómata celular invertible es aquel para el cual, dada su regla dinámica directa, existe una nueva regla inversa cuya aplicación provoca que el autómata recorra las configuraciones, que conforman su trayectoria, en sentido inverso. Es claro que esto es posible sólo si el sistema definido por la regla directa también es determinista para la regla inversa, es decir, que para cada configuración del autómata celular existe una y sólo una configuración precedente.

La propiedad importante que poseen los autómatas celulares invertibles es que la información total sobre la configuración inicial se conserva en todo momento. En tal sentido habrán tantas "constantes de movimiento" (observables que se conservan) como celdas en el sistema. La mayoría de estas constantes de movimiento serán de poco interés ya que en general sólo interesan los rasgos de las celdas activas. En cualquier caso un autómata celular invertible posee el mayor número posible de observables que se conservan, lo que lo convierte en una herramienta plausible para el cifrado de

información debido a que ella podría ser desordenada de manera compleja por reglas invertibles y luego ser recuperada efectivamente invirtiendo la dinámica.

5.2.6.1 *Vecindad de Margolus*

Este tipo especial de vecindad para autómatas celulares bidimensionales fue introducida por Norman Margolus y resulta muy útil en el modelado de sistemas físicos. Para implementar la vecindad de Margolus se define un *autómata celular particionado* de la siguiente manera:

- El arreglo de células (rejilla) se particiona en un conjunto finito de partes disjuntas dispuestas uniformemente y que se denominan bloques.

- Se especifica una regla de bloque que sirve para actualizar un bloque completo en términos de su configuración actual.

- La regla no actualiza una sola celda de bloque, lo hace para todas las células del bloque.

- Los bloques no se superponen de tal forma que no hay intercambio de información entre bloques adyacentes.

- La partición se cambia de una iteración a la otra de manera de permitir el intercambio de información entre los bloques.

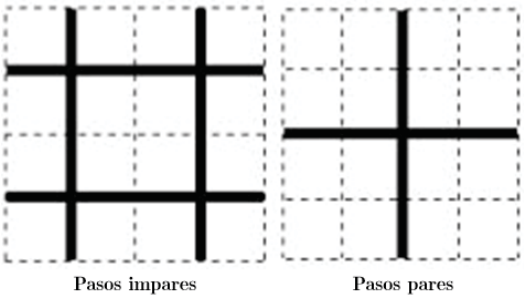


Ilustración 3. Partición del espacio celular en la vecindad de Margolus

En general, se supone que las diferentes particiones empleadas en cada paso de actualización por un autómata celular particionado deben ser finitas en número y deben emplearse (actualizarse) cíclicamente de tal forma de conservar la uniformidad en el espacio-tiempo. En particular en el caso del particionamiento de la vecindad de Margolus se requieren dos pasos de actualización en la regla de bloque (uno aplicado sobre la rejilla par y el otro sobre la rejilla impar) esto implica que un paso de tiempo corresponde a dos iteraciones del autómata celular. Dependiendo de cuál rejilla se esté considerando (par o impar), cambia de punto la vecindad y por ende la regla de bloque se aplica en cada caso a una vecindad diferente.

En los autómatas celulares particionados las reglas se aplican al bloque completo, por ende la operación representada por la regla tendrá tantas salidas como entradas. Con esto es posible asegurar que la regla no pierda información (pueden reconstruirse las entradas a partir de las salidas) dando la posibilidad de programar la invertibilidad.

En cada actualización ninguna de las entradas se comparte con entradas de bloques adyacentes. En esta situación, las reglas determinan el flujo temporal de información de los bloques con control total sobre los mismos. De manera que si la regla individual es invertible, también lo será el proceso global. Por el contrario si hay pérdida de información en un bloque, ninguno de los bloques vecinos podrá compensar esta perdida, por lo que el proceso será no-invertible. Así, una regla en la partición de Margolus será invertible si y solo si esta establece una correspondencia uno a uno entre las configuraciones "viejas" y "nuevas" de cada bloque.

Para invertir la dinámica de un autómata celular particionado en el caso del movimiento de partículas, simplemente se aplica la regla en el orden inverso:

- Orden directo: $Rejilla\ par \rightarrow\ Regla\ \rightarrow Regla\ impar\ \rightarrow Regla$
- Orden inverso: $Rejilla\ impar\ \rightarrow Regla\ \rightarrow Rejilla\ par\ \rightarrow Regla$

5.2.6.2 *Reglas invertibles*

La parte central del estudio de un autómata celular invertible es su regla de evolución, ya que ésta indica cómo se comporta el sistema a través del tiempo, lo interesante es que esta regla de evolución es de influencia local, es decir, sólo afecta las vecindades que forman parte de la configuración, sin embargo induce un mapeo global que es invertible. Desde esta perspectiva se puede tomar la evolución como una función que esencialmente transforma elementos de un conjunto a otro, donde tales elementos son las configuraciones globales que puede tener el autómata celular y la función el mapeo global producto de la regla de evolución.

Un ejemplo del uso de la vecindad de Margolus es el modelaje de un gas de partículas en el cual las partículas se mueven a velocidad constante y uniforme sin interacción. En este caso las células pueden acceder dos estados, cero (ausencia de partícula) y uno (célula ocupada por una partícula), siendo los bloques de tamaño 2x2. Para simular el movimiento de una partícula se aplica una regla bloque muy simple que consiste en intercambiar estados en la diagonal, intercambiando la posición de la rejilla en cada iteración, como se muestra en a continuación.

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \Leftrightarrow \begin{bmatrix} a_{11} & a_{10} \\ a_{01} & a_{00} \end{bmatrix}$$

Las reglas modifican todas las celdas ocupadas en el bloque. En este estudio se emplean bloques de 2x2 y nos concentramos en reglas que realizan desplazamientos de los contenidos de las celdas. De esta forma tras la aplicación de la regla, el bloque contiene las mismas unidades de información, pero distribuidas de forma diferente.

6. Desarrollo del proyecto

6.1 Descripción de la solución propuesta

Desarrollar e implementar en lenguaje Java un proceso de confusión de píxeles (Cat map) junto con una dinámica reversible de autómatas celulares bidimensionales no uniformes como método de cifrado simétrico robusto de imágenes digitales cuadradas en formato JPEG a color

Se establece una imagen digital cuadrada a colores definida por $N\ x\ N$ píxeles de manera que p_{ij} representa el valor numérico del color de para cada $(i,\ j)$ píxel. Sea $I = (p_{ij})$ de la matriz definiendo la imagen digital. Como se menciona en la introducción, el algoritmo criptográfico propuesto in este trabajo consta de 2 fases iterativas: La confusión (Cat Map) y la Difusión (RCA).

6.2 Fase de Cifrado

6.2.1 Primera Fase del cifrado – Confusión (Cat map).

La fase de confusión (Cat Map) se lleva a cabo utilizando un mapa caótico discreto y el número de iteraciones es T_0.

En la fase de confusión los píxeles de la imagen serán permutados usando el mapeo del gato (Cat Map) sobre la red de la imagen:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & a \\ b & ab+1 \end{pmatrix} \cdot \begin{pmatrix} x_n \\ y_n \end{pmatrix} \cdot (mod\ 1)$$

Habiendo desarrollado las herramientas matemáticas para abordar el corazón de esta discusión, un experimento se puede realizar para mostrar la

fiabilidad del mapeo caótico del gato. Para este propósito se desarrolló la primera parte del software y a continuación se muestra un ejemplo para una imagen de 124 x 124 píxeles a color pues se conoce el número de iteraciones que tiene que hacer para regresar a la imagen original.

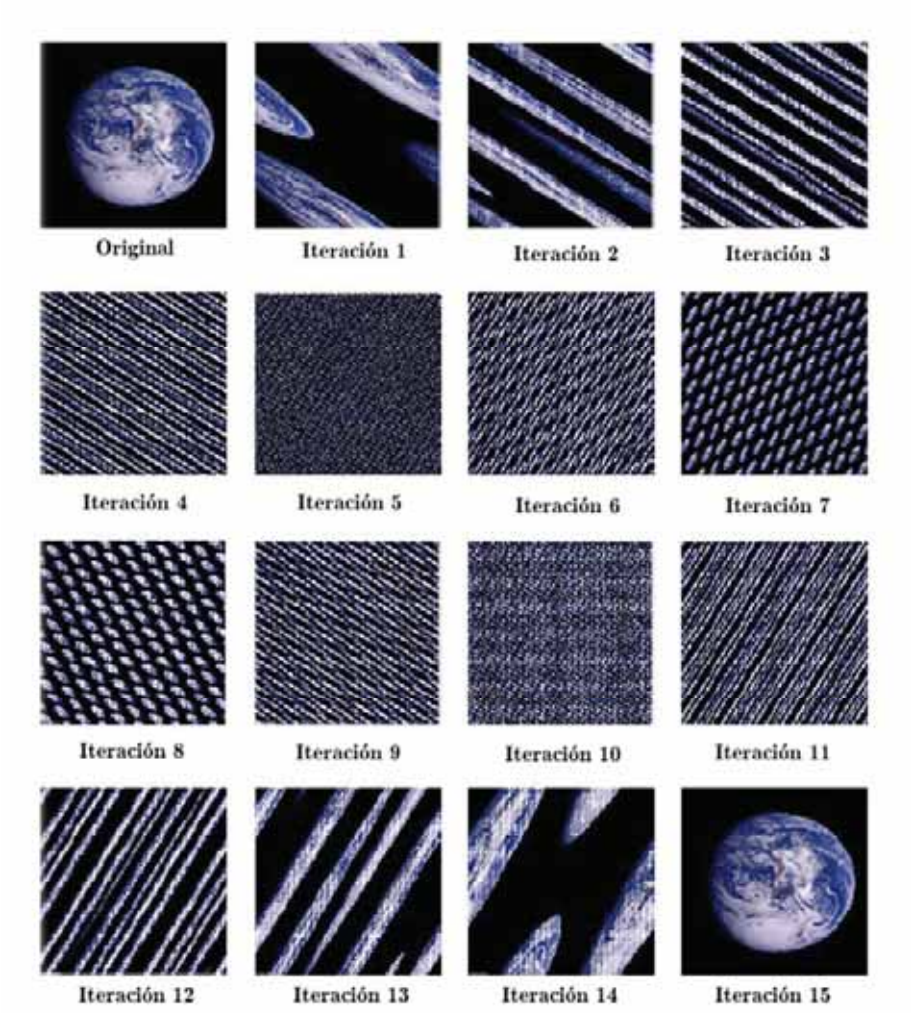


Ilustración 4. Imagen muestra de una imagen a color de tamaño 124 x 124 píxeles iterada mediante el algoritmo Cat Map desarrollado.

Una de las iteraciones que no será la final será elegida al azar y será la imagen utilizada como entrada del software que se encargará de llevar a cabo la difusión de píxeles (ACR) con el fin de hacer más fuerte el proceso de cifrado, imaginemos que para el ejemplo anterior se utilizará la imagen de la iteración número 10 como entrada para el Autómata Celular, si bien ya esa imagen es muy confusa y pareciera que no podría ser descifrada, el autómata celular le agregará mayor complejidad al cifrado de la imagen.

6.2.2 Segunda fase del cifrado – Difusión (ACR)

Del planteamiento del problema se sigue que el esquema propuesto de encriptamiento simétrico para imágenes digitales a color debe estar basado en una dinámica reversible de autómata celular bidimensional no homogéneo y variable en el tiempo.

El primer concepto a tratar es el de vecindades. Se dice que una célula o celda tiene alrededor un conjunto definido de células vecinas en cada espacio de tiempo discreto. Para el problema planteado la adopción de la vecindad de Margolus facilita la definición de funciones de transición reversibles que actúan sobre espacios celulares bidimensionales. Con este tipo de vecindad el espacio celular bidimensional se divide en bloques. En el esquema propuesto cada bloque B está formado por cuatro celdas b_{00}, b_{01}, b_{10}, y b_{11} que están dispuestas de forma tal que en la iteración i conforman una matriz B_i de *2x2* celdas.

$$B^i = \begin{bmatrix} b_{00}^i & b_{01}^i \\ b_{10}^i & b_{11}^i \end{bmatrix}$$

El siguiente concepto es el de función biyectiva. Definimos que una función biyectiva si para cada elemento del conjunto de llegada existe un elemento único en el conjunto de partida y ambos conjuntos tienen el mismo número de elementos. Aplicando esto a las vecindades de Margolus obtenemos que la reversibilidad está garantizada ya que al aplicar la función sobre el bloque resultante obtendremos el bloque original.

En este trabajo se propone un mecanismo que modifica la vecindad de Margulos, al introducir en esta dinámica de autómata celular bidimensional, el parámetro *distancia intercelular: DI*. Durante la ejecución, este parámetro va disminuyendo o aumentando, tomando valores enteros positivos, y particiona el espacio celular C creando un conjunto de *células activas* C_A y un conjunto de células inactivas C_I. Durante la aplicación de una regla sólo las

células activas conforman vecindades de Margolus (bloques) y son transformadas. Las células inactivas permanecen inalteradas.

Esto implica además que el número de células tanto activas como inactivas que puede contener un bloque está en función de la distancia intercelular. Ya que si bien un bloque solo puede contener cuatro células activas, el número de células inactivas entre cada una de las 4 células activas está determinado por *DI*.

La distribución de los bloques puede generarse o bien usando una rejilla par o una impar. En la rejilla par se comienza a hacer efectiva la división del espacio celular en bloques en el punto (0,0). En la rejilla impar, dicho punto es (1,1).

La aplicación conjugada de desplazar y agrandar los bloques asegura que todas las células en el espacio celular serán desplazadas sin importar las dimensiones del espacio celular y contribuye a elevar el grado de desorden en la imagen encriptada.

6.2.2.1 *Reglas del autómata*

Otro punto importante a describir es el efecto de la reglas sobre los bloques. De manera general, se tiene que la clave suministrada al autómata celular determina la forma en que la función de transición F ha de aplicarse sobre el espacio celular y sus respectivos bloques. Definimos una función de transición F como un conjunto de reglas, y definimos a las reglas como un valor decimal cuya representación binaria nos indica la manera en que serán reordenadas las células en un bloque.

Existen 24 funciones biyectivas $F : \{00, 01, 10, 11\} \rightarrow \{00, 01, 10, 11\}$ que pueden ser consideradas a la hora de armar una combinación de reglas que ocasionen un desorden reversible sobre la imagen representada en la configuración inicial de un espacio celular bidimensional.

Por su capacidad de producir desorden fueron seleccionadas las reglas 57 y 27 para dirigir el proceso de cifrado.

Para ilustrar el efecto de las reglas, tomemos como ejemplo gráfico algunas:

La regla 57, corresponde a la función biyectiva de $\{00, 01, 10, 11\}$ en $\{00, 01, 10, 11\}$ definida por:

$$f(00) = 01 \quad f(01) = 10 \quad f(10) = 11 \quad f(11) = 00$$

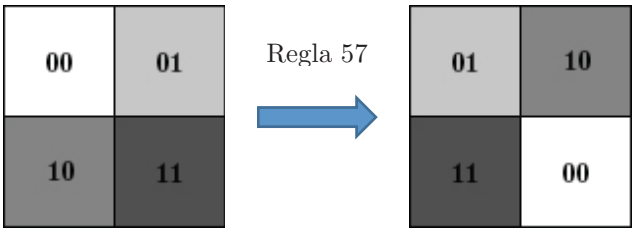


Ilustración 5. Combinación de píxeles al aplicar regla 57

La ilustración 5 muestra cómo todas las células del bloque han cambiado su posición y su vecindad dentro del bloque.

La regla 27, corresponde a la función biyectiva de $\{00, 01, 10, 11\}$ en $\{00, 01, 10, 11\}$ definida por:

$$f(00) = 11 \quad f(01) = 10 \quad f(10) = 01 \quad f(11) = 00$$

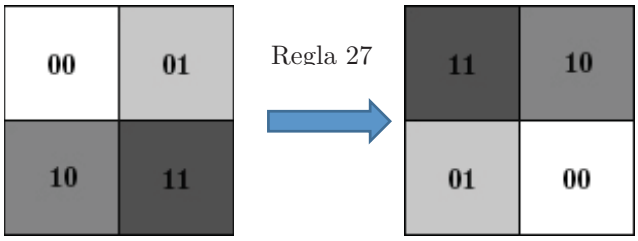


Ilustración 6. Combinación de píxeles al aplicar regla 27.

La ilustración 6 muestra la aplicación de la regla 27. Esta regla efectúa el intercambio de las diagonales dentro del bloque.

La dinámica de cifrado estará fundamentada en la aplicación alternada de las reglas 57 y 27 durante las iteraciones par (rejilla par) e impar (rejilla impar) respectivamente.

Para completar la acción de las reglas 57 y 27, se seleccionaron las reglas 180, 225, 108 y 198.

La regla 180 y 225, corresponde a las funciones biyectivas de $\{00, 01, 10, 11\}$ en $\{00, 01, 10, 11\}$ definida por:

$$Regla\ 180. \quad f(00) = 00 \quad f(01) = 01 \quad f(10) = 11 \quad f(11) = 10$$

$$Regla\ 225. \quad f(00) = 10 \quad f(01) = 00 \quad f(10) = 10 \quad f(11) = 11$$

La aplicación alternada de las reglas anteriores durante las iteraciones par e impar respectivamente, produce un corrimiento (circular) dentro de las filas impares del espacio celular, mientras las filas pares permanecen inalteradas

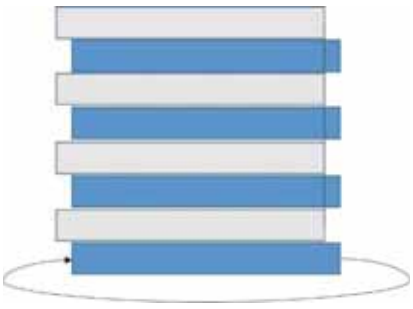


Ilustración 7. Ejemplo de cómo se mueven las filas impares al aplicar las reglas 180 y 225 alternadamente.

La regla 108 y 198, corresponde a las funciones biyectivas de $\{00, 01, 10, 11\}$ en $\{00, 01, 10, 11\}$ definida por:

$$Regla\ 108. \quad f(00) = 00 \quad f(01) = 11 \quad f(10) = 10 \quad f(11) = 01$$

$$Regla\ 198. \quad f(00) = 10 \quad f(01) = 01 \quad f(10) = 00 \quad f(11) = 11$$

La aplicación alternada de las reglas anteriores durante las iteraciones par e impar respectivamente, produce un corrimiento hacia abajo (circular) dentro de las columnas impares del espacio celular, mientras las columnas pares permanecen inalteradas.

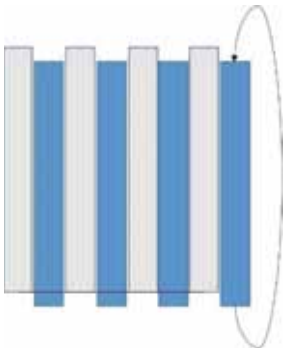


Ilustración 8. Ejemplo de cómo se mueven las filas impares al aplicar las reglas 108 y 198 alternadamente

6.2.2.2 *Ciclo de cifrado*

Al aplicar la dinámica de cifrado sobre un espacio celular de dimensión nxm un $cicloe(x)$ consiste en la aplicación de la secuencia de reglas 180, 225, 57, 27, 108, 198 para cada uno de los valores de DI (distancia intercelular).

$$DI: 0, 1, 2, \dots, maxDI;$$

$$maxDI = \frac{(\max(n, m) - 4)}{3}$$

El $cicloe(x)$ tiene un parámetro x, que toma valores en el conjunto $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$ y que indica cuántas veces debe aplicarse la combinación de reglas 57 y 27. Las combinaciones de regla 180, 225 y 108, 198 se aplican sólo una vez cada una.

Regla	Iteración	Aplicaciones
180 225	Par Impar	1
57 27	Par Impar	Valor del dígito en la clave hexadecimal suministrada
108 198	Par Impar	1

Tabla 1. Aplicación de las reglas en el ciclo de cifrado

Pseudocódigo del cicloe(x)

```
Inicio
    Por cada X en Clave
        Cicloe(X)
        Para cada DI desde 0 hasta (max(n,m)-4)/3
            Dividir C en rejilla par usando DI
            Aplicar regla 180
            Dividir C en rejilla impar usando DI
            Aplicar regla 225
            Para cada xi desde 0 hasta X-1
                Dividir C en rejilla par usando DI
                Aplicar regla 57
                Dividir C en rejilla impar usando DI
                Aplicar regla 27
            Dividir C en rejilla par usando DI
            Aplicar regla 108
            Dividir C en rejilla impar usando DI
            Aplicar regla 198
Fin
```

Donde X es el valor del dígito en la clave hexadecimal suministrada, n y m se refieren al número de células verticales y horizontales presentes respectivamente en el espacio celular y C.

6.2.2.3 *La Clave del cifrado*

En la dinámica de cifrado se usa una clave $K = k_0\, k_1\, k_2 \ldots k_{15}$ de 16 dígitos hexadecimales (o 64 bits). Los primeros 10 dígitos $k_0 k_1 k_2 \ldots k_9$ son usados, uno por uno como parámetros de $cicloe(k_1)$. Los dígitos $k_{10} k_{11} k_{12} k_{13}$ son utilizados como semilla de un algoritmo que produce una secuencia de mxn números pseudoaleatorios que son dispuestos sobre un espacio celular M de dimensión mxn que es sometido al procesamiento determinado por $cicloe(k_{14})$ y $cicloe(k_{15})$. La configuración del espacio celular M obtenida por este procesamiento es usada como máscara que se superpone mediante la operación XOR a la configuración del espacio celular original previamente procesado.

El algoritmo utilizado para la generación de los números pseudoaleatorios que conforman la máscara está descrito en [5] donde se implementa mediante el código que se muestra a continuación.

Código generador de números pseudoaleatorios

```
unsigned int sed;                    /* global variable */
#define a 16807                      /* 7^5 */
#define m 2147483647                 /* 2^31 -1 */
#define q 1277773                    /* m/a */
#define r 2836                       /* m%a */
double
random (void)
        {
        int tmp_seed;
        tmp_seed = a * (seed % q) – r * (seed / q);
        if(tmp_seed >= 0)
                seed = tmp_seed;
        else
                seed = tmp_seed+m;
        return((double) seed) / m);
        }
```

6.3 Dinámica de cifrado

Ahora que se conocen todas las piezas que conforman las piezas que conforman la dinámica de cifrado, es posible ensamblar una descripción paso a paso. Estos pasos se enlistan a continuación.

1. La imagen de nxn píxeles (imagen cuadrada) a cifrar es leída y sus píxeles son dispuestos sobre un espacio celular C de dimensión nxn (imagen cuadrada), conservando las posiciones relativas de los píxeles dentro de la imagen. Adicionalmente se suministra una clave de encriptamiento cuya longitud debe ser $4 < k < 16$ o bien $k_0 k_1 k_2 \dots k_{15}$ que pueden ser elegidos del siguiente conjunto $\{1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$.

2. La imagen seleccionada es procesada por el módulo del mapeo caótico (Cat map) y entrega una nueva matriz nxn de las mismas dimensiones que la original pero con los píxeles confundidos al módulo del Autómata celular.

3. El espacio celular C es transformado por la aplicación de la secuencia $ciclo(K_i);\ i = 0, 1, 2, \ldots, 9$.

4. Tomando como semilla $k_{10}k_{11}k_{12}k_{13}$ se genera una secuencia pseudoaleatoria de nxn píxeles que son dispuestos sobre un espacio celular M de dimensión nxn. Adicionalmente el espacio celular M es transformado por la aplicación de la secuencia $ciclo(K_i);\ i = 14, 15$.

5. $C = C\ xor\ M$

Esta secuencia de pasos, que describen el funcionamiento de la dinámica de cifrado, es esquematizada en la siguiente ilustración. Se usa como ejemplo la (muy conocida) imagen "Lena". Se aprecia en la figura la confusión y difusión que sobre la imagen causa el módulo (Cat map) y después el módulo del autómata celular (AC) que a su vez usa las reglas seleccionadas con diferentes valores de Distancia Intercelular DI.

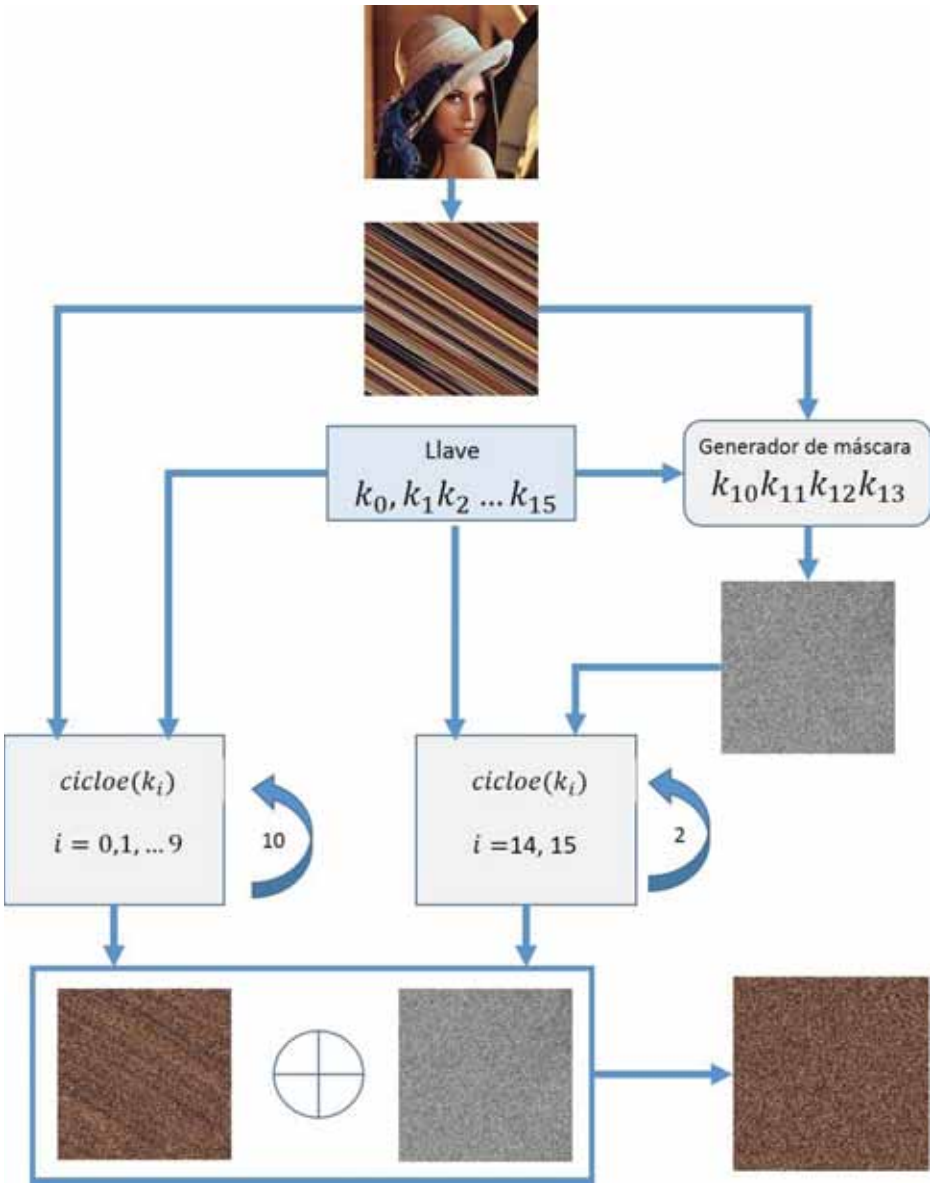


Ilustración 9. Diagrama del proceso de cifrado.

6.4 Fase de descifrado

Para realizar el proceso de descifrado se aplican las mismas transformaciones mencionadas anteriormente, pero en orden inverso. Como transformación inversa al $cicloe(x)$ del proceso de cifrado, en el proceso de descifrado se usará el $ciclod(x)$.

6.4.1 Ciclo de descifrado

Al aplicar la dinámica de descifrado sobre un espacio celular de dimensión nxm, un $ciclod(x)$ consiste en la aplicación de la secuencia de reglas 198, 108, 27, 57, 225 y 180 para cada uno los valores de Distancia Intercelular DI.

$$DI: maxDI, \dots, 2, 1, 0;$$

$$maxDI = \frac{(\max(n, m) - 4)}{3}$$

El $ciclod(x)$ tiene un parámetro x, que toma valores en el conjunto $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$ y que indica cuántas veces debe aplicarse la combinación de reglas 27 y 57. Las combinaciones de regla 198, 108 y 225, 180 sea aplican sólo una vez cada una. A continuación se detalla la forma en que estas reglas son aplicadas durante el $ciclod(x)$ para un valor particular de DI.

El ciclo de descifrado viene dado por la aplicación revertida del $cicloe(x)$, sin ninguna modificación salvo que la regla 57 requiere una consideración particular. Por no ser esta directamente reversible, requiere que para obtener el valor original se sea aplicada 3 veces luego de la primera aplicación. La regla 147, resulta como la regla inversa, que devuelve el orden original tras ser aplicada sobre el resultado de una única aplicación de la regla 57.

Regla	Iteración	Aplicaciones
198 108	Impar Par	1
27	Impar	Valor del dígito en la clave hexadecimal suministrada
57 (3 veces) ó 147	Par	
225 180	Impar Par	1

Tabla 2. Aplicación de las reglas en el ciclo de descifrado

Pseudocódigo del ciclod(x)

```
Inicio
    Por cada X en Clave-1
        Ciclod(X)
        Para cada DI desde (max(n,m)-4)/3 hasta 0
            Dividir C en rejilla impar usando DI
            Aplicar regla 198
            Dividir C en rejilla par usando DI
            Aplicar regla 108
            Para cada xi desde 0 hasta X-1
                Dividir C en rejilla impar usando DI
                Aplicar regla 27
                Dividir C en rejilla par usando DI
                Aplicar regla 147

                /* Nótese que:
                    Aplicar regla 147
                 Es análogo a:
                    Aplicar regla 57
                    Aplicar regla 57
                    Aplicar regla 57
                */

            Dividir C en rejilla impar usando DI
            Aplicar regla 225
            Dividir C en rejilla par usando DI
            Aplicar regla 180
Fin
```

6.4.2 La clave del descifrado

En la dinámica del descifrado se usa la misma clave $K = k_0\, k_1\, k_2 \ldots k_{15}$ de 16 dígitos hexadecimales. Los dígitos $k_{10}k_{11}k_{12}k_{13}$ son utilizados como semilla del algoritmo que produce una secuencia de mxn número pseudoaleatorios. Que son dispuestos sobre un espacio celular M de dimensión mxn que es sometido al procesamiento determinado por $cicloe(k_{14})$ y $cicloe(k_{15})$. La configuración del espacio celular M obtenida por este procesamiento es usada como máscara que se superpone mediante la operación XOR a la configuración del espacio celular de entrada C. Finalmente el espacio celular $C' = C\ xor\ M$ es sometido a la secuencia de transformaciones $ciclod(k_i); i = 9, 8, 7, \ldots, 0$. Es

decir, los dígitos $k_9 k_8 k_7 \ldots k_0$ son usados, uno por uno, como parámetros de $ciclod(k_i)$.

Las mismas operaciones fundamentales usadas anteriormente para realizar el cifrad, son ahora utilizadas para realizar el proceso de descifrado.

La secuencia de pasos, que describen el funcionamiento de la dinámica de descifrado, es esquematizada en el diagrama que se presenta a continuación.

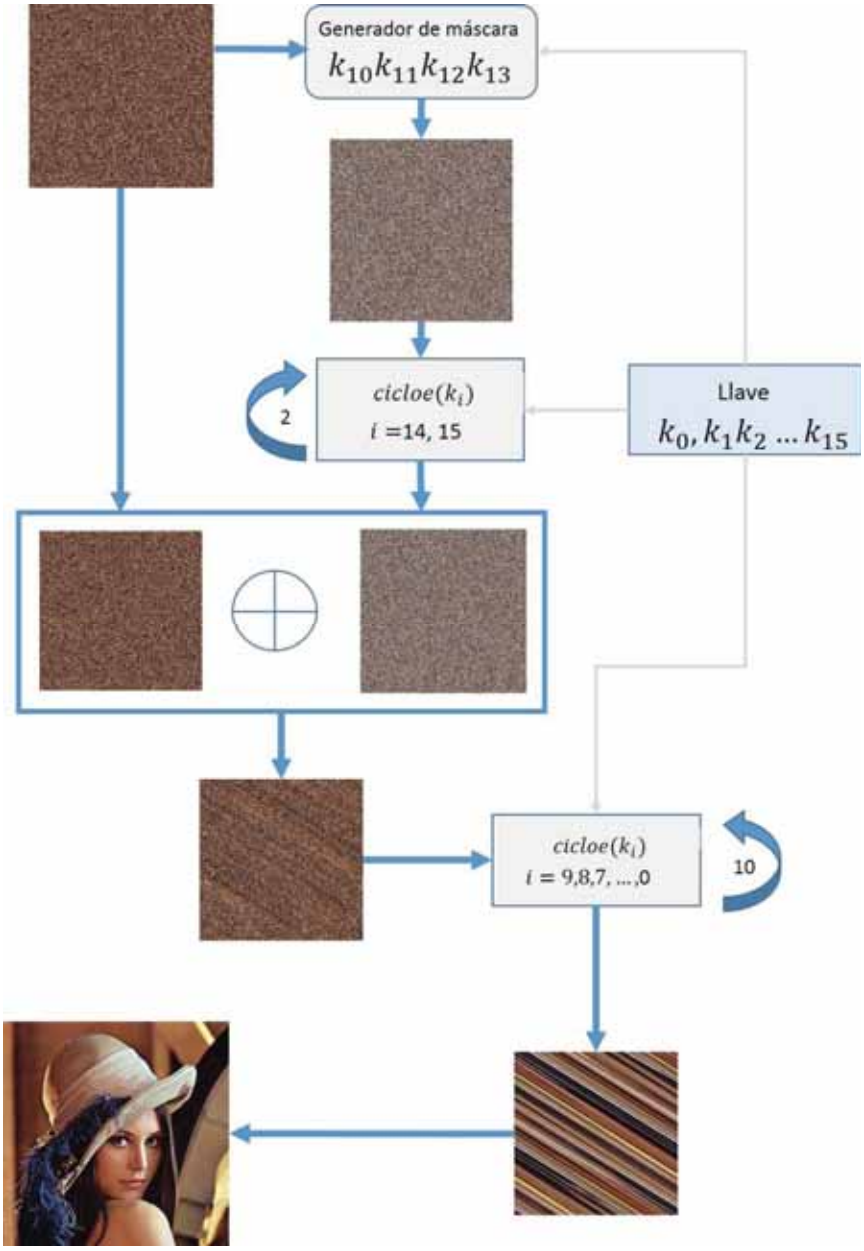


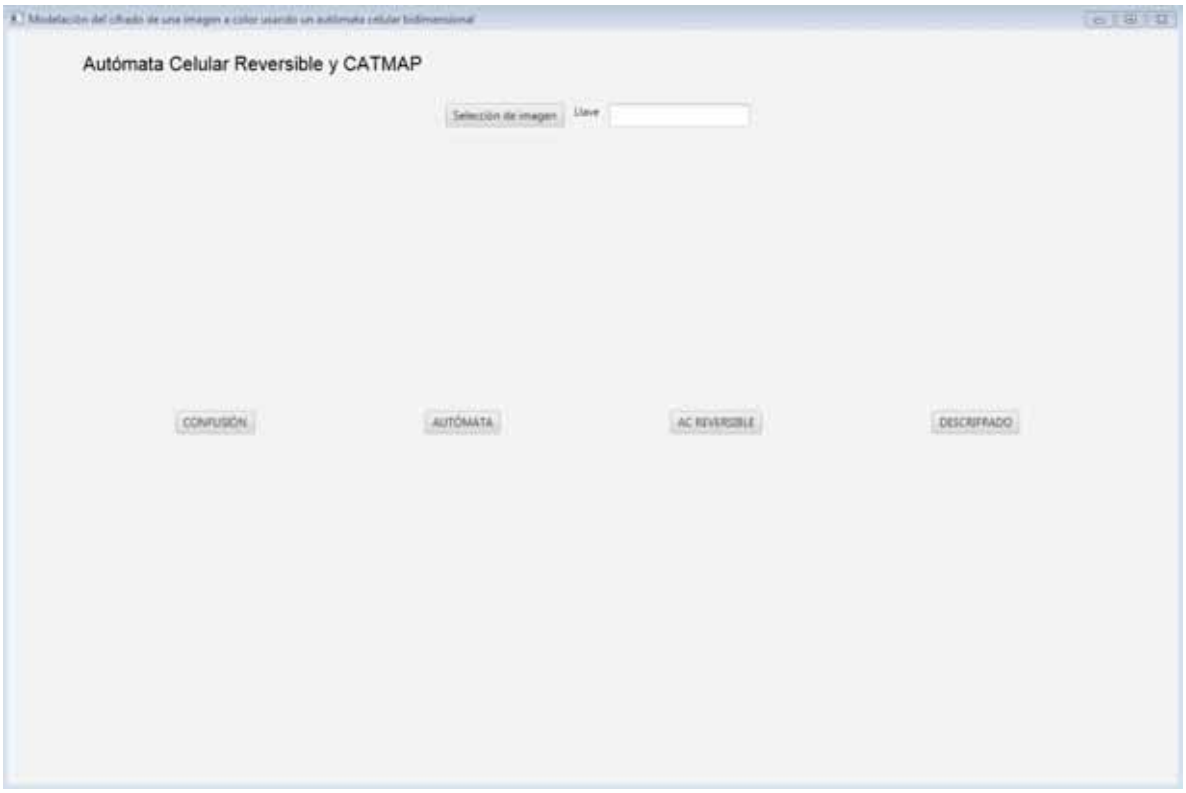
Ilustración 10. Diagrama del proceso de descifrado.

7. Resultados

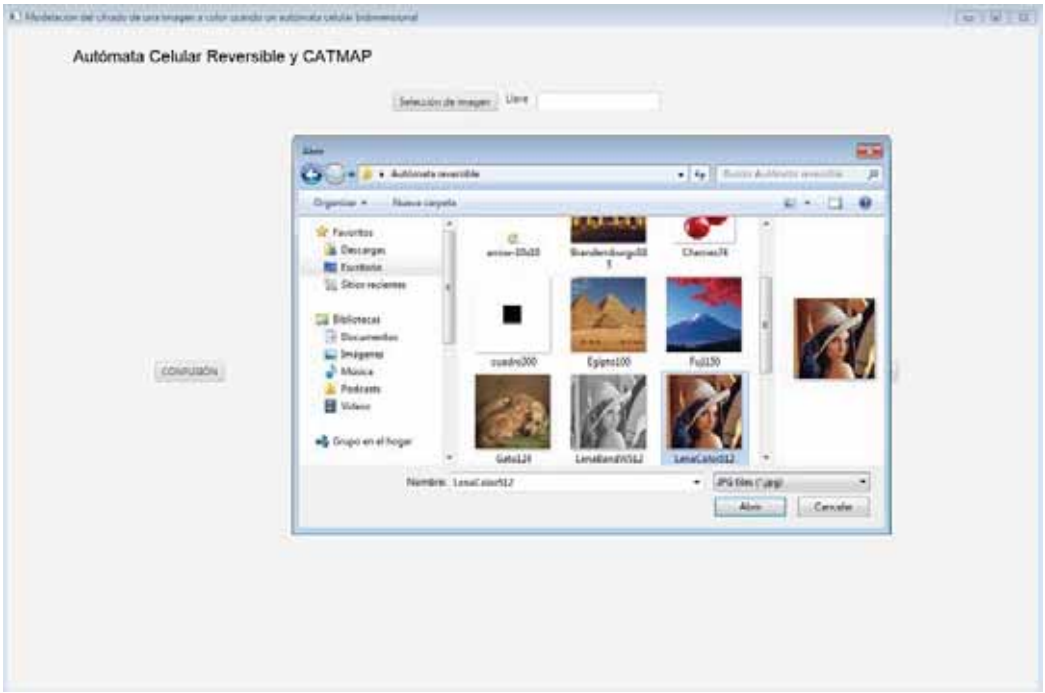
A continuación se muestran los resultados de utilizar el software desarrollado para probar la eficiencia del sistema creado a partir de una serie de algoritmos que por la forma en la que fueron propuestos son programables y desarrollando una interfaz gráfica se pueden ver los procesos tanto de confusión como de difusión aplicados a ciertas imágenes seleccionadas con diferentes tamaños (nxn píxeles para que sean cuadradas), se muestra la imagen a utilizar y las salidas que tienen en cada paso del sistema de cifrado y descifrado.

La interfaz será muy simple y siguiendo una serie de pasos que a continuación se enlistan, podremos utilizarla y veremos en pantalla el cifrado o descifrado según sea el caso:

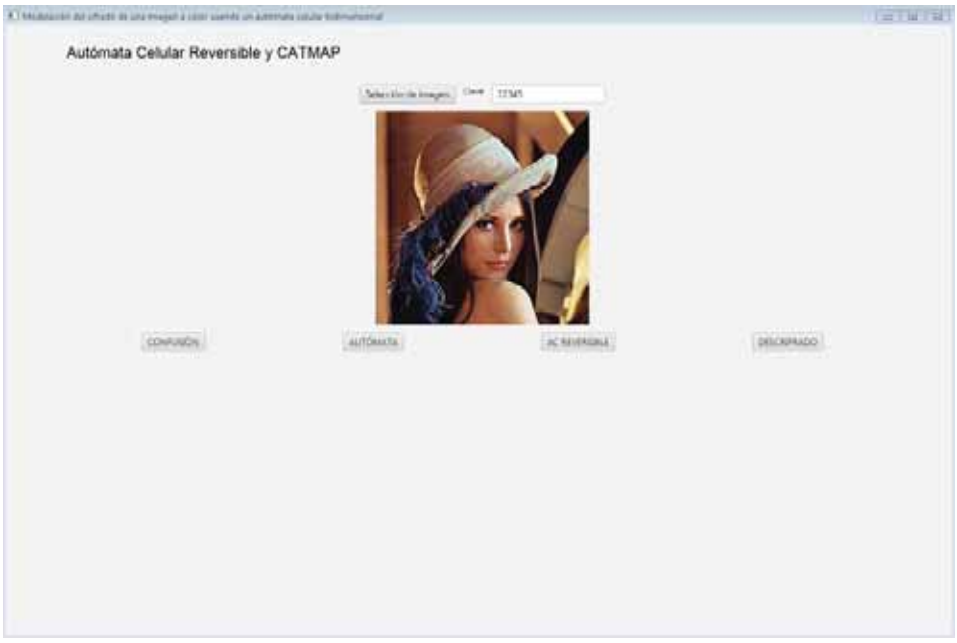
1. Al ejecutar el programa, se abre la interfaz y las opciones a elegir.



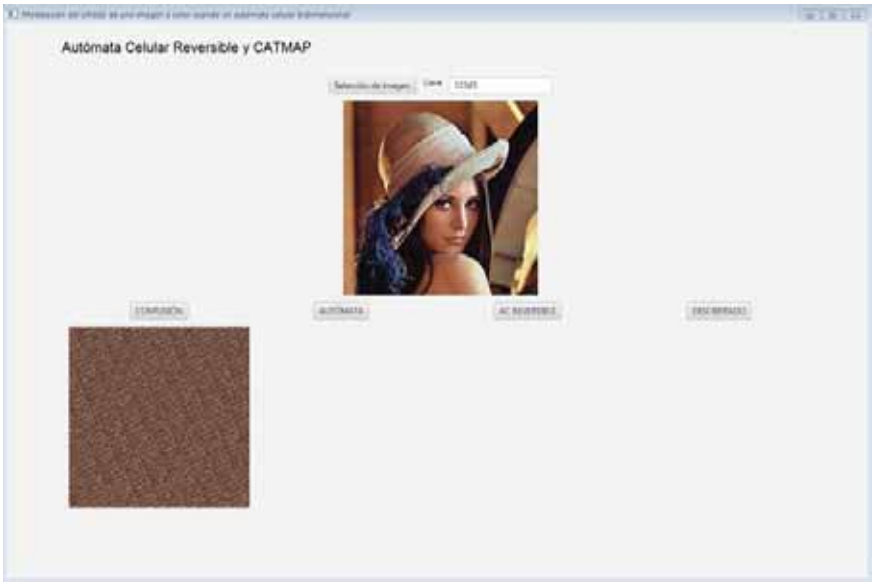
2. La primera parte comienza con el cifrado de la imagen así que, damos click en el botón selección de imagen y la buscamos con el explorador de archivos.



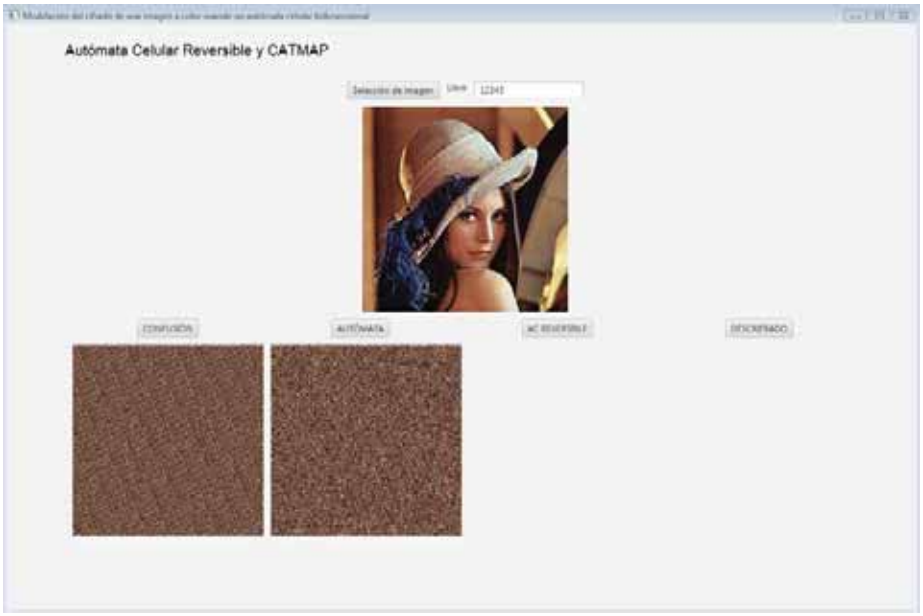
3. Ya seleccionada la imagen, introducimos una llave que tiene que ser mayor o igual a 5 y menor o igual a 15, los caracteres a elegir deben estar entre el conjunto: $\{1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$ pueden repetirse y alternarse. Por default la clave siempre será 12345.



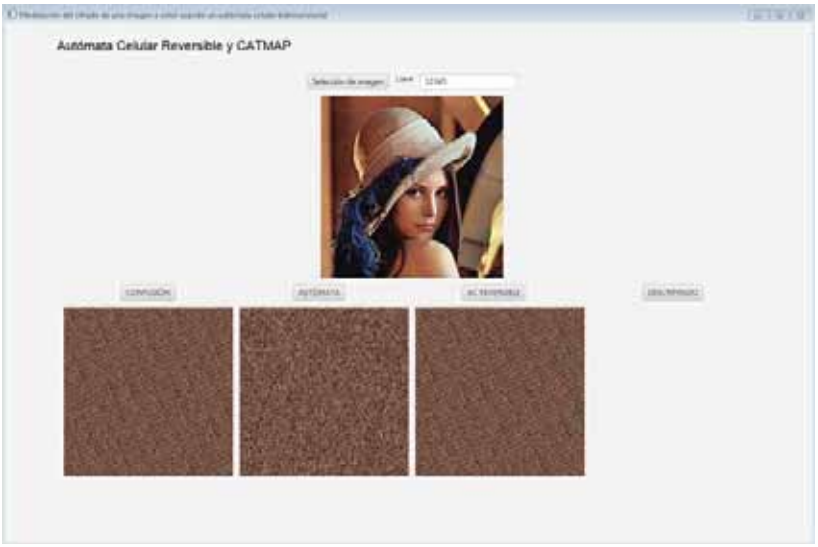
4. Ya que tenemos la imagen y la llave, procedemos a dar click en el botón "CONFUSIÓN" que mediante el algoritmo Cat map, se encargará de la primera parte del cifrado.



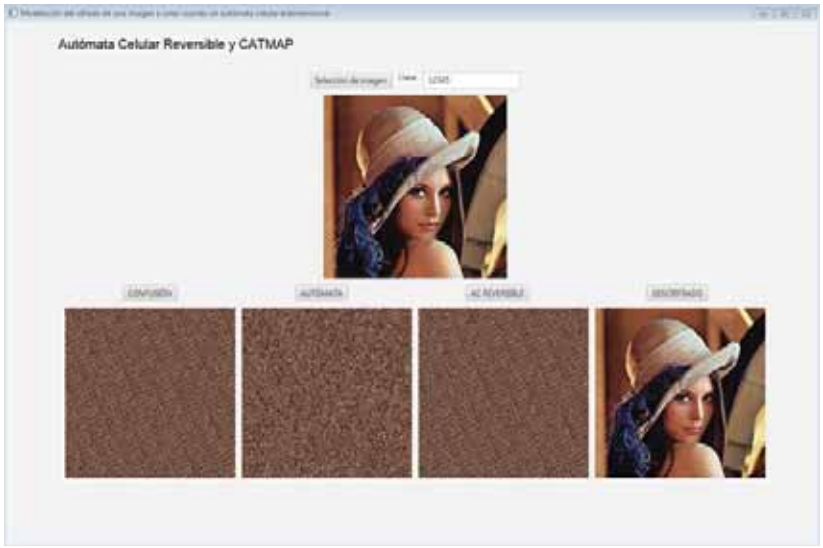
5. Una vez realizada la confusión, procedemos a utilizar el autómata, entonces daremos click al botón "AUTÓMATA" y veremos el resultado en la pantalla, la imagen mostrada será la imagen cifrada y esto cierra el ciclo de cifrado de la imagen.



6. Utilizando la misma interfaz mostraremos el proceso de cifrado continuando con lo ya obtenido en la imagen anterior, para comprobar la reversibilidad del autómata, daremos click en el botón “AC REVERSIBLE” y sí la llave es la misma, nos mostrará la imagen ya obtenida en “CONFUSIÓN”.



7. Una vez obtenida la imagen podremos darnos cuenta que la imagen “CONFUSIÓN” Y “AC REVERSIBLE” son las mismas, ahora sólo resta dar click en el botón “DESCIFRADO” que mostrará la imagen original siempre y cuando la llave sea la misma.

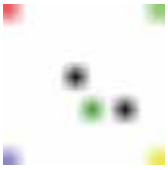
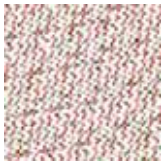
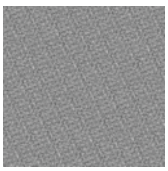


8. Se comprueba que la interfaz ha recuperado la imagen original después de cifrarla en la primera parte.

8. Análisis y discusión de resultados

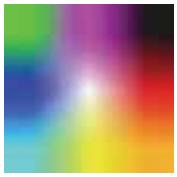
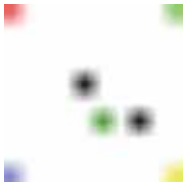
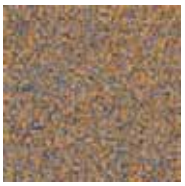
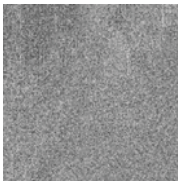
A continuación se muestra el resultado de la simulación de algunas imágenes que fueron procesadas por el criptosistema.

CATMAP

Imagen original	Tamaño *nxn* Píxeles	Iteraciones para llegar a la original	Iteraciones a realizar para el ejemplo	Imagen procesada
	3x3	4	2	
	10x10	30	10	
	74x74	114	14	
	100x100	150	100	
	124x124	15	7	
	300x300	300	80	
	500x500	750	400	

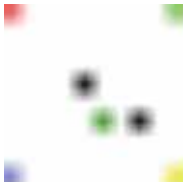
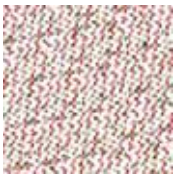
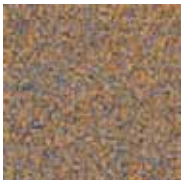
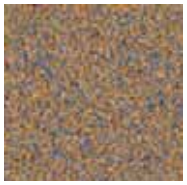
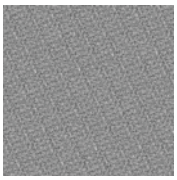
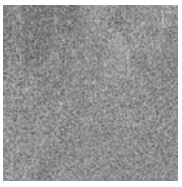
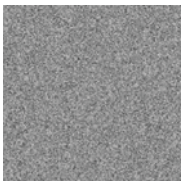
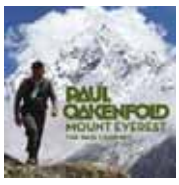
Simulación 1. CAT MAP Difusión de píxeles en algunas imágenes de diferentes tamaños.

Autómata celular sin CATMAP para diferente imágenes con diferentes llaves

Imagen original	Tamaño	Llave	Imagen procesada
	3x3	12345	
	10x10	11111	
	74x74	987654321	
	100x100	1A2B3C4D	
	124x124	98FE76DC54	
	300x300	5F4F3F2F1F	
	500x500	12321	
	900x900	FFFFFF1FFFFFF	

Simulación 2. Difusión de píxeles usando un autómata celular. Diferentes tamaños de imagen con diferentes llaves.

Confusión (CATMAP) y Difusión (Autómata celular) Aplicados a la misma imagen para completar el ciclo de cifrado

Imagen original	CATMAP	AUTÓMATA	LLAVE	Imagen obtenida
			12345	
			11111	
			987654321	
			1A2B3C4D	
			98FE76DC54	
			5F4F3F2F1F	
			12321	
			FFFFFF1FFFFFF	

Simulación 3. Proceso de cifrado completo (CATMAP + AUTÓMATA CELULAR).

Comportamiento de una imagen procesada por sólo el autómata (Sin CATMAP) con diferentes llaves para mostrar cómo el resultado no es siempre el mismo.

Imagen Original 500x500	Llave 11111	Llave 789AB	Llave 1A2B3C4D5E6F789
			

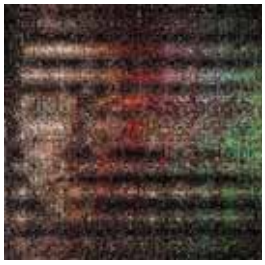
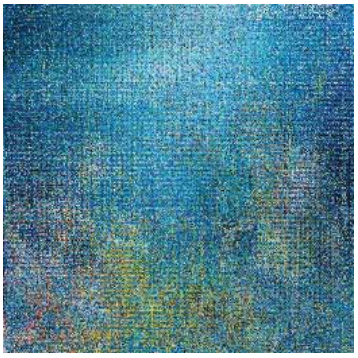
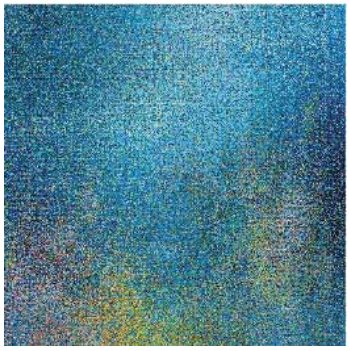
Imagen Original 500x500	Llave 654321	Llave 11BB22EE	Llave FEDCBA654321FA5
			

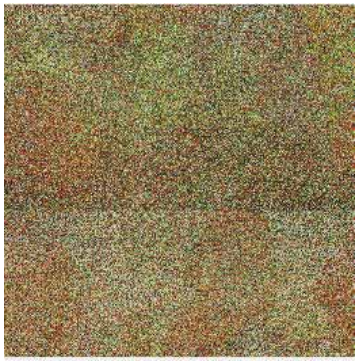
Imagen Original 500x500	Llave 123456789A	Llave CABACEFADE	Llave CA9BA8CE7FAD6E5
			

Simulación 4. Cómo actúan diferentes llaves sobre una misma imagen en el autómata celular.

Comportamiento del Autómata Celular Reversible (ACR) cuando no se utiliza la misma llave que previamente se utiliza en el autómata (AC).

Imagen original 900x900	Autómata Llave: ABC123	Autómata reversible Llave: ABC12
		

Simulación 5. Introduciendo una llave diferente en el autómata celular reversible (ACR)

Imagen original 1111x1111	Cifrado (CATMAP+AC) Llave: 44444	Descifrado (ACR + CATMAP) Llave: AAAAA
		

Simulación 6. Introduciendo una llave diferente entre el cifrado y descifrado.

8.1 Análisis de simulaciones

En la *simulación 1* podemos observar que el tamaño de la imagen no siempre tiene que ver con el número de iteraciones que debe hacer el módulo de CATMAP para llegar nuevamente a la imagen original, por ejemplo, la imagen de 74x74 píxeles completa su ciclo en 114 iteraciones, mientras que la imagen de 124x124 píxeles lo completa en 15. Para este proyecto se han elegido imágenes de las que a base de pruebas se tiene el número de iteraciones necesarias para completar el ciclo del Cat map, una vez que se tiene el número

de iteraciones se divide a la mitad y ése dato será el número de iteraciones que ocupará el modulo Cat map para el proceso de cifrado.

Tamaño de imagen nxn	Número de iteraciones que completan el ciclo
3x3	4
10x10	30
30x30	60
74x74	114
100x100	150
101x101	25
124x124	15
150x150	300
183x183	60
225x225	300
300x300	300
500x500	750
900x900	300
1111x1111	25
1499x1499	749

Tabla 3. Número de iteraciones que necesita una imagen procesada por el módulo CATMAP para completar un ciclo.

En la *simulación 2* podemos ver algunos ejemplos de cómo funciona el autómata celular sobre algunas imágenes, se pusieron diferentes llaves con diferente longitudes y distribución de caracteres. Nótese que si la llave es muy repetitiva en cuanto a caracteres y pequeña en longitud se puede intuir que de qué imagen pudiera tratarse. En las *simulación 4* se utilizaron imágenes con mayor calidad para mostrar que para fortalecer el cifrado en la etapa del autómata, se necesita incluir una llave de mayor longitud y de mayor diversidad en cuanto a caracteres se refiere.

Para el cifrado de la imagen se decidió ejemplificar en la *simulación 3,* cómo cada módulo hace un "cifrado parcial" para después combinarlos y hacer más fuerte el proceso de encriptado.

El peor de los casos en el criptosistema es cuando no se introduce la llave correcta y por eso en la *simulación 5* se tomaron 2 imágenes como ejemplo, en la primer imagen se aprecia cómo sólo le faltó un digito a la llave para que fuera la correcta y el resultado que produce el descifrado es muy cercano a mostrar la imagen original; en la fase completa del descifrado (CATMAP y ACR) y cuando la llave tiene una diferente longitud y diferentes caracteres el resultado que produce es completamente diferente y no es para nada similar a la imagen original, lo que nos vuelve a confirmar que se deben utilizar llaves de mayor longitud y con mayor número de caracteres para fortalecer el criptosistema.

Como resultado tenemos que, en una imagen digital a color un conjunto de píxeles dispuestos según una distribución particular sobre una matriz bidimensional representan una imagen, la correlación entre los píxeles vecinos es uno de los indicadores fundamentales de la información contenida en la imagen. Por lo que un método de encriptamiento robusto aplicado a imágenes digitales debe construir una imagen cifrada en la cual se minimiza la correlación entre los píxeles vecinos, ocultando la imagen original.

9. Conclusiones

En este proyecto se propuso y presentó un protocolo simétrico de cifrado para imágenes digitales que consta de dos etapas: La etapa de confusión y la etapa de difusión. La etapa de confusión se da por medio del mapeo caótico del gato, mientras que en la etapa de la difusión algunas evoluciones de un autómata celular con memoria reversible son procesadas usando como configuraciones iniciales matrices que son extraídas de la imagen original a ser cifrada.

- La notación utilizada y los mecanismos creados para describir la dinámica, evidencian que la teoría de autómatas celulares ofrece un marco de referencia apropiado para el diseño de algoritmos de cifrado.
- La descripción formal del algoritmo de cifrado facilita el proceso de codificación en un lenguaje de programación.
- Los resultados obtenidos demuestran que la dinámica propuesta produce un cifrado de alta calidad.
- La incorporación del parámetro *DI* (Distancia celular) propicia un proceso de difusión de naturaleza no lineal que favorece la calidad de la dinámica.
- Si el uso de autómatas celulares reversibles es un buen método de cifrado por sí solo, complementarlo con el mapa caótico del gato lo hace aún más efectivo.

Para futuros trabajos dirigidos a desarrollar protocolos de cifrado similares que involucren autómatas celulares en su totalidad, una fase de confusión que use autómatas y no mapeos caóticos sería muy interesante así como también el estudio y la clasificación de los autómatas celulares más adecuados para su uso en ambas etapas.

10. Bibliografía

[1] Álvarez Marañón, G., Hernández Encinas, L., Martín del Rey, "A new secret sharing scheme for images base don additive 2-dimensional celular autómata", LNCS, vol. 3522, pp. 411-418, 2005

[2] Chang, C., Hwang, M. and Chen, T. "A new encryption algorithm for image cryptosystems". Journal of Systems and Software, 58(2), pp.83-91. (2001)

[3] Fredkin, E., "Digital Mechanics. An informal process based on reversible universal cellular autómata, Physica D 45, 254-270, (1990).

[4] Fridrich, J., "Symmetric ciphers based on two-dimensional chaotic maps", Int. J. Bifucar. Chaos 8, 1259-1284 (1998).

[5] Gao, H., Zhang, Y., "A new chaotic algorithm for image encryption", Chaos Soliton Frac. 29, 393-399, (2006).

[6] Toffoli, T., Margolus, N., "Cellular Automata Machines: A new Environment for Modeling", The MIT Press, Cambridge (1987).

[7] Toffoli, T., Margolus, N., "Invertible cellular automata: A review", Physica D 45, 229-253, (1990).

[8] J. D. Villegas Hernández, "Transmisión de mensajes en archivos de imágenes y texto usando esteganografía", proyecto terminal, División de Ciencias Básicas e Ingenierías, Universidad Autónoma Metropolitana Unidad Azcapotzalco, México, 2009.

[9] M. E. Valencia Arana, "Esteganografía de archivos usando redundancia cíclica en imágenes JPEG", proyecto terminal, División de Ciencias Básicas e Ingenierías, Universidad Autónoma Metropolitana Unidad Azcapotzalco, México, 2011.

[10] A. Álvarez Gaona, "Implementación software-hardware de aritmética sobre campos finitos binarios GF(2m) en curvas elípticas para aplicaciones criptográficas de llave pública", proyecto terminal, División de Ciencias Básicas e Ingenierías, Universidad Autónoma Metropolitana Unidad Azcapotzalco, México, 2012.

[11] Atieh Bakhshandeh, Ziba Eslami, "An authenticated image encryption scheme based on chaotic maps and memory cellular automata", Optics and Lasers in Engineering, vol. 51, pp. 665-673, 2013.