Cifrado de imágenes y Matemáticas

Ángela Rojas Matas [1], Alberto Cano Rojas[2]

[1] Departamento de Matemáticas, Universidad de Córdoba, España
[2] Alumno de Ingeniería Informática, Universidad de Córdoba, España

Resumen

Un tema que debe interesar al profesorado de Matemáticas de todos los niveles educativos es cómo hacer comprender a nuestros alumnos la utilidad de los conceptos matemáticos que están estudiando en nuestras asignaturas. Si se les presenta aplicaciones de las Matemáticas en temas atractivos e interesantes conseguiremos motivar a nuestros alumnos. Una de estas aplicaciones es el cifrado de la información.

La Criptografía es un tema de gran actualidad debido al auge de Internet, la telefonía móvil, etc. y el correspondiente aumento de intercambio de información: comercio electrónico, transacciones bancarias, etc. Es imprescindible poder intercambiar información de manera segura a salvo de intrusos malintencionados.

No sólo se producen intercambios de mensajes de texto sino que también se producen intercambios de otro tipo de ficheros digitales como imágenes, ficheros de audio, etc.

Este trabajo se va a dedicar a presentar una experiencia docente realizada con alumnos de una asignatura de Matemáticas de primer curso de Ingeniería Técnica Informática sobre técnicas de cifrado de imágenes digitales. Los métodos de cifrado han sido muy variados: un par de métodos basados en cálculo matricial, un método basado en el Teorema Chino de los Restos y un método basado en secuencias caóticas.

De esta forma conseguimos relacionar las Matemáticas que se estudian en clase con temas de interés para nuestros alumnos. Así conseguimos que valoren más los conocimientos que están adquiriendo.

Palabras clave: Criptografía, Imágenes, Matemáticas.

1. Introducción

No es raro escuchar por los pasillos a los alumnos decir la temida frase: "¿y esto para qué me sirve?". Nosotros pensamos que los profesores de Matemáticas debemos dar respuesta a esta pregunta, presentando aplicaciones interesantes de los contenidos matemáticos que estamos trabajando en clase. De esta forma conseguiremos que nuestros alumnos aprendan Matemáticas sabiendo para qué sirven y cómo están muy relacionadas con temas de indudable interés para su titulación.

Con esta forma de pensar, hemos trabajo en clase con muchos y variados tópicos como la criptografía, la esteganografía digital, la compresión de imágenes digitales, códigos detectores y correctores de errores, etc. Este trabajo se va a dedicar a exponer detenidamente uno de estos temas: el cifrado de imágenes digitales.

Es indudable que este tema es útil para un futuro informático. Las empresas reclaman métodos seguros para el intercambio de información: el comercio electrónico va en aumento, las transacciones bancarias por Internet también., etc. Esto ha provocado que la Criptografía se haya convertido en un tema de interés para investigadores en todo el mundo. También ha provocado que la Criptografía, que antes estaba restringida a usos militares, se haya extendido a otros ámbitos: protección del correo electrónico, protección de patentes industriales, etc.

Por otro lado, la criptografía atrae poderosamente la atención de nuestros alumnos. Podemos despertar el interés por el cifrado y descifrado poniendo a prueba las dotes de espías de nuestros alumnos. En nuestro caso, nos vamos a centrar en cómo cifrar y descifrar imágenes digitales.

Existen ocasiones donde puede ser necesario mantener en secreto una imagen digital. Por ejemplo:

- Es habitual el intercambio de fotografías por telefonía móvil o Internet y en ocasiones, deseamos privacidad para ello.
- En Medicina es deseable disponer de técnicas que permitan cifrar las imágenes de pruebas radiológicas de los pacientes para proteger la privacidad del paciente.
- Los militares o servicios secretos pueden desear mantener en secreto posiciones

estratégicas dentro de un mapa o una fotografía de una región secreta.

- Etc.

Todas estas cuestiones justifican el interés por las técnicas de cifrado de imágenes digitales.

En este trabajo se presentan algunas técnicas muy sencillas de cifrado de imágenes digitales llevadas a cabo por alumnos de una asignatura de Matemáticas de primer curso de Ingeniería Técnica Informática. En sesiones prácticas, se les proporcionaba imágenes cifradas y ellos, trabajando por grupos, debían descifrar dichas imágenes conociendo las claves para hacerlo. Con esto conseguimos además que nuestros alumnos trabajen con: aritmética modular, matrices, funciones caóticas, etc. Se empleaba el software Mathematica para hacerlo.

La organización del artículo es la siguiente: la sección 2 se dedica a una técnica de cifrado matricial llamado cifrado Hill, la sección 3 se dedica a una técnica también matricial pero distinta de la anterior que hemos llamado cifrado del mapa de Arnold, la sección 4 se dedica a una técnica de cifrado basada en el Teorema Chino de los Restos, la sección 5 está dedicada a una técnica de cifrado basado en secuencias caóticas y, por último, presentamos las conclusiones.

2. Cifrado Hill

Lester Hill propuso en 1929 su método de cifrado de un mensaje en la revista *The American Mathematical Monthly* [1]. La idea es bastante sencilla, como vamos a exponer a continuación directamente adaptada al caso de imágenes digitales y con un ejemplo concreto que siempre se entiende mejor.

Una imagen digital no es más que una matriz de números. Por ejemplo, la imagen de la figura 1 es una matriz de tamaño 256×256 donde los niveles de gris de la imagen varían desde 0 correspondiente al negro hasta 255 correspondiente al blanco. Los números comprendidos entre 0 y 255 se escriben en binario con 8 bits, por eso esta imagen necesita 1 byte por píxel.

Usaremos una matriz secreta K sólo conocida por emisor y receptor, por ejemplo de tamaño 2×2, como la siguiente:

$$K=\begin{pmatrix}21 & 35\\ 18 & 79\end{pmatrix}$$

Iremos cogiendo los niveles de gris de los píxeles también de dos en dos, empezando en la esquina superior izquierda de la matriz y moviéndonos de izquierda a derecha y de arriba a abajo: el primer bloque será a_{11}, a_{12}, el segundo bloque será a_{13}, a_{14}, y así sucesivamente. Supongamos que los dos primeros niveles de gris son: 125 y 137. El cifrado se obtiene de la siguiente forma:

$$\begin{pmatrix}21 & 35\\ 18 & 79\end{pmatrix}\begin{pmatrix}125\\ 137\end{pmatrix}=\begin{pmatrix}7420\\ 13073\end{pmatrix}=\begin{pmatrix}252\\ 17\end{pmatrix}(\mathrm{mod}\,256)$$

Es necesario hacer la congruencia módulo 256 para obtener siempre un nivel de gris válido, es decir, un número entre 0 y 255. De esta forma, los dos niveles de gris originales que eran 125 y 137 se transformarán en 252 y 17 respectivamente.

En la figura 1 podemos ver la imagen original y en la figura 2 la imagen cifrada usando la clave K anterior.

Figura 1: Imagen original

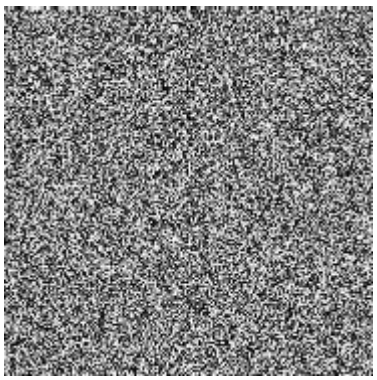

Figura 2. Imagen cifrada con el método de Hill

Hay que hacer una observación importante: no vale cualquier matriz clave K. Por ejemplo, si se usa la matriz:

$$K=\begin{pmatrix}20 & 8\\ 15 & 7\end{pmatrix}$$

el emisor podrá cifrar la imagen, pero el receptor no podrá descifrar, por lo tanto, no sirve para nada. Veamos por qué. Sabemos que:

Si $K=\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ y $|K| = a\,d - b\,c \neq 0 \Rightarrow$

$$K^{-1} = \frac{1}{a\,d-b\,c}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Como estamos trabajando módulo 256 es necesario además que $|K|$ sea un número inversible módulo 256. Para que eso ocurra $|K|$ debe ser primo relativo con 256, es decir: $\text{mcd}(|K|,\ 256) = 1$

Por esta razón, $K=\begin{pmatrix} 20 & 8 \\ 15 & 7 \end{pmatrix}$ no es una matriz de cifrado válida ya que: $|K| = 20$ y este número no es primo relativo con 256.

Sin embargo $K=\begin{pmatrix} 21 & 35 \\ 18 & 79 \end{pmatrix}$ resulta que:

$$|K| = 1029 = 5 \pmod{256} \Rightarrow \text{mcd}(5,256) = 1$$

Eso quiere decir que 5 tiene inverso módulo 256, es decir, existe un número que multiplicado por 5 da 1, trabajando módulo 256. Este número resulta ser 205, de modo que:

$$K^{-1} = 205\begin{pmatrix} 79 & -35 \\ -18 & 21 \end{pmatrix} = \begin{pmatrix} 67 & 249 \\ 150 & 209 \end{pmatrix} \pmod{256}$$

El receptor usará la matriz de descifrado anterior y podrá recuperar la imagen original.

Como se puede comprobar, por ejemplo en [2], existe aún interés en este tipo de cifrado.

3. Cifrado del mapa de Arnold

Ahora el enfoque es distinto. De nuevo, por simplicidad, nos vamos a un ejemplo. Supongamos que tenemos una imagen de tamaño 124×124 como la imagen de la figura 3 y que (x, y) son las coordenadas de un píxel.

Figura 3: Imagen original

Estas coordenadas (x, y) con $x, y = 0, 1, \ldots, 123$ (x es la fila, y es la columna) se transformarán en otras (x', y') de la siguiente forma:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = K\begin{pmatrix} x \\ y \end{pmatrix} \pmod{124} \text{ con } K=\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$$

Entonces el píxel que ocupa la posición (x, y) pasa a la posición (x', y'). En esta ocasión no se cambia el nivel de gris del píxel sino que el píxel se lleva a otro lugar con su nivel de gris si la imagen es en escala de grises, o su color si la imagen es una imagen en color como en este caso.

La operación anterior se realiza para todos los píxeles de la imagen y esto nos da lugar a una permutación de los píxeles que componen la imagen. Repetimos el proceso un número determinado de veces ¿qué ocurrirá?. En la figura 4 se muestra el resultado

Resulta sorprendente que tras 15 iteraciones todo vuelva a su lugar. Lo que acabamos de hacer se conoce como "mapa del gato de Arnold" por el matemático ruso Vladimir Arnold (1937-) y porque se hizo este proceso iterativo con la imagen de un gato.

La matriz K de nuevo es secreta y la única condición además es que tenga un determinante primo relativo con el módulo, 124 en este caso.

La imagen cifrada a enviar a nuestro receptor podría ser la imagen con aspecto aleatorio obtenida después de 5 iteraciones como verse en la figura 4. La clave secreta que el receptor debe conocer para poder descifrar dicha imagen es la matriz K empleada en el cifrado y el número de iteraciones.

El número de iteraciones a realizar para que todo vuelva a su lugar depende de N (el tamaño de la imagen original es $N\times N$) y de la matriz K utilizada. Generalmente, si N crece, el número de iteraciones suele ser más elevado pero eso no siempre es así. Por ejemplo, si seguimos con la misma matriz K, puede comprobarse fácilmente que para $N=101$ el número de iteraciones es 25, si $N=124$ son 15 y si $N=150$ son 300. En [3] se pueden ver los periodos de otras matrices distintas a las empleadas en este ejemplo.

Imagen original | Iteración 1

Iteración 2 | Iteración 5

Iteración 14 | Iteración 15

Figura 4: Resultado de distintas iteraciones

4. Cifrado con el Teorema Chino de los Restos

En esta sección vamos a implementar un algoritmo de cifrado de imágenes digitales descrito en un trabajo fin de master de 2007 cuya referencia de Internet puede verse en [1].

Se basa en el Teorema Chino de los Restos, una herramienta de Teoría de Números con mucha utilidad en la criptografía actual.

Supongamos que deseamos resolver el problema de hallar x tal que:

$$\left.\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_k \pmod{m_k} \end{aligned}\right\}$$

siendo los módulos m_i primos relativos entre sí.

Supongamos que llamamos $M = m_1\, m_2 \cdots m_k$

$$M_1 = \frac{M}{m_1},\ M_2 = \frac{M}{m_2},\ ...,\ M_k = \frac{M}{m_k}$$

Entonces, M_i y m_i serán primos relativos también, y usando el algoritmo extendido de Euclides seremos capaces de averiguar un par de números enteros x_i e y_i tales que: $x_i\, M_i + y_i\, m_i = 1$

Entonces, el Teorema Chino de los Restos nos dice que la solución del problema anterior viene dada por:

$$x \equiv a_1\, x_1\, M_1 + \cdots + a_k\, x_k\, M_k \pmod{M}$$

Por otro lado, supongamos que tenemos una imagen en escala de grises, variando entre 0 y 255, como la mostrada en la figura 5.

Figura 5: Imagen original

Los pasos a seguir para su cifrado son los siguientes:

1) Se escogerán un conjunto de k módulos $\{m_1, m_2, \cdots, m_k\}$ de modo que sean todos números mayores o iguales que 256 y primos relativos entre sí. Serán secretos y sólo conocidos por emisor y receptor. Sea $M = m_1\, m_2 \ldots m_k$.

2) Dividimos la imagen en bloques de tamaño k. Cogemos un bloque de k niveles de gris de la imagen original que indicaremos por: $\{a_1, a_2, \ldots, a_k\}$ y, aplicando el Teorema Chino de los Restos, resolveremos el siguiente sistema de ecuaciones en congruencia:

$$\left.\begin{aligned} x &\equiv a_1 (\bmod m_1) \\ x &\equiv a_2 (\bmod m_2) \\ &\cdots \\ x &\equiv a_k (\bmod m_k) \end{aligned}\right\}$$

Como sabemos el valor de x que proporciona el Teorema Chino de los Restos es:

$$x = a_1\, x_1\, M_1 + \cdots + a_k\, x_k\, M_k \pmod{M}$$

Pues bien, enviaremos al receptor de la imagen cifrada los valores de x anteriormente obtenidos (uno por cada bloque). El receptor, conocedor del conjunto de

módulos utilizados, recuperará los niveles de gris originales calculando:

$$\left.\begin{array}{l} x \pmod{m_1} = a_1 \\ x \pmod{m_2} = a_2 \\ \cdots \\ x \pmod{m_k} = a_k \end{array}\right\}$$

pudiendo así recuperar los niveles de gris originales: $\{a_1, a_2, \ldots, a_k\}$.

Vamos a usar bloques de tamaño 4, por ejemplo, que vamos a ir tomando empezando en la esquina superior izquierda y moviéndonos de izquierda a derecha y de arriba a abajo. Así si *A* es la imagen original, el primer bloque es: $\{a_{11}, a_{12}, a_{13}, a_{14}\}$ y así sucesivamente.

Los cuatro módulos escogidos (sólo conocidos por emisor y receptor) podrían ser, por ejemplo:

$$\{m_1, m_2, m_3, m_4\} = \{256,\ 257,\ 259,\ 261\}$$

Los módulos escogidos son números mayores o iguales que 256 y primos relativos.

Entonces:

$$M = m_1\, m_2\, m_3\, m_4 = 4447473408$$

Usando Mathematica se puede comprobar que se obtienen los siguientes valores:

$$M_1 = 17372943 \Rightarrow \text{mcd}\,(M_1, m_1) = 1 \Rightarrow$$
$$M_1 x_1 + m_1 y_1 = 1 \Rightarrow \quad x_1 = -17$$

Análogamente:

$M_2 = 17305344 \qquad x_2 = 32$

$M_3 = 17171712 \qquad x_3 = 108$

$M_4 = 17040128 \qquad x_4 = -124$

Usaremos el algoritmo Extendido de Euclides para el cálculo de los x_i. Estos números se tendrán que calcular solamente una vez.

Supongamos que el bloque de 4 píxeles que nos toca tomar es: $\{152, 153, 152, 155\}$

La solución del sistema de congruencias:

$$\left.\begin{array}{l} x \equiv 152 \pmod{256} \\ x \equiv 153 \pmod{257} \\ x \equiv 152 \pmod{259} \\ x \equiv 155 \pmod{261} \end{array}\right\}$$

se obtendrá aplicando el Teorema Chino de los Restos:

$$x = a_1\, x_1\, M_1 + \cdots + a_k\, x_k\, M_k \pmod{M}$$
$$= (152)\,(17372943)\,(-17) + (153)\,(17305344)(32) +$$
$$+ (152)\,(17171712)\,(108) + (155)\,(17040128)\,(-124)$$
$$= 3109790360 \pmod{M}$$

Como hemos dicho son estos números x así construidos los que enviaremos al receptor de la imagen.

En este caso, el receptor cuando reciba el número $x = 3109790360$ hará los siguientes cálculos:

$$3109790360 \pmod{256} = 152$$
$$3109790360 \pmod{257} = 153$$
$$3109790360 \pmod{259} = 152$$
$$3109790360 \pmod{261} = 155$$

pudiendo recuperar exactamente los 4 niveles de gris de este bloque.

Vamos a dar a continuación detalles de cómo se van a enviar los números x anteriormente mencionados (un x por cada bloque de 4 píxeles).

Los posibles valores de x variarán entre 0 y $M-1$, que son los posibles restos módulo M. Puede comprobarse que $M-1$ tiene exactamente 33 bits en nuestro ejemplo.

Por otro lado, como la imagen es de tamaño $256 \times 256 \Rightarrow$ nº de bloques $= \dfrac{256 \times 256}{4} = 16384$

Hay que enviar 16384 números y cada uno de ellos ocupa 33 bits, por lo tanto, el número de bits a enviar será: $16384 \times 33 = 540672$.

Obtenemos dicha lista de 540672 bits y después los agruparemos de 8 en 8, obteniendo una nueva lista de elementos.

Cada grupo de 8 bits se pasa a decimal, obteniendo un número comprendido entre 0 y 255 que se puede interpretar como un nivel de gris. Por lo tanto conseguiremos una lista de 67584 elementos donde cada elemento es un número entre 0 y 255.

La imagen original era de tamaño $256 \times 256 = 65536$ y este tamaño es insuficiente para la lista anterior con 67584 elementos. Así que vamos a construir una matriz B de tamaño algo más grande, concretamente $264 \times 256 = 67584$ para tener espacio suficiente. Iremos rellenando los elementos de *B* empezando en la esquina superior izquierda y moviéndonos de izquierda a derecha y de arriba a abajo.

La matriz resultado *B* se puede mostrar como una imagen y será la imagen codificada que enviaremos al receptor. La imagen cifrada se muestra en la figura 6.

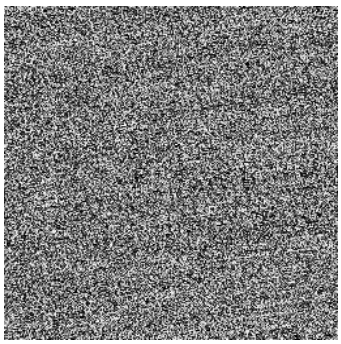

Figura 6: Resultado del cifrado usando el Teorema Chino de los Restos

5. Cifrado con funciones caóticas

Se pueden encontrar muchos artículos que proponen el uso de funciones caóticas para cifrar imágenes digitales como [4] y [5]. Veamos un ejemplo que usa *la función logística*. Esta función se define de la forma:

$$\left.\begin{array}{rcl} f:[0,1] & \to & [0,1] \\ x & \to & r\,x(1-x) \end{array}\right\}$$

Para que la imagen de esta aplicación quede confinada al intervalo $[0,1]$ debe ocurrir que $r \in [0,4]$.

Supongamos que partiendo de un valor fijo x_0 generamos una secuencia del tipo:

$$x_1 = f(x_0)$$
$$x_2 = f(x_1)$$
$$\vdots$$

De esta forma obtenemos una sucesión $\{x_n\}$. Pues bien puede observarse de forma experimental un comportamiento completamente distinto de esta sucesión según el valor de r empleado aunque se parta siempre del mismo punto inicial x_0.

Por ejemplo, partimos de $x_0 = 0.1$ y hacemos en todos los casos 100 iteraciones. Mostramos a continuación los últimos términos obtenidos para varios valores de r.

$r = 2.7 \Rightarrow \{\ldots,\ 0.6296,\ 0.6296,\ 0.6296,$
$0.6296,\ 0.6296,\ 0.6296,\ 0.6296,\ 0.6296\}$

$r = 3.3 \Rightarrow \{\ldots,\ 0.4794,\ 0.8236, 0.4794,$
$0.8236, 0.4794,\ 0.8236, 0.4794,\ 0.8236, 0.4794\}$

En el primer caso la sucesión converge al punto 0.6296. En el segundo caso se obtiene un ciclo de periodo 2. Sin embargo, para $r = 4$ y partiendo también de $x_0 = 0.1$ no existe ninguna conexión entre unos términos y otros de la sucesión por mucho que iteremos. De hecho, para valores de $r > 3.5699456$, la sucesión obtenida a partir de cualquier $z_0 \in (0,\ 1)$ será *caótica* ([6]).

Vamos a mostrar a continuación un fractal muy famoso conocido con el nombre de *fractal de Feigenbaum* (EEUU, 1944-) o *diagrama de bifurcación*. En el eje de abscisas representamos distintos valores de r (entre 2.4 y 4 en nuestro caso) y haremos lo siguiente: partiremos de $x_0 = 0.1$, efectuaremos 1000 iteraciones y representaremos los 100 últimos términos.

De manera que, por ejemplo, para $r = 2.7$ sólo dibujaremos un punto que será (2.7, 0.62926). Para $r = 3.3$ sólo dibujaremos dos puntos (3.3, 0.4794) y (3.3, 0.8236) y para $r = 4$ dibujaremos muchos puntos. Pues bien, la figura que se obtiene se representa en la figura 6. Se trata de un gráfico donde observamos una primera región donde las sucesiones convergen a un solo punto, luego le sigue una región donde las sucesiones convergen a dos puntos (periodo 2), luego le sigue una región donde las sucesiones convergen a cuatro puntos (periodo 4), luego periodo 8, etc. hasta que después de esta zona, donde se van produciendo duplicaciones de periodo, viene el caos.

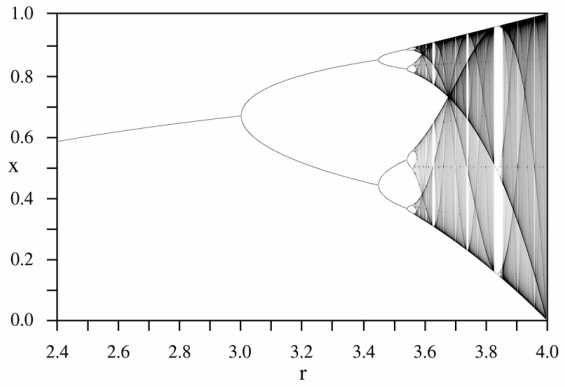

Figura 7: Fractal de Feigebaum

Una propiedad muy importante de las funciones caóticas es *la sensibilidad a las condiciones iniciales* que a continuación explicamos con un ejemplo.

Para un valor de r no caótico como $r = 2.7$, podemos comprobar cómo, sea cual sea el valor inicial de x_0, siempre se obtiene convergencia a 0.6296.

$x_0 = 0.1 \Rightarrow \{\ldots,\ 0.6296,\ 0.6296,\ 0.6296,$
$0.6296,\ 0.6296,\ 0.6296,\ 0.6296,\ 0.6296\}$

$x_0 = 0.101 \Rightarrow \{\ldots,\ 0.6296,\ 0.6296,\ 0.6296,$
$0.6296,\ 0.6296,\ 0.6296,\ 0.6296,\ 0.6296\}$

$x_0 = 0.9 \Rightarrow \{\ldots,\ 0.6296,\ 0.6296,\ 0.6296,$
$0.6296,\ 0.6296,\ 0.6296,\ 0.6296,\ 0.6296\}$

Es decir, sea cual sea el valor x_0 de inicio de la iteración, la sucesión $\{x_n\}$ es convergente al valor 0.6296.

Sin embargo, si $r = 4$ se obtienen sucesiones totalmente distintas aunque los valores iniciales x_0 sean muy próximos entre sí. Por ejemplo:

$$x_0 = 0.1 \Rightarrow \{\ldots, 0.4795,\ 0.9983,\ 0.0067,\ 0.0267,\ 0.1039,\ 0.3724\}$$

$$x_0 = 0.101 \Rightarrow \{\ldots, 0.8371,\ 0.5455,\ 0.9917,\ 0.0329,\ 0.1271,\ 0.4439\}$$

Los resultados anteriores dependen de la precisión utilizada para hacer los cálculos y, por lo tanto, pueden variar según el software que utilicemos, en nuestro caso realizados en Mathematica.

Así se pueden obtener resultados totalmente dispares a largo plazo aún partiendo de valores iniciales muy próximos. El efecto anterior se conoce como "*efecto mariposa*" o *sensibilidad a la condiciones iniciales* y ya fue observada de forma accidental por Edward Lorentz en 1968. Este meteorólogo seguía un proceso iterativo para la predicción del tiempo que le llevaba varias horas de trabajo en un ordenador. Un día repitió los cálculos proporcionando como dato de entrada el mismo valor de otra ocasión anterior, pero, por simplificar, sólo proporcionó seis cifras decimales de precisión y puso el sistema a trabajar. Cuando volvió se encontró con unos resultados totalmente distintos. En un sistema dinámico como la atmósfera cambios pequeñísimos en una variable pueden resultar amplificados y provocar efectos enormes. En teoría, como dijo el propio Lorenz en su artículo: *"Can the flap of a butterfly's wing stir up a tornado in Texas*?" ([6]), el simple aleteo de una mariposa podría provocar un tornado en el polo opuesto del mundo.

Esta propiedad se puede aprovechar para cifrar una imagen como vamos a ver a continuación. Cogemos un valor de r que dé lugar a una función caótica como, por ejemplo, $r = 4$ y un valor de x_0, por ejemplo $x_0 = 0.6530$. Con estos valores de r y x_0 generamos la secuencia $\{x_n\}$:

$$x_{n+1} = f(x_n) = r\,x_n(1 - x_n) \quad \text{para } n = 1, 2, 3, \ldots$$

Tanto el valor de x_0 como de r son secretos y sólo serán conocidos por emisor y receptor. Los valores de $\{x_n\}$ serán números reales entre 0 y 1. A continuación los transformamos en otros valores $\{y_n\}$ que ahora serán números enteros entre 0 y 255. Para ello hacemos lo siguiente:

$$y_n = \text{mod}(E(1000\,x_n),\ 256)$$

siendo E la función parte entera y mod la función módulo.

Por otro lado supongamos que deseamos cifrar la imagen de la figura 4 que es de tamaño 256×256. Esta imagen se convertirá en un vector unidimensional escribiendo los píxeles de la imagen uno detrás de otro. Como $256 \times 256 = 65536$, resulta que la imagen quedará como un vector unidimensional del tipo:

$$I = \{i_1,\ i_2, \ldots,\ i_{65536}\}$$

A continuación calculamos:

$$i_n^* = y_n \oplus i_n \quad \text{para} \quad 1 \le n \le 65536$$

La expresión $i_n^* = y_n \oplus i_n$ lo que hace es una suma XOR (suma bit a bit módulo 2) entre el número y_n obtenido y el nivel de gris del píxel que nos toque i_n. Se comienza en la esquina superior izquierda y nos moveremos de izquierda a derecha y de arriba a abajo.

El resultado obtenido se muestra en la figura 8.

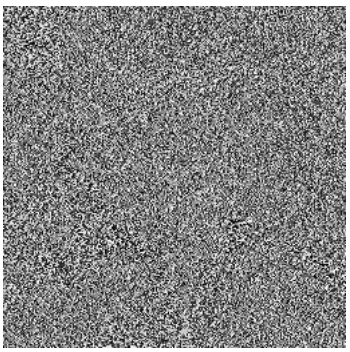

Figura 8: Imagen cifrada con función caótica.

Si la imagen cifrada de la figura 8 es descifrada con $r = 4$ y $x_0 = 0.6530$ se recupera exactamente la imagen original de la figura 4. Sin embargo, si un intruso, conocedor del método empleado pero no de las claves intentase descifrarla, ¿qué obtendría?.

Supongamos que prueba a descifrarla con $r = 4$ y $x_0 = 0.6531$, el resultado que obtiene se muestra en la figura 9.

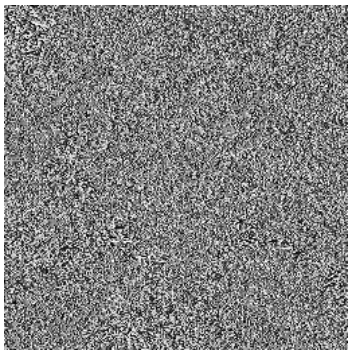

Figura 9: Imagen descifrada con clave incorrecta.

No sólo se puede utilizar la función logística para generar una secuencia caótica ya que existen una gran

variedad de procesos dinámicos que también generan secuencias caóticas.

Conclusiones

Hemos comprobado cómo nuestros alumnos de primer curso de Ingeniería Técnica Informática se interesan notablemente por el tema del cifrado y descifrado en general. En concreto este trabajo se ha dedicado al tema del cifrado de imágenes digitales que actualmente atrae el interés de muchos investigadores. Hemos contado en este trabajo algunas de las técnicas desarrolladas con nuestros alumnos de primer curso de Ingeniería Técnica en Informática.

En clases prácticas con ordenador, se les proporcionaba las imágenes cifradas y su trabajo consistía en descifrarlas. El trabajo se hacía por grupos de alumnos, favoreciendo así el trabajo colaborativo. Para ello se les describía minuciosamente el proceso seguido en el cifrado y las claves empleadas. Hemos visto la satisfacción mostrada por aquellos grupos que conseguían ser los primeros en descifrar la imagen propuesta.

De camino hemos conseguido que nuestros alumnos trabajen con herramientas matemáticas importantes como: aritmética modular, matrices, iteración de funciones, etc. También hemos conseguido concienciar a nuestros alumnos de las importantes aplicaciones de los conceptos matemáticos desarrollados en clase y cómo éstos se aplican en artículos de investigación recientemente publicados y en áreas de interés para la titulación que están cursando.

Referencias

[1] L. S. Hill, "Cryptography in an algebraic alphabet. The American Mathematical Monthly", Vol. 38, pp. 135-154, 1929

[2] I. A. Ismail et al.. "An efficient modified Hill Cipher adapted to image encryption". ICGST-CNIR Journal, Vol. 5, nº 2. pp. 53-62, 2006

[3] M. R. Zhang et al.. "T-matrix and its applications in image processing". IEEE Electronics Letters Vol. 40, nº 25, pp. 1583-1584, 2004

[4] C. Fu et al. "An improved chaos-based image encryption scheme". ICCS 2007. Lectures and Notes in Computer Science, Vol. 4487, pp. 575-582, 2007.

[5] H. Gao et al. "A new chaotic algorithm for image encryption". Chaos, Solitons and Fractals, Vol. 29, pp. 393-399. 2006

[6] H.O. Peitgen et al. "Chaos and Fractals. New frontiers of Science". Springer-Verlag, 1992.

Sitios en Internet

C. H. Lin, "Some Visual Cryptoschemes for Secret Images", trabajo fin de master , 2007. Consulta a 26 de noviembre de 2009.

http://ethesys.lib.fcu.edu.tw/ETD-search/view_etd?URN=etd-0704107-151337

Dirección de Contacto de los Autores:

Ángela Rojas Matas
Departamento de Matemáticas
Edificio Einstein
Campus de Rabanales
Universidad de Córdoba
Córdoba (14071)
España
e-mail: ma1romaa@uco.es

Alberto Cano Rojas
C) Madroño nº 2
Córdoba (14012)
España
e-mail: i52caroa@uco.es

Ángela Rojas Matas. Licenciada en Matemáticas. Doctora en Informática. Profesora del Departamento de Matemáticas de la Universidad de Córdoba (España) desde 1982. Ha participado en proyectos de innovación y mejora docente desarrollados en la Universidad de Córdoba. Autora de artículos y comunicaciones sobre innovación docente en Matemáticas.

Alberto Cano Rojas. Alumno de Ingeniería Informática de la Escuela Politécnica Superior de la Universidad de Córdoba. Alumno colaborador del Departamento de Matemáticas. Ha participado en proyectos de innovación y mejora docente de la Universidad de Córdoba. Becario de la Escuela Politécnica Superior de la Universidad de Córdoba durante el curso 2008-2009.