Encryption of Images with Key Matrixes of Different Sizes Applied to Hill Cipher

Erick Israel Villanueva Pulido
SEPI ESIME Zacatenco
Instituto Politécnico Nacional
52-55-5885 6548
ing.erickivp@gmail.com

Marco Antonio Acevedo M
SEPI ESIME Zacatenco
Instituto Politécnico Nacional
52-55-9199 3461
macevedo@ipn.mx

ABSTRACT
In this paper, different algorithms for the Hill Cipher are proposed as well as different sizes of key matrixes and intervals of the numbers that create them in order to reduce the encryption time of images. A second of video usually has 30 fps so to encrypt a video in Real-Time it is needed to encrypt 30 images in less than a second. The time of the encryption changes by modifying the size of the key matrix, and the form of how this matrix is used to encrypt the image, but this also affects the robustness of the Cipher. Key matrixes are created with different sizes and intervals of numbers and then tested encrypting images. The experimental analyses of the encryptions demonstrate the efficiency and sturdiness of some key matrixes over the others.

CCS Concepts
• Security and Privacy → Cryptography → Mathematical Foundations of Cryptography.

Keywords
Hill Cipher; Key Matrix; Dimensions; Inverse Matrix; Real-Time; Encryption; Decryption

1. INTRODUCCIÓN
En este artículo se propone utilizar el Cifrado de Hill empleando matrices llave de diferentes tamaños. Para crear las matrices se usan intervalos de números diferentes, los intervalos utilizados son 1 y 2, 1 al 7, 1 al 10 y 1 al 15. El objetivo de este trabajo es reducir el tiempo de cifrado de imágenes utilizando diferentes tamaños de las matrices así como los números para crearlas, sin disminuir su robustez. Al disminuir el tiempo de cifrado de una imagen este método pretende utilizarse para encriptar las imágenes de video. Se debe tomar en cuenta que un segundo de video usualmente consta de 30 imágenes, por lo que si se quiere cifrar video en tiempo real es necesario cifrar 30 imágenes en menos de un segundo.

El tamaño de matriz llave, así como el intervalo de números contenidos en ella, influye directamente en el tiempo de cifrado. Por lo que la elección adecuada de la matriz llave permitirá cifrar las 30 imágenes en el menor tiempo posible, lo que implica el menor retardo en la transmisión del video.

En la sección 2 se explica una técnica de cifrado matricial llamado Cifrado de Hill (CH). Se muestra en detalle los conceptos matemáticos necesarios para entender el algoritmo del CH.

En la sección 3 se muestran los diferentes pasos que se requieren para implementar el CH. En la sección 4 se realiza un análisis de los resultados obtenidos.

Por último en la sección 5 se muestran las conclusiones de este trabajo.

Las pruebas se realizaron en un equipo de cómputo con procesador Intel Core i3-2377M a 1.5 GHz, 4.00 GB de memoria RAM, sistema operativo Windows 7 Ultimate y MATLAB R2012A.

978-1-5090-2435-3/16/$31.00 ©2016 IEEE

2. CIFRADO DE HILL
El Cifrado de Hill (CH) [1] es un algoritmo conocido por su facilidad de implementación computacional y velocidad, éste emplea multiplicaciones entre datos ordenados en forma matricial y un par de matrices denominadas Matriz llave y Matriz llave inversa.

Una de las ventajas del CH es el de esconder el número de repetición de los pixeles de una imagen, mostrando un histograma con una distribución de valores casi iguales para todos los pixeles, de manera que un ataque utilizando probabilidad de ocurrencia de letras o valores de pixeles contra el mensaje o la imagen cifrada sea casi imposible. La robustez del cifrado se entiende como que tan uniformemente distribuido es el histograma. La robustez del cifrado depende de los valores que conforman la matriz llave y de que tan grande sea esta matriz, pues así toma más valores al mismo tiempo en las multiplicaciones eliminando la posibilidad de que el cifrado sea vulnerable en partes de la imagen con el mismo color, por ejemplo un fondo. De acuerdo con lo anterior, la matriz ideal para la robustez del cifrado debería ser del tamaño de la imagen, sin embargo, construir una matriz que cumpla con las condiciones para realizar el cifrado de Hill se va haciendo un problema matemático muy complejo a medida que se construyen matrices más y más grandes, y aún más el obtener la matriz inversa modular para poder descifrar la información, ya que sin esta se podrá cifrar la imagen, pero no se podrá recuperar la información original.

2.1 Algoritmo del Cifrado de Hill
El CH se basa en multiplicaciones con matrices. Para el caso de las imágenes el valor de sus pixeles se ingresa en la matriz $\boldsymbol{M}$. La matriz $\boldsymbol{M}$ es de tamaño *nxm* donde *n* es la dimensión de la matriz llave $\boldsymbol{K}$. La matriz llave $\boldsymbol{K}$ y la matriz llave inversa $\boldsymbol{K^{-1}}$ son siempre cuadradas y de tamaño *nxn*. En general el cifrado de Hill puede expresarse de la siguiente manera:

Para el cifrado:

$$\boldsymbol{C = mod\ m(K * M)} \quad \textbf{(1)}$$

Para el descifrado:

$$\boldsymbol{D = mod\ m\left(K^{-1} * C\right)} \quad \textbf{(2)}$$

Donde ***C*** es la matriz con el mensaje cifrado y ***D*** es la matriz con el mensaje recuperado. Es necesario hacer la congruencia en módulo ***m*** después de la multiplicación para que el resultado contenga valores válidos, en el caso de las imágenes el módulo es 256 para que en las matrices resultantes se tengan valores de gris válidos entre 0 y 255.

2.2 Matriz K

Las matriz llave ***K*** es el núcleo del cifrado de Hill, a partir de ella se obtiene la llave inversa $\boldsymbol{K^{-1}}$**.** No cualquier matriz puede ser la matriz llave. Para poder usar una matriz como matriz llave, esta debe ser invertible en módulo ***m*** (***m***=256 para el caso de las imágenes), ya que de no ser así el emisor podrá cifrar la imagen, pero el receptor no será capaz de descifrarla, por lo tanto las matrices que no sean invertibles en el módulo deseado no serán de utilidad [2]. Para asegurar que la matriz propuesta para cifrar es invertible, debe cumplir con lo siguiente:

$$\boldsymbol{Det(K) \neq 0} \quad \textbf{(3)}$$

$$\boldsymbol{gcd(Det(K), m) = 1} \quad \textbf{(4)}$$

2.3 Matriz K^{-1}

La matriz llave inversa $\boldsymbol{K^{-1}}$ se obtiene a partir de la matriz llave ***K***. Calcular la matriz inversa de una matriz dada se define como:

$$\boldsymbol{A^{-1} = Det(A)^{-1}(A^{*})^{T}} \quad \textbf{(5)}$$

Para calcular la matriz inversa modular de ***K***, se necesita tomar en cuenta que el inverso de la determinante es el inverso modular en módulo ***m*** de dicha determinante [3]. El inverso modular cumple con lo siguiente:

$$\boldsymbol{mod(N * N^{-1m}, m) = 1(mod\ m)} \quad \textbf{(6)}$$

3. DESARROLLO

Se expondrán los diferentes algoritmos empleados, así como la construcción de las matrices llave que fueron probadas para el cifrado de las imágenes. Se utilizó el primer cuadro del video ***xylophone.mpg*** que viene incluido en las librerías de Matlab el cual se muestra en la Figura 1, así como la imagen mostrada en la Figura 2 que posee un fondo homogéneo.

Figura 1. Primer cuadro del video *xylophone.mpg* que viene en Matlab transformado a escala de grises de tamaño 320x240.

Figura 2. Imagen con fondo homogéneo en escala de grises de tamaño 400x400.

Creación de Matrices K y K^{-1}

Para las pruebas del cifrado se hicieron 8 matrices llave, 4 matrices con el mismo intervalo de valores (Matrices conformadas por valores 1 y 2) que cambian de tamaño y otras 4 matrices del mismo tamaño (8x8) cuyo intervalo de valores con las que fueron creadas va cambiando.

Tabla 1. Relación del intervalo de los valores de las Matrices Llave y el tamaño máximo que fue posible crear con esos valores.

Intervalo de los valores de la Matriz *K*	**1-10**	**1-5**	**1-3**	**1-2**
Tamaño Máximo de la Matriz *K* que fue posible crear	**17x17**	**23x23**	**31x31**	**44x44**

Crear matrices llave grandes es complejo, mientras más grande es la matriz ***K*****,** más complicado se hace el obtener la matriz $\boldsymbol{K^{-1}}$ debido a que la complejidad para obtener la matriz transpuesta de la adjunta aumenta, así como el cálculo de la determinante, que comienza a llenarse de decimales que en algún punto comienzan a ser redondeados o truncados por el programa, y eso se convierte en un problema, haciendo que funcione hasta un límite del tamaño de las matrices llave, así como los intervalos de sus números. Este límite se puede observar en la Tabla 1. Si tan solo uno de los números de la matriz ***K*** o la matriz $\boldsymbol{K^{-1}}$ cambia en lo más mínimo de su valor original este par de matrices ya no serán de utilidad.

La matriz llave más grande creada fue de 44x44 conformada solo por valores 1 y 2, esto debido a que estos valores no afectan tanto los decimales como valores más grandes. Pasado el límite de tamaño visto en la Tabla 1 para cada intervalo de valores el programa entra en un ciclo sin fin creando matrices, probándolas y volviendo a empezar sin encontrar una matriz que sea de utilidad para el cifrado.

3.1 Algoritmo 1: Ordenar y Desordenar la Imagen en una Matriz nxm

El primer algoritmo para cifrar la imagen consiste en ordenar la imagen ***A*** en una Matriz *nxm*, de manera que se pueda hacer una multiplicación directa de matrices con una matriz llave ***K*** de dimensiones *nxn* y finalmente reordenar la matriz cifrada en el orden en que se encontraba en un principio. Este es el algoritmo que se utiliza para cifrar texto usualmente. A continuación se muestra un ejemplo para una matriz llave de 2x2:

Imagen Original **Matriz Llave**

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1x} \\ a_{21} & \vdots & \vdots & \vdots \\ a_{y1} & \cdot & \cdot & a_{yx} \end{bmatrix} \qquad K = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

Imagen Ordenada

$$Ao = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{(y-1)x} \\ a_{21} & a_{22} & \dots & a_{yx} \end{bmatrix}$$

Cifrado

$$C = K * Ao = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} * \begin{bmatrix} a_{11} & a_{12} & \dots & a_{(y-1)x} \\ a_{21} & a_{22} & \dots & a_{yx} \end{bmatrix}$$

Finalmente se reordena la imagen cifrada. Para el descifrado se repite el mismo proceso, ordenando en una matriz *nxm* y

multiplicando esta vez por la matriz $\boldsymbol{K^{-1}}$ y reordenando al final para obtener el cuadro recuperado.

3.2 Algoritmo 2: Multiplicación directamente sobre la imagen utilizando ciclos *for*

En este algoritmo para evitar perder tiempo ordenando y desordenando la imagen. Se propone emplear ciclos *for* para multiplicar cada ***n*** filas de la imagen por la matriz ***K*** hasta llegar al final de la matriz. El número de filas de la imagen debe ser múltiplo de la dimensión de la matriz ***K***, al final se obtiene una matriz cifrada de la misma dimensión que la imagen original. Para descifrar se repite el mismo proceso pero con la matriz $\boldsymbol{K^{-1}}$. Enseguida se puede observar el proceso para una matriz ***K*** de dimensiones 2x2:

Imagen Original **Matriz Llave**

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1x} \\ a_{21} & a_{22} & \dots & . \\ a_{31} & a_{32} & \dots & . \\ a_{41} & a_{42} & \dots & a_{4x} \end{bmatrix} \qquad K = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

Primera Multiplicación

$$C(1{:}2,:) = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} * \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1x} \\ a_{21} & a_{22} & \dots & a_{2x} \end{bmatrix}$$

Segunda Multiplicación

$$C(3{:}4,:) = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} * \begin{bmatrix} a_{31} & a_{32} & \dots & a_{3x} \\ a_{41} & a_{42} & \dots & a_{4x} \end{bmatrix}$$

3.3 Algoritmo 3: Creación de una matriz llave grande a partir de una matriz llave pequeña

En este último algoritmo se busca obtener una matriz llave ***D*** de dimensiones ***dxd*** donde ***d*** es la dimensión más grande de la imagen. En este caso la imagen es de 240x320 por lo que la matriz llave a crear debe ser de dimensiones 320x320. En este trabajo se propone una forma nueva de crear la matriz ***D***. La matriz ***D*** está conformada por la matriz llave ***K*** repetida el número de veces que sea múltiplo de las filas de la matriz ***D*** formando una diagonal, el resto de la matriz ***D*** serán ceros, esto para que en las multiplicaciones cada ***n*** filas sean multiplicadas solo una vez por la matriz llave ***K*** que conforma la diagonal de ***D***. También se crea una matriz $\boldsymbol{D^{-1}}$ de la misma forma que ***D*** solo que su diagonal estará formada por copias de $\boldsymbol{K^{-1}}$ en lugar de ***K***. Si la imagen no es cuadrada sus dimensiones se deben de ajustar hasta que esta sea cuadrada. Una vez creadas ***D*** y $\boldsymbol{D^{-1}}$ el procedimiento solo consiste en multiplicar directamente por ***D*** y luego aplicar el módulo ***m*** a la matriz resultante. Para descifrar solo queda multiplicar la matriz resultante por $\boldsymbol{D^{-1}}$ para finalmente aplicar el módulo ***m*** a la matriz descifrada. El proceso se ilustra a continuación para una matriz ***K*** de dimensiones 2x2:

Matriz Llave

$$K = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

Matriz D

$$D = \begin{bmatrix} \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \dots & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} & \dots & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\ \vdots & \vdots & \ddots & \vdots \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \dots & \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \end{bmatrix}$$

Cifrado

$$C = mod\ m(D * A)$$

Descifrado

$$P = mod\ m(D^{-1} * C)$$

4. PRUEBAS Y ANÁLISIS DE RESULTADOS

4.1 Prueba 1: Probando los 3 algoritmos con las mismas matrices K y K^{-1}

La primer prueba consistió en aplicar los 3 algoritmos vistos en la sección 3 con las mismas matrices llave y llave inversa. Las matrices usadas fueron de tamaño 40x40 y solo con valores 1 y 2 en ellas. Como se trataba de las mismas matrices llave, lo que se comparó en esta prueba fue el tiempo en que se cifraba un solo cuadro del video ***xylophone.mpg***. Para notar más la diferencia entre los tiempos se cifró el cuadro del video en los 3 canales RGB. Los resultados se muestran en la Tabla 2, estos resultados son el promedio de 10 pruebas para cada algoritmo.

Tabla 2. Algoritmos para el Cifrado de Hill y su tiempo promedio.

Algoritmo	1	2	3
Tiempo Promedio Cifrado	0.03379s	0.03208s	0.03944s

4.2 Prueba 2: Cifrando con matrices llave formadas por el mismo intervalo de valores pero con diferente tamaño

Para esta prueba se empleó el algoritmo 2, ya que fue el más rápido de los 3 probados en la prueba 1. En esta ocasión lo que se cambió fue el tamaño de las matrices. Los tamaños usados fueron 2x2, 4x4, 20x20 y 40x40. Se eligieron los tamaños por ser múltiplos de las 2 dimensiones del cuadro que se cifró (240x320). Los resultados se muestran en las figuras 3-6. Se puede observar que mientras más grande es la matriz mayor robustez ofrece en el cifrado. Los tiempos de cada cifrado se pueden apreciar en la Tabla 3. De igual manera es el promedio de 10 cifrados.

Tabla 3. Tiempos promedio de cifrado para diferentes tamaños de matrices llave.

Cifrado	2x2	4x4	20x20	40x40
Tiempo	0.01052s	0.0066s	0.0104s	0.0104s

Se observa que los tiempos con las matrices de 2x2, 20x20 y 40x40 son muy parecidos, pero la matriz de 4x4 muestra el menor tiempo de cifrado. En el caso de la matriz de 2x2 el tiempo aumenta debido a que realizan más multiplicaciones. Y en el caso de las matrices de 20x20 y 40x40 ocurre el mismo comportamiento de la prueba 1, donde al utilizar matrices de mayor dimensión aumenta el tiempo de cifrado. El cifrado no es el adecuado en las figuras 3 y 4, ya que la imagen cifrada tiene rasgos de la imagen, por lo que utilizar una matriz de 2x2 o 4x4 no es conveniente. En las figuras 5 y 6 se muestra un cifrado que homogeniza la imagen sin tener rasgos de la imagen original. Se puede concluir que para cifrar la imagen se deben utilizar ciclos *for* y matrices de tamaño de 20X20 o 40x40 ya que se obtienen resultados favorables y son rápidos en su ejecución.

Figura 3. Cifrado con la matriz llave 2x2 y valores 1-2.

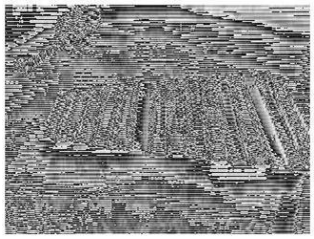

Figura 4. Cifrado con la matriz llave 4x4 y valores 1-2.

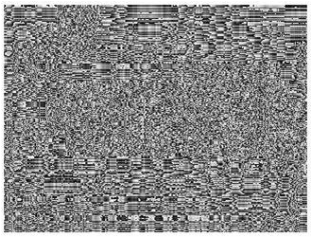

Figura 5. Cifrado con la matriz llave 20x20 y valores 1-2.

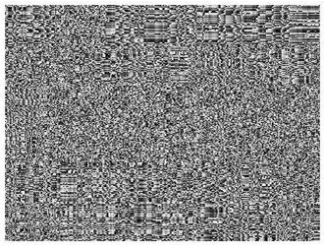

Figura 6. Cifrado con la matriz llave 40x40 y valores 1-2.

Una de las desventajas del Cifrado de Hill es que para partes de una imagen con el mismo color de pixeles (fondos homogéneos), el cifrado genera bloques cifrados iguales, de manera que no oculta todas las características de una imagen revelando patrones, tal como se ve en la figura 7.

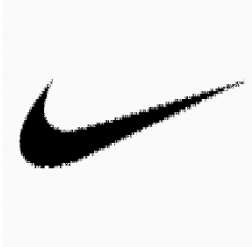

Figura 7. Cifrado con la matriz llave 4x4 y valores 1-2 para la imagen con fondo homogéneo.

Sin embargo si la matriz llave es más y más grande este problema disminuirá, eso se muestra en las figuras 8 y 9.

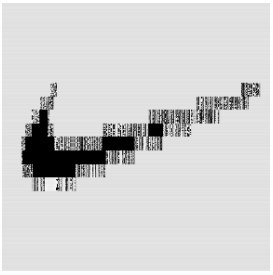

Figura 8. Cifrado con la matriz llave 20x20 y valores 1-2 para la imagen con fondo homogéneo.

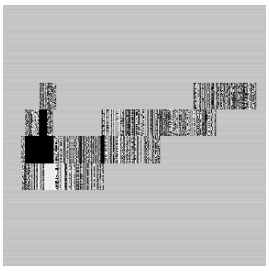

Figura 9. Cifrado con la matriz llave 40x40 y valores 1-2 para la imagen con fondo homogéneo.

Hay que tomar en cuenta que la imagen de la figura 2 es el caso más extremo al solo tener 2 colores (negro y blanco) y aun así en la figura 9 la imagen empieza a perder la forma de la imagen original empleando una matriz llave de un décimo de tamaño de la imagen a cifrar.

Dado que las matrices llave más grandes creadas en este artículo son de 44x44, no se puede hacer un cifrado con una matriz llave del mismo tamaño de la imagen, pero para ejemplificar se cambió el tamaño de la imagen de la figura 2 a una imagen de 40x40 y se cifró con una matriz llave del mismo tamaño, el cifrado se observa en la figura 10.

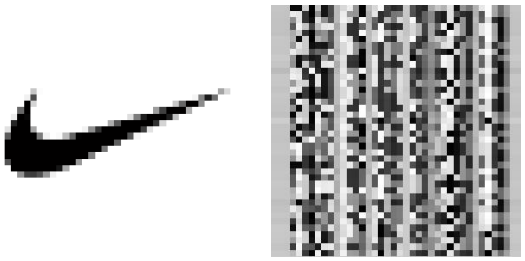

Figura 10. Imagen con fondo homogéneo de 40x40 y su cifrado correspondiente con la matriz llave 40x40 y valores 1-2.

De la figura 10 se observa que las características de la imagen original no son apreciables en el cifrado, el cual podría mejorar considerablemente si se utiliza un intervalo más amplio de números para formar la matriz llave.

4.3 Prueba 3: Cifrando con matrices llave de 8x8 pero formadas con diferentes valores

Para esta prueba se utilizaron 4 matrices llave de 8x8 para equilibrar el tiempo del cifrado y la robustez, donde ahora lo que cambia es el intervalo de valores con el que están formadas dichas matrices. Los cifrados con intervalos 1-2 y 1-7 presentan histogramas con distribuciones uniformes aún al tener intervalos pequeños, como se aprecia en las figuras 11 y 12, demostrando que utilizar matrices de 8x8 es una buena opción.

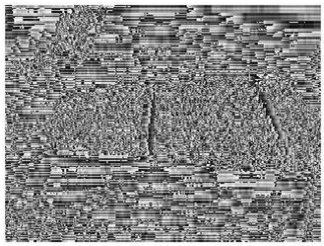

Figura 11. Cifrado con la matriz llave 8x8 y valores 1-2.

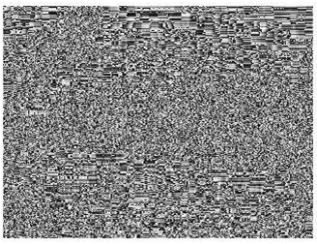

Figura 12. Cifrado con la matriz llave 8x8 y valores 1-7.

Comparando las imágenes cifradas de las figuras 11-14 se puede concluir que los mejores resultados se obtienen cuando se utilizan los intervalos de 1-10 y 1-15.

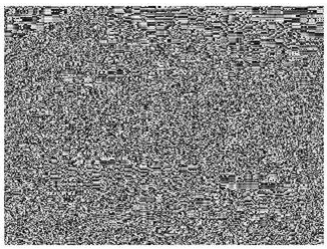

Figura 13. Cifrado con la matriz llave 8x8 y valores 1-10.

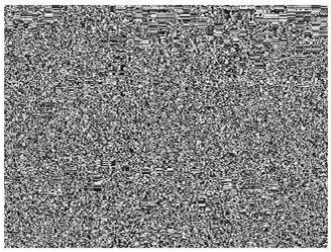

Figura 14. Cifrado con la matriz llave 8x8 y valores 1-15.

Los tiempos de los cifrados se encuentran desplegados en la Tabla 4, como se puede apreciar los tiempos son muy similares al cambiar los valores de las matrices de 8x8, pero se incrementa la robustez del cifrado al ampliar el intervalo de los valores de las matrices llave.

Tabla 4. Tiempos promedio de cifrado para diferentes intervalos de valores en matrices llave de 8x8.

Cifrado	1-2	1-7	1-10	1-15
Tiempo	0.0072s	0.0075s	0.0074s	0.0073s

5. CONCLUSIONES

Después del análisis de las pruebas con los algoritmos y matrices se observó que el algoritmo más rápido es el algoritmo 2 que realiza multiplicaciones sin desordenar la imagen por medio de ciclos *for*. Aunque el algoritmo 3 posee la matriz para cifrar más grande, la robustez del cifrado no es afectada en lo más mínimo a comparación de los otros 2 algoritmos, esto debido a que aunque la matriz llave es grande, es creada por matrices más pequeñas conservando el nivel de robustez de las matrices llave con la que fue creada.

Mientras más grande es la matriz llave mayor robustez ofrece el cifrado, sin embargo mientras más grande se vuelve la matriz más complejo se vuelve obtener su inversa, en el caso del programa utilizado se ve limitando a reducir el número de valores con el que se pueden crear las matrices llave, esa es la razón del por qué en la prueba 2 todas las matrices tenían solo valores 1 y 2.

Mientras más grande sea la matriz llave mayor será el tiempo del cifrado, el costo-beneficio es mayor tamaño-mayor robustez, pero se requiere de más tiempo en el proceso de cifrado.

Otra forma de incrementar la robustez del cifrado es el ampliar el intervalo de los valores de las matrices llave a utilizar y dado que el tiempo no cambia al modificar los valores es más conveniente hacerlo de esta manera. Al final el cifrado con la matriz 8x8 e intervalo de valores 1-15 es superior que el de una matriz más grande como la de 40x40 pero con un intervalo de valores más pequeño (1-2).

Si se quiere lograr una mejor robustez y no se está interesado en el tiempo que dure el cifrado se pueden crear matrices más grandes y con mayor intervalo de valores, sin embargo a matrices llaves más grandes mayor complejidad. Con el programa utilizado crean matrices de hasta 44x44 pero solo con valores 1 y 2, si se incrementa el intervalo el tamaño de las matrices decrece.

Los tiempos de la matriz 8x8 e intervalo 1-15 cumplen con el cifrado en tiempo real cifrando 30 imágenes en 0.219 segundos, lo cual está muy por debajo del segundo además de tener la mejor robustez de entre todas las demás matrices creadas.

6. AGRADECIMIENTOS

Los autores desean agradecer al Instituto Politécnico Nacional (COFAA, EDI, y SIP), CONACyT-SNI por el apoyo económico para el desarrollo de este trabajo

7. REFERENCIAS

[1] Hill, L. S., "Cryptography in an Algebraic Alphabet", *American Mathematical Monthly*, 36(6): pp.306-312, 1929.

[2] Rojas Ángela y Cano Alberto, "Una clase de aritmética modular, matrices y cifrado para la ingeniería". *Revista Iberoamericana de Educación Matemática*, No. 25: pp.89-107, 2011.

[3] Stanoyevitch, A., (2011) "*Introduction to Cryptography with Mathematical Foundations and Computer Implementations*", California, U.S.A: CRC Press, pp.156-166