El código de los números perfectos

Agustín Moreno Cañadas
Profesor Universidad Nacional de Colombia
Bogotá D.C, Colombia
amorenoca@unal.edu.co

Resumen

Vistos los Números Perfectos y los primos de Mersenne como un texto cifrado se describe un proceso de decripción.

1. Introducción

Aún con la reticencia de una parte de la comunidad matemática, el uso del computador para ayudar a obtener soluciones de problemas en diferentes áreas de la matemática ha venido en aumento en los ultimos años. Tal es el caso de las ecuaciones diferenciales parciales, la Teoría de números y la criptografía en donde la asociación de un criptosistema a un problema no resuelto de la teoría de números (por ejemplo la factorización de números grandes o la consecución de de una función que genere números primos) lo hace mas fuerte a un criptoanálisis. En diciembre del 2001 Peter Borwein y Löki Jorgenson publicaron un artículo,en el que se exploró el uso del computador para estudiar el problema propuesto por Bernoulli que consiste en encontrar algún tipo de patrón en el desarrollo binario de las cifras de π problema este que puede ser generalizado a cualquier número irracional y que aún permanece abierto. De hecho condujo a otros problemas abiertos como lo son el estudio de la normalidad de tales números. Cabe anotar que aún no se sabe si constantes como π, $\ln 2$, $\sqrt{2}$ ó e, son normales mientras que se presume la normalidad de constantes de la forma $\sqrt{n}$ con n= 2,3,5,6,7,8,10,11,12,13,14,15. Además los números que se saben son normales han sido construidos artificialmente como, por ejemplo, la constante de Erdös-Copeland 0.23571113... . La idea de Borwein y Jorgenson es encontrar algún tipo de estructura visible en la desarrollo binario de constantes como π, $\frac{22}{7}$, $\frac{1}{65537}$ y de e modulo 4, la observación de sus desarrollos binarios les llevó a concluir la aleatoriedad de las cifras binarias de π (estudiados 1600 de sus dígitos) y la presencia de estructuras regulares (patrones) en los números racionales.

El objetivo primordial de este artículo consiste en encontrar estructuras visibles en las cifras de los Números Perfectos bajo la hipótesis que constituyen un texto cifrado, creando a partir de estas un mapa de bits que no será más que una "sopa de imágenes"[1] dispuestas de tal forma que todas ellas se convierten en ideogramas cuya lectura permite el conocimiento del texto que dichas cifras encriptan.

2. Preliminares

2.1. Perfección

Los números perfectos son una de las fuentes de mayor número de conjeturas y problemas abiertos de la teoría de números. De hecho muchos de los matemáticos importantes desde Pitagoras pasando por Euclides hasta Paul Erdös, estudiaron alguna conjetura o problema relacionado con ellos. Por ejemplo el problema de examinar la primalidad, de forma computacionalmente eficiente tiene que ver directamente con la consecución de números perfectos. Este antiguo problema fue resuelto en el año 2002 por un grupo de matemáticos de la universidad de Kanpur en India. El algoritmo correspondiente se conoce como el algoritmo AKS(Agrawal,Kayal,Saxena) en honor a sus descubridores.

El interés general en esta clase de números, se debe sobre todo al hecho que desde que Euclides los definió formalmente en el libro IX de los Elementos, no ha sido probada su infinitud y los pocos que hasta la fecha han sido descubiertos han sido generados por primos denominados titánicos por su gran cantidad de, cifras. Examinar su primalidad requiere algoritmos suficientemente rápidos y en esa dirección apunta la utilización del algoritmo AKS.

Definición 1. *Un número natural es perfecto si es la suma de sus divisores sin incluirse él.*

Definición 2. *Todo primo de la forma* $2^n - 1$ *se llama un primo de Mersenne.*

Definición 3. *La función arimetica* $\sigma(n)$ *asigna a cada número natural* n *la suma de sus divisores positivos.*

Nota 1. $\sigma(n)$ *es multiplicativa y además de acuerdo a la definición 3 un numero es perfecto si* $\sigma(n)$=$2n$.

[1] por sopa de letras

Hasta ahora se conocen 42 numeros perfectos. El último fue descubierto el 27 de febrero del 2005 y tiene 15.632.458 cifras.

Algunos números perfectos son 6, 28, 496, 8128, $2^{13466916}(2^{13466917}-1)$, $2^{20996010}(2^{20996011}-1)$, $2^{24036582}(2^{24036583}-1)$, siendo el último encontrado $2^{25964950}(2^{25964951}-1)$.

Reseñamos ahora algunos teoremas y conjeturas relacionados con los números perfectos.

Teorema 1. *k es un número perfecto si y solamente si es de la forma $2^{n-1}(2^n-1)$ con 2^n-1 primo.*

Demostración. Supongamos que k= $2^{n-1}(2^n-1)$ con (2^n-1) primo . Entonces $\sigma(k)$= $\sigma(2^n-1)\sigma(2^n)$=$\frac{2^n-1}{2-1}2^n$= $2^n(2^n-1)$=2n.
Ahora bien, si k es un número perfecto par, lo podemos escribir en la forma k=$2^{p-1}r$ donde $p \geq 1$ y r impar positivo. Como k es perfecto tenemos que $2n$=$2^p r$=$\sigma(2^{p-1})\sigma(r)$=$\frac{2^p-1}{2-1}\sigma(r)$=$(2^p-1)\sigma(r)$. Por lo tanto, $\sigma(r)$= $2^p\frac{r}{2^p-1}$= $r+\frac{r}{2^p-1}$. Como $2^p r$= $2^{p-1}\sigma(r)$, entonces $(2^p-1)/2^p r$, y ya que $(2^p-1,\ 2^p)$=1 tenemos que $(2^p-1)/r$ y, en consecuencia, $\frac{r}{2^p-1}/r$.

Como $\sigma(r)$ es la suma de los divisores positivos de r, y como $\sigma(r) = r+\frac{r}{2^p-1}$ se sigue que r tiene únicamente dos divisores positivos y que $\frac{r}{2^p-1} = 1$.

Luego $r = 2^p-1$ es primo y $k = 2^{p-1}(2^p-1)$, como se quería probar. □

Teorema 2. *Si un número es de la forma 2^p-1, entonces p es primo.*

Demostración. Supongamos que p no es primo, es decir, $p = rs$ con $p \geq 2$, $s \leq p-1$. Por lo tanto,

$$2^p-1 = 2^{rs}-1 = ((2^r)^s-1 = (2^r-1)(2^{r(s-1)}+2^{r(s-2)}+...+1). \qquad (1)$$

De donde se deduce que 2^p-1 no es primo. Esto contradice la hipótesis que 2^p-1 es primo. □

Teorema 3. *La suma iterada de los dígitos de un número perfecto converge a 1.*

Demostración. Sea $s(n)$ la suma de los dígitos de n. Es facil ver que $s(n) \equiv n$ mod 9. Así que para probar el teorema, basta probar que los números perfectos son congruentes a 1 modulo 9. Si n es un número perfecto, entonces tiene la forma

$2^{p-1}(2^p-1)$ con p primo. Por lo que p es 2,3 o es congruente a 1 o 5 modulo 6. Note que el caso $p = 2$ (n =6) ha sido excluido. Finalmente, las potencias de 2 se repiten con periodo 6, así que módulo 9 n es congruente a uno de los números $2^{1-1}(2^1-1)$, $2^{3-1}(2^3-1)$, ó $2^{5-1}(2^5-1)$, los cuales son congruentes a 1 módulo 9. □

Teorema 4. *Sean p y q primos impares. Si $p/2^q-1$, entonces $p = 1$ $modq$ y $p = \pm 1$ $mod 8$*

Demostración. Sí $p/2^q-1$, entonces $2^q = 1$ *modp* y el orden de 2 *modp* divide al primo q, por lo que debe ser q. Por el teorema de Fermat el orden de dos también divide $p-1$, así $p-1 = 2kq$ y por tanto $2^{(p-1)/2} = 2^{qk} = 1$ *modp.*

Así 2 es residuo cuadrático *modp* y p =$\pm$1 *mod*8. □

El test de Lucas-Lehmer ha sido uno de los más usados para examinar la primalidad de los números de Mersenne y lo establecemos aquí como un teorema.

Teorema 5. *Si p es un primo impar, el número de Mersenne 2^p-1 es primo si y solamente si $2^p-1/S(p-1)$ donde $S(n+1) = S(n^2)$-2 y $S(1) = 4$.*

Otros algoritmos como el programa GLUCAS han sido usados recientemente para examinar primalidad, por ejemplo este programa requirió cinco dias para establecer la primalidad del último primo de Mersenne encontrado.

2.1.1. Algunas Conjeturas y Problemas Abiertos Sobre Números Perfectos.

1. Existencia de un perfecto impar.
2. La infinitud de los primos de Mersenne.
3. La infinitud de números de Mersenne compuestos.
4. ¿Los números de Mersenne son primitivos?

Describimos ahora el algoritmo AKS.

Input: integer $n \geq 2$

1. if (n is of the form a^b, $b \geq 2$) output COMPOSITE;
2. $r = 2$;
3. while $(r \leq (n-1))$
4. if $(gcd(n, r) \neq 1)$ output COMPOSITE;
5. if (r is prime);
6. let q be the largest prime factor of $r - 1$
7. if $(q \geq 4\sqrt{r}logn)$ and $(n^{\frac{r-1}{q}} \not\equiv 1\ modr)$
8. break
9. $r \leftarrow (r + 1)$;
10. for $a = 1$ to $2\sqrt{r}\ logn$
11. if $(x - a)^n \not\equiv (x^n - a)\ mod(x^r - 1, n)$ output COMPOSITE;
12. output PRIME.

Por lo que tenemos el siguiente

Teorema 6. *El algoritmo arriba descrito retorna PRIMO si y solamente si n es primo.*

3. Criptografía

Criptografía, significa escritura secreta y su estudio se divide en dos ramas principales: la criptografía clásica en la que el uso del computador no es esencial y la criptografía moderna, usada primordialmente para cifrar mensajes transmitidos a través de internet. Este último tipo de criptografía tambien posee dos categorías,una de ellas es la criptografía de clave simétrica, la cual es muy útil,cuando gran cantidad de texto debe ser cifrado, a este tipo de criptografía pertenecen los cifradores en bloque como DES, IDEA, AES(RINDAHL) ó Triple DES. La otra es la criptografía de clave publica ó asimétrica a la cual pertenecen cifradores como RSA, Merkle-Hellman, Rabin, Elgamal y otros basados en las propiedades de las curvas elipticas, usados sobre todo para autenticación de mensajes y la administración de claves.

Definición 4. *Un criptosistema es una quintupla* $S = \{P, C, K, E, D\}$ *definida así :*

1. P *es el conjunto finito de unidades de mensaje o textos claros.*
2. C *es el conjunto finito de textos cifrados.*
3. K *es el conjunto finito de claves.*
4. E *es el conjunto de funciones de encripción; esto es* $e \in E$ *si* $e_k : P \longrightarrow C$ *con* $e_k(x) = y \in C$
5. D *es el conjunto de funciones de decripción; esto es* $d \in D$ *si* $d_k : C \longrightarrow P$ *con* $d_k(y) = x \in C$ *y* $d_k(e_k(x)) = x$ *para cada* $k \in K$.

Nota 2. *Las funciones* d_k *y* e_k *deben ser computacionalmente eficientes.*

3.1. Criptografía Clásica

A continuación se describen algunos de los criptosistemas clásicos más conocidos.

Definición 5. *El criptosistema afín tiene las siguientes características:*

$P = C = Z_{26}$, $K = (a, b) \in Z_{26} \times Z_{26} : m.c.d(a, b) = 1$

$e_{(a,b)}(x) = ax + b \mod 26$, $d_{(a,b)}(y) = a^{-1}(y - b) \mod 26$ *si* $y \in Z_{26}$.

Nota 3. *En el caso afín* $k = (a, b)$ *y* $d_{(a,b)}(ax+b) = a^{-1}(ax+b-b) = x$ *sí* $x \in Z_{26}$

Definición 6. *Para* m *dado el criptosistema por permutación tiene las siguientes características:*

$P = C = (Z_{26})^m)$, K *es el conjunto de permutaciones del conjunto* $1 \dots m$ *el cual se nota* S_m

Si $\pi \in S_m$, *entonces* $e_\pi(x) = \pi(x)$, *para cada* $x \in P$ *y* $d_\pi(y) = \pi^{-1}(y)$, *para cada* $y \in C$.

Nota 4. *El criptosistema de permutación se llama de sustitución si* $m = 26$

Por ejemplo si consideramos $P = C = Z_{26}$, $m = 6$ *con* π *la permutación*

$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 5 & 2 & 6 & 4 & 2 \end{pmatrix}$ *siendo* π^{-1} *la permutación* $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 6 & 1 & 5 & 2 & 4 \end{pmatrix}$

Por lo que si el texto claro es shesellsseasshellbytheseashore dividiendolo en bloques de 6 letras y aplicando π *a cada uno de esos bloques obtenemos:*

EESLSHSALSESLSHBLEHSYEETHRAEOS.

El texto puede ser decriptado usando π^{-1}.

Aunque el criptosistema de sustitución es un criptosistema clásico, deben usarse herramientas computacionales para aplicarlo en imágenes binarias o en aquellas sin mucha información de la escala de grises. Por ejemplo, 012015642012012012189412012035 78 forma parte del texto cifrado cuando el texto claro es la figura 1.[2]

Figura 1: Texto Claro

3.2. Criptografía Moderna

En esta sección, se describen criptosistemas de clave publica o asimétrica y se dan algunos ejemplos de criptosistemas de clave privada.

Definición 7. *Un criptosistema que usa la misma clave para encriptar como para decriptar se llama de clave simétrica.*

[2]En este caso $P = C = Z_2$ y el ciframiento ha sido hecho sobre bloques de 144 pixeles.

Son criptosistemas de clave simétrica el cifrador por permutación, DES y Triple DES.

Definición 8. *El criptosistema RSA tiene las siguientes características:*

$P = C = Z_{26}$

$K = \{(n, p, q, a, b); n = pq, ab \equiv 1 \mod \varphi(n)\}$,$a, p$ *y* q *son secretos mientras que* n *y* b *son publicos con* $\varphi(n)$ *la función de Euler.*

$$e_k(x) = x^b \mod n \tag{2}$$

$$d_k(y) = y^a \mod n \tag{3}$$

Definición 9. *Una sucesión* $\boldsymbol{s} = s_1, s_2, \ldots, s_n$ *es supercreciente sí* $s_j \gneq \sum_{i=1}^{j-1} s_i$.

Definición 10. *El problema de partición consiste en encontrar un vector binario* $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$*, tal que para los enteros positivos* $T, s_1, s_2, \ldots, s_n$

$$\sum_{i=1}^{n} x_i s_i = T. \tag{4}$$

Definición 11. *El criptosistema de Merkle-Hellman, tiene las siguientes características:*

Sea $\boldsymbol{s} = (s_1, s_2, \ldots, s_n)$ *una sucesión supercreciente de enteros,*$p \gneq \sum_{i=1}^{j-1} s_i$*. primo y* $1 \leq a \leq p-1$*. Entonces para* $1 \leq i \leq n$,*definimos*

$$t_i = as_i \mod p. \tag{5}$$

Notamos $\boldsymbol{t} = (t_1, t_2, \ldots, t_n)$*. Sea* $P = \{0, 1^n\}$*,* $C = \{0, 1, \ldots n(p-1)\}$.

$K = (\boldsymbol{t}, \boldsymbol{s}, p, a)$ *donde* $\boldsymbol{s}, p, a$ *son secretos y* $\boldsymbol{t}$ *es público. Entonces*

$$e_k(x) = \sum_{i=1}^{n} x_i t_i. \tag{6}$$

Para $0 \leq y \leq n(p-1)$*, se define* $z = a^{-1}y \mod p$ *y se resuelve el problema de partición* $(s_1, s_2, \ldots, s_n, z)$*, obteniendo* $d_k(y) = ((x_1, x_2, \ldots, x_n))$

Por ejemplo si $\mathbf{s} = (2, 5, 9, 21, 45, 103, 215, 450, 946)$ es la lista de enteros supercreciente secreta, $p = 2003$ y $a = 1289$. Entonces
$\mathbf{t} = (575, 436, 1586, 1030, 1921, 569, 721, 1183, 1570)$. Luego si el texto claro es $x = (1, 0, 1, 1, 0, 0, 1, 1, 1)$ su correspondiente texto cifrado será $y = 6665$.

Si lo que se tiene es y, entonces $z = a^{-1}y \mod p = 1643$, por lo que se debe resolver el problema $(\mathbf{s}, z)$ de partición, para obtener de nuevo x.

Nota 5. *La seguridad del cifrador RSA, consiste en la imposibilidad que se tiene de factorizar, números grandes, mientras que la seguridad del criptosistema de Merkle-Hellman, está en la imposibilidad de resolver el problema de partición, tambień conocido como el problema de la maleta de viaje.*

4. Visión Computacional

Definición 12. *Un álgebra heterogénea, es una colección de conjuntos eventualmente vacios, con elementos posiblemente distintos, junto con un número finito de operaciones, las cuales proveen las reglas para que una combinación de elementos produzca uno nuevo.*

Definición 13. *Un álgebra homogenea es un conjunto dotado de un número finito de operaciones. Tales álgebras se llaman también conjuntos valuados.*

Los conjuntos valuados más usados en álgebra son los enteros, los números reales, los números complejos, los reales extendidos, los números binarios de longitud finita y los números reales no negativos extendidos.

Definición 14. *Sea $\boldsymbol{F}$ un conjunto valuado y $\boldsymbol{X}$ un conjunto. Una $\boldsymbol{F}$ imagen valuada sobre $\boldsymbol{X}$ es un elemento sobre $\boldsymbol{F}^{\boldsymbol{X}}$. Dada una imagen $\boldsymbol{F}$ valuada $a \in F^{\boldsymbol{X}}$[3], entonces $\boldsymbol{F}$ se llama el rango de todos los posibles valores de a y $\boldsymbol{X}$ es el dominio espacial de a.*

A veces el gráfico de $a \in \mathbf{F}^{\mathbf{X}}$ se nota $\mathbf{a}$. El gráfico de una imagen es referido como la representación de la estructura de datos. Dada la representación de estructura de datos $\mathbf{a} = \{(x, \mathbf{a}(x)) : x \in \mathbf{X}\}$, entonces un elemento $(x, \mathbf{a}(x))$ de la estructura de datos se llama elemento de la figura o pixel. la primera coordenada x de un pixel se llama la localización del pixel, mientras que la segunda coordenada $\mathbf{a}(x)$ se denomina el valor del pixel en la localización x.

4.1. Operaciones Inducidas Sobre Imágenes.

Las operaciones inducidas sobre imágenes $\mathbf{F}$ valuadas son aquellas inducidas naturalmente por el sistema álgebraico $\mathbf{F}$. Por ejemplo si γ es una operación binaria

[3]Por lo que $a : C \longrightarrow P$

sobre $\mathbf{F}$, entonces γ induce una operación binaria - denotada por $\gamma-$ sobre $\mathbf{F}^{\mathbf{X}}$ definida de la siguiente forma:

Si $\mathbf{a}, \mathbf{b} \in \mathbf{F}^{\mathbf{X}}$. Entonces

$$\mathbf{a}\gamma\mathbf{b} = \{(x, \mathbf{c}(x) : \mathbf{c}(x) = \mathbf{a}(x)\gamma\mathbf{b}(x), x \in \mathbf{X}\}. \tag{7}$$

Por ejemplo si $\mathbf{a}, \mathbf{b} \in \mathbf{R}^{\mathbf{X}}$ en donde $\mathbf{X}$ es el conjunto de los números reales dotado de las operaciones $(R, +, \cdot, \wedge, \vee)$, reemplazando γ por las operaciones binarias $+, \cdot, \wedge, \vee$ se obtienen las operaciones binarias básicas

$$\begin{aligned}
\mathbf{a} + \mathbf{b} &:= \{(x, \mathbf{c}(x)) : \mathbf{c}(x) = \mathbf{a}(x) + \mathbf{b}(x)\} \\
\mathbf{a} \cdot \mathbf{b} &:= \{(x, \mathbf{c}(x)) : \mathbf{c}(x) = \mathbf{a}(x) \cdot \mathbf{b}(x)\} \\
\mathbf{a} \wedge \mathbf{b} &:= \{(x, \mathbf{c}(x)) : \mathbf{c}(x) = \mathbf{a}(x) \wedge \mathbf{b}(x)\} \\
\mathbf{a} \vee \mathbf{b} &:= \{(x, \mathbf{c}(x)) : \mathbf{c}(x) = \mathbf{a}(x) \vee \mathbf{b}(x)\} \\
\mathbf{a} + \mathbf{b} &:= a \in \mathbf{R}^{\mathbf{X}}
\end{aligned} \tag{8}$$

Además de las operaciones binarias entre imágenes, la operación binaria γ induce sobre $\mathbf{F}$ las siguientes operaciones escalares:

Para $k \in \mathbf{F}$ y $\mathbf{a} \in \mathbf{F}^{\mathbf{X}}$, y sí en particular, $k \in \mathbf{R}$

$$\begin{aligned}
k\gamma a &:= \{(x, \mathbf{c}(x)) : \mathbf{c}(x) = k\gamma\mathbf{a}(x)\} \\
\mathbf{a}\gamma k &:= \{(x, \mathbf{c}(x)) : \mathbf{c}(x) = \mathbf{a}(x)\gamma k\} \\
ka &:= \{(x, \mathbf{c}(x)) : \mathbf{c}(x) = k\mathbf{a}(x)\} \\
k + \mathbf{a} &:= \{(x, \mathbf{c}(x)) : \mathbf{c}(x) = k + \mathbf{a}(x)\} \\
k\mathbf{a} &:= \mathbf{a}k \\
k + \mathbf{a} &:= \mathbf{a} + k
\end{aligned} \tag{9}$$

Definición 15. *La topología de von Neumann de un conjunto discreto $\boldsymbol{X}$ es la topología generada por vecindades del tipo $N : \boldsymbol{X} \longrightarrow 2^{\boldsymbol{Z}^2}$ con*

$$N(x) = \{y : y = (x_1 \pm j, x_2) \quad ó \quad y = (x_1, x_2 \pm k), j, k \in \{0, 1\}\}. \tag{10}$$

Las vecindades de von Neumann, se llaman 4-vecindades.

Definición 16. *La topología de Moore de un conjunto $\boldsymbol{X}$ es generada por vecindades del tipo $M : \boldsymbol{X} \longrightarrow 2^{\boldsymbol{Z}^2}$, con*

$$M(x) = \{y : y = (x_1 \pm j, x_2 \pm k)), j, k \in \{0, 1\}\}. \tag{11}$$

Las vecindades de Moore, se llaman 8-vecindades.

Nota 6. *La figura 2 muestra los dos tipos de vecindades, generando estas topologías.*

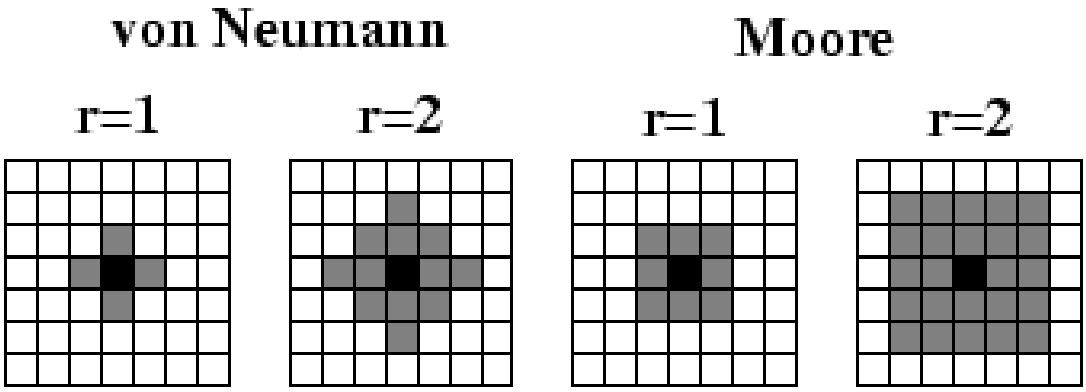

Figura 2: Vecindades de Moore y von Neumann de radios 1 y 2.

4.2. Procesamiento de Imágenes.

El propósito del procesamiento de imágenes es dar la apariencia visual de una imagen de tal forma que sea mas facil su interpretación para el ojo humano o para el análisis de una maquina.

4.2.1. Suavización de imágenes Binarias por Asociación

El objetivo de la suavización es reducir los efectos del ruido en una imagen binaria. La idea básica de este procedimiento es que las unidades debidas al ruido son diseminadas ó esparcidas,mientras que las unidades que dan la información tienden a mantenerse juntas [4]. La imagen original es particionada en regiones rectangulares. Si el número de unos excede cierta cota, entonces la región se mantiene invariante, en la alternativa los unos se transforman en ceros. Las regiones por tanto son tratadas como celdas: a una celda se le asigna un uno si existe un uno en la correspondiente región y un cero en la alternativa. Esta nueva colección de celdas puede verse como una imagen con baja resolución. La minima pixelización de la imagen con baja resolución y la imagen original proveen la suavización de la imagen original. La formulación algebraica de este proceso es como sigue:

[4]Aquí una vecindad que consta solo de unos se llama una unidad.

Sea T una cota y $\mathbf{a} \in \{0,1\}^{\mathbf{X}}$ la imagen fuente con $\mathbf{X} \subset \mathbf{Z}^2$. Para un entero fijo $k \geq 2$, definimos una vecindad funcional $N(k) : \mathbf{X} \longrightarrow 2^{\mathbf{X}}$, por

$$[N(k)](y) = \left\{ x \in \mathbf{X} : \left\lfloor \frac{x}{k} \right\rfloor = \left\lfloor \frac{y}{k} \right\rfloor \right\}. \tag{12}$$

Aquí $\left\lfloor \frac{x}{k} \right\rfloor$ significa que sí $x = (x, y)$, entonces. $\left\lfloor \frac{x}{k} \right\rfloor = \left(\left\lfloor \frac{x}{k} \right\rfloor, \left\lfloor \frac{x}{k} \right\rfloor \right)$.

La imagen suavizada $a_1 \in \{0,1\}^{\mathbf{X}}$ se calcula usando el hecho

$$a_1 := \mathbf{a} \wedge_{\lambda \geq T} (\mathbf{a} \oplus N(k)). \tag{13}$$

Si se requiere recurrencia entonces hacemos:

$$a_1 := \mathbf{a} \wedge_{\lambda \geq T} (a_{i-1} \oplus N(k+i-1)).^5 \tag{14}$$

La siguiente formulación reintroduce pixels con valor 1 que han sido eliminados en los pasos previos

$$a_1 := a_{i-1} \wedge_{\lambda \geq T} (a_{i-1} \oplus N(k+i-1)). \tag{15}$$

4.3. Detección de bordes y fronteras en imágenes binarias.

Un punto de la frontera de una imagen binaria es un punto con la propiedad de que toda vecindad intersecta el objeto y su complemento.

Para $\mathbf{a} \in \{0,1\}^{\mathbf{X}}$, sea $\mathbf{A}$ el soporte de $\mathbf{a}$. La frontera de la imagen $\mathbf{b} \in \{0,1\}^{\mathbf{X}}$ se clasifica por su conectividad y según que $\mathbf{B} \subset \mathbf{A}$ ó $\mathbf{B} \subset \mathbf{A}'$. Aquí $\mathbf{B}$ es el soporte de b.

1. La imagen b es una imagen 8-frontera exterior si $\mathbf{B}$ es 8-conexo, $\mathbf{B} \subset \mathbf{A}'$ y $\mathbf{B}$ es el conjunto de puntos en $\mathbf{A}'$, cuyas 4-vecindades intersectan $\mathbf{A}$. Esto es

$$\mathbf{b}(x) = \begin{cases} 1 & N(x) \cap \mathbf{A} \neq \emptyset, \quad x \in \mathbf{A} \\ 0 & \text{En la alternativa.} \end{cases}$$

[5] $\oplus$ es ó exclusivo

2. La imagen b es una imagen 8-frontera interior si $\mathbf{B}$ es 8-conexo, $\mathbf{B} \subset \mathbf{A}$, y $\mathbf{B}$ es el conjunto de puntos en $\mathbf{A}$ cuyas 4-vecindades intersectan $\mathbf{A}'$. La 8-frontera interior puede expresarse como

$$\mathbf{b}(x) = \begin{cases} 1 & N(x) \cap \mathbf{A}' \neq \emptyset, \quad x \in \mathbf{A} \\ 0 & \text{En la alternativa.} \end{cases}$$

3. La imagen b es una imagen 4-frontera exterior si $\mathbf{B}$ es 4-conexo, $\mathbf{B} \subset \mathbf{A}'$ y $\mathbf{B}$ es el conjunto de puntos en $\mathbf{A}'$ cuyas 8-vecindades intersectan $\mathbf{A}$. Esto es

$$\mathbf{b}(x) = \begin{cases} 1 & M(x) \cap \mathbf{A} \neq \emptyset, \quad x \in \mathbf{A}' \\ 0 & \text{En la alternativa.} \end{cases}$$

4. La imagen b es una imagen 4-frontera interior sí $\mathbf{B}$ es 4-conexo, $\mathbf{B} \subset \mathbf{A}$, y $\mathbf{B}$ es el conjunto de puntos en $\mathbf{A}$ cuyas 8-vecindades intersectan $\mathbf{A}'$. La 4-frontera interior puede expresarse como

$$\mathbf{b}(x) = \begin{cases} 1 & M(x) \cap \mathbf{A}' \neq \emptyset, \quad x \in \mathbf{A} \\ 0 & \text{En la alternativa.} \end{cases}$$

4.4. Transformada de la distancia

La transformada de la distancia, asigna a cada pixel característico de una imagen binaria, un valor igual a la menor distancia que hay desde él hasta el conjunto de pixels no característicos. El algoritmo puede ser desarrollado, en paralelo o secuencialmente.

Un subconjunto atenuado de la imagen original puede ser obtenido de la transformada de la distancia, extrayendo la imagen que consta unicamente de los máximos de dicha transformada. Este conjunto se llama el esqueleto de la distancia y tiene la propiedad de que la imagen binaria original puede ser obtenida a partir de él. La formulación algebraica está dada por:

$$\mathbf{b} := (\mathbf{a} \wedge_{\prec} t) \wedge_{\succ} t' \tag{16}$$

La siguiente figura, muestra la distancia del tablero de ajedrez(conocida también como la distancia del máximo).

[6] Aquí, $\lambda_{\leq b}(\mathbf{a}) = \{(x, \mathbf{c}(x)) \mid x \in \mathbf{X}\}$ con

$$\mathbf{c}(x) = \begin{cases} 1 & \mathbf{a}(x) \leq \mathbf{b}(x) \\ 0 & \text{En la alternativa.} \end{cases}$$

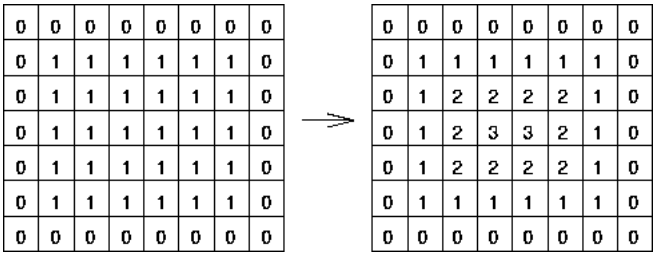

Figura 3: Distancia del tablero de ajedrez.

La siguiente figura muestra el efecto de la transformada de la distancia en una imagen binaria.

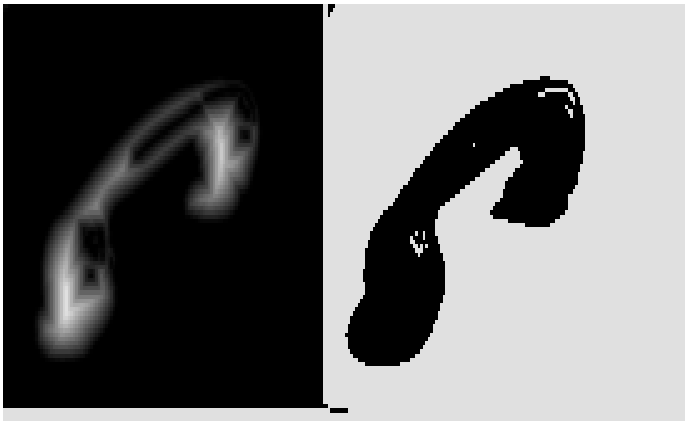

Figura 4: Efecto de la transformada de la distancia.

5. El Código De Los Números Perfectos.

En esta sección nos dedicamos a descifrar los números perfectos, vamos a suponer que el criptosistema del cual ellos resultan, tiene como texto cifrado los dígitos y que además el texto claro $P = Z_2^m$ con m fijo. De hecho este criptosistema es una combinación de los cifradores Merkle-Hellman(una variación) y sustitución, en adelante llamaremos a este criptosistema Isis. A continuación describimos los pasos requeridos para desarrollar su proceso de decripción.

1. Se aplica una permutación inicial IP.

2. Una secuenciación de las cifras se realiza a fin de convertir el arreglo inicial en un mapa de bits.

3. Una permutación FP se aplica al mapa de bits.

4. El resultado de aplicar FP al mapa de bits es una colección de imagenes entrelazadas que constituyen el texto cifrado, la disposición de las imagenes en el arreglo hace de cada una de ellas un ideograma cuya lectura le da el nombre al mapa de bits.

5.1. La secuenciación.

Llamamos secuenciación al proceso que nos permite transformar un conjunto de dígitos en una lista binaria. Tal secuenciación no es mas que un criptoanálisis, de una variación del criptosistema de Merkle-Hellman, puesto que en este caso el texto cifrado está formado por dígitos y no se trabaja con listas supercrecientes de enteros.

La idea esencial de este proceso consiste en considerar que la lista de enteros que va a tratarse esta conformada por listas de números $(s_1, s_2, \ldots, s_k, t_1, t_2, \ldots, t_n)$ que satifacen la condición de igualdad:

$$\sum_{i=1}^{k} s_i = \sum_{i=1}^{n} t_i. \tag{17}$$

A continuación, se describe en detalle el proceso de decripción.

Sea $n = \sum_{i=0}^{k} \alpha_i 10^i$. Dispuesto en un arreglo matricial determinado por una permutación IP, cada dígito está en una celda p_j y cada celda es un pixel del mapa de bits con soporte nulo[7]. Entonces si :

$\alpha_k = \alpha_{k-1}$, $p_k = 1$ y $p_{k-1} = 0$.

Si $\alpha_k \neq \alpha_{k-1}$ entonces se hace:

$$\alpha_k + \alpha_{k-1}. \tag{18}$$

[7]El valor del pixel $p_k = 0$ para todo k

Si $\alpha_k + \alpha_{k-1} = \alpha_{k-2}$, entonces $p_k = p_{k-1} = 1$ y $p_{k-2} = 0$. Si $\alpha_k + \alpha_{k-1} \neq \alpha_{k-2}$ hacemos :

$$\alpha_k + \alpha_{k-1} + \alpha_{k-2}. \tag{19}$$

Si $\alpha_k + \alpha_{k-1} + \alpha_{k-2} = \alpha_{k-3}$, entonces $p_k = p_{k-1} = p_{k-2} = 1$ y $p_{k-3} = 0$. En la alternativa se compara $\alpha_k + \alpha_{k-1}$ con $\alpha_{k-2} + \alpha_{k-3}$, si son iguales, entonces $p_k = p_{k-1} = 1$ y $p_{k-2} = p_{k-3} = 0$.

Si se llega al caso $\sum_{i=0}^{3} \alpha_{k-i} \neq \alpha_{k-4}$, entonces se compara $\sum_{i=2}^{4} \alpha_{k-i}$ con $\alpha_{k-1}+\alpha_k$ o $\sum_{i=0}^{2} \alpha_{k-i}$ con $\alpha_{k-3}+\alpha_{k-4}$. Si no es satifecha la condición (17), entonces se eligen sublistas $(\alpha_k, \ldots, \alpha_{k-j})$, $(\alpha_{k-t}, \ldots, \alpha_{k-l})$ de $(\alpha_k, \ldots, \alpha_{k-10})$, con $j \leq 5$, $l - t \leq 5$, $1 \leq t \leq 5$, y :

$$\sum_{i=0}^{j} \alpha_{k-i} = \sum_{i=t}^{l} \alpha_{k-i}. \tag{20}$$

Se selecciona la comparación que involucre el menor número de celdas, teniendo en cuenta que prevalece el primer procedimiento. En el caso que los dos procedimientos descritos exhiban listas de dígitos con igual longitud, para seleccionar los pixeles a los que se le asigna cero o uno, de tal forma que sí $(\alpha_s, \alpha_{s+1}, \ldots, \alpha_u) \in \{\alpha_q, \ldots, \alpha_r\} \cap \{\alpha_h, \ldots, \alpha_m\}$, entonces
$p_q = p_{q-1} = p_{q-2} = \ldots = p_s = \ldots = p_u = 1$, mientras que $p_{u+1} = \ldots = p_m = 0$, sí en el arreglo $q \leq s \leq r \leq h \leq m$.

El proceso puede continuar hasta encontrar el término α_{k-9} o décimo termino de n.

Si $\sum_{i=0}^{9} \alpha_{k-i} \neq \alpha_{k-10}$. Entonces comparamos las sumas de dos hasta ocho términos, si tampoco se satisface la condición de igualdad, se repite el proceso con nueve términos de la cadena $(\alpha_k, \ldots, \alpha_{k-10})$. Sí tampoco es sastisfecha la condición de igualdad a este nivel, agregamos el termino α_{k-11} y se repite el proceso, contando entre los términos que pueden sumarse aquellos con $p_t = 0$ y trasladando el índice k[8] cada vez que la condición de igualdad se satisfaga.

Si no es posible satisfacer la condición (17) con la lista $(\alpha_k, \ldots, \alpha_{k-19})$ de n, el proceso de secuenciación falla.

5.2. Extracción de Imágenes.

Una vez realizada la secuenciación, se obtiene un mapa de bits de tamaño $m \times n$. Tal arreglo se divide en subarreglos de tamaños $m_1 \times n_1, m_2 \times n_2, \ldots, m_t \times n_t$,

[8] Sí en la posicion j-esima con $j \gneq 11$, entonces $k \to j$

con $\sum_{i=1}^{t} m_i = m$ y $\sum_{i=1}^{t} n_i = n$.

Se aplica una permutación PF a dichos subarreglos y el resultado es el texto descifrado.

El texto claro obtenido es una colección de imágenes con la propiedad que si $\mathbf{a}(x)$ es una de tales imágenes y $\mathbf{b}(x)$ es la imagen que se obtiene al rotar $\mathbf{a}(x)$ 180 grados, entonces :

$$\mathbf{a}(x) + \mathbf{b}(x) = \mathbf{a}(x). \tag{21}$$

Nota 7. *La situación de la ecuación (20), se conoce en visión computacional como una ambigüedad. Esto es, un par de imágenes compartiendo el mismo espacio pero con más de un significado o apariencia.*

A continuación se describe en detalle el algoritmo que permite extraer las imágenes del mapa de bits.

5.2.1. Algoritmo.

El objetivo de este procedimiento es el de obtener la frontera de cada una de las imágenes que conforman el mapa de bits obtenido en el paso anterior.

Sea X_k, una imagen binaria a la que le es asociado un conjunto de parejas de pixeles, no necesariamente distintas

$$M = \{\overset{\leftarrow}{\leftarrow}(m_{il}, m_{ir})\overset{\mapsto}{\mapsto}, m_{il} \neq m_{ir}, 1 \leq i \leq m\}. \tag{22}$$

m_{il} se llama, el marcador izquierdo de la imagen X_k en la posición, il. m_{ir} se llama el marcador derecho de la imagen X_k en la posición ir, la flecha en el parentesís indica que el algoritmo debe ser desarrollado a partir de él en una dirección indicada hacia arriba ó hacia abajo por simplicidad notaremos el marcador izquierdo hacia arriba en la forma $\mapsto m_{ij}$[9].

Nota 8. *La presencia de los marcadores, hace de este algoritmo, un código en cadena, utilizados en el procesamiento de imágenes binarias para reconocer contornos. El algoritmo propuesto en este trabajo es una solución alternativa a este tipo de problema. Pués hasta ahora ha sido usado para extraer las imágenes que el cifrador Isis ha encriptado, usando números enteros.*

[9]Las flechas,$\longleftarrow, \longmapsto$ indican que el algoritmo debe desarrollarse en todos los bloques en los que se divide el mapa de bits en esa dirección

Se requieren tres etapas para desarrollarlo.

1. Movimiento.
2. Eliminación de apéndices.
3. Podamiento.

5.2.2. Movimiento

En esta etapa, el algoritmo recorre el arreglo a partir de los marcadores y selecciona una frontera temporal para la imagen X_k, la cual notaremos en adelante X'. En este caso basta describir el comportamiento del algoritmo en dirección vertical [10]. Supondremos que el mapa de bits $a(X)$ está dividido en los bloques B_i, determinados por el proceso de encripción. Tales bloques constituidos por m_i filas, numeradas desde abajo hacia arriba y n_i columnas numeradas de izquierda a derecha. Por las caracteristicas del algoritmo, cada paso se realizará bloque por bloque desde el correspondiente marcador.

Definición 17. *$X \subseteq a(X)$ es arcoconexo si cada par de puntos $a_i, a_j \in X$, con valores de pixel $p(a_i) = p(a_j) = 0$, pueden conectarse por medio de una trayectoria admisible σ con la propiedad $a_t \in \sigma$, entonces $p(a_t) = 0$.*

Nota 9. *La trayectoria σ, con punto inicial a_i y punto final a_j, suele escribirse $\sigma = \overline{a_i a_j}$. Mientras que el conjunto de trayectorias admisibles conectando a_i con a_j se nota S.*

Definición 18. *Si d es la distancia del taxista y $\sigma \in S$, la longitud de σ, se nota $\|\sigma\| = d(a_i, a_j)$.*

Nota 10. *Sea $a(X)$, un mapa de bits rectangular, $A \subseteq a(X)$ y $a_{ij} \notin A$, con valor de pixel $p(a_{ij}) = 0$, entonces h_i es el elemento de A en la i-esima fila, con valor de pixel $p(h_i) = 0$, tal que $d(a_{ij}, A) = \|\sigma\|$, para alguna trayectoria σ, $\overline{a_{ij}h_i}$ admisible. Si no existe tal σ, entonces $h_i \in A$, debe ser tal que, $d(h_i, a_{ij}) = d(A, a_{ij})$.*

Las definiciones y notas anteriores, permiten describir un movimiento, en el i-esimo bloque de la siguiente forma :

[10] Hacia arriba, pues el movimiento en otra dirección tambien sera vertical luego de una rotación apropiada.

Sea ${}^{\leftarrow}m$, un marcador de X' en la posición ij y R_k, el k-esimo renglon del i-esimo bloque. Entonces se genera el conjunto H, sobre el cual actua el algoritmo de la siguiente forma :

$$H = \{h_t : h_t \in R_t, i+1 \leq t \leq n_i\} \tag{23}$$

Si notamos $A(a_{ij}) = p(a_{ij})$, la salida del algoritmo en cualquiera de sus etapas, entonces durante un movimiento si σ es $a_{ij}h_{i+1}$-admisible de mínima longitud, entonces para cada $a_{ik} \in \sigma$ se tiene :

$$A(a_{ik}) = \begin{cases} p(a_{ik}) & j \leq k \quad \text{ó} \quad p(a_{ik}) = 1 \\ p(a_{ik}) + 1 & p(a_{ik}) = 0 \quad \text{y} \quad k \lneq j. \end{cases}$$

Luego, hacemos el cambio $h_{i+1} \longleftrightarrow a_{ij}$ y aplicamos el algoritmo a R_{i+1}.

En general se tiene que para $h_t, \quad i+1 \leq t \leq n_i, \quad h_t \longleftrightarrow a_{ij}$. Para aplicar el algoritmo A a cada fila del i-esimo bloque.

Nota 11. *$H \subseteq X'$, y a cada $h_t \in H$, le és asociada una curva σ_t, $\overline{a_{ij}h_t}$-admisible, de menor longitud, tal que $h_s \in \sigma_t, \quad i \leq s \leq t$, la admisibilidad implica también que el tramo de σ_t, que conecta los puntos, h_{t-1}, con h_t, son de la forma a_{t-1j} ó a_{tj}, lo cual implica que los elementos h_t, están todos contenidos en la misma componente arcoconexa de X'.*

Los elementos $a_{ij} \in X'$, generados por un movimiento, se clasifican en tres categorías, las cuales definimos a continuación.

Definición 19. *Un **puente**, es el elemento $a_{tj} \in b(X)$, donde $h_t \in H, \delta = Min\{d(h_t, h_{t-1}), d(h_t, h_{t+1})\} \gneq 2$ y $j = Max\{j_1, j_2\}$, si $j_i, i = 1, 2$ es la columna que le corresponde al elemento $h_{t\pm1}$, en el arreglo.*

Un puente cambia temporalmente su valor de pixel, para extender la arcoconexidad, de la componente a la cual pertenenece h_{t-1}.

Definición 20. *$a_{ij} \in X'$ con valor de pixel $p(a_{ij}) = 0$, es un **apéndice** si $p(a_{i+1j}) = p(a_{i-1j}) = 1$ y para a_{kj} el primero en la columna j-esima con valor de pixel $p(a_{kj}) \neq 1$, $Sop\{a_{lj} : i+1 \leq l \leq k\} \geq 2$ ó a_{ij}, con valor de pixel $p(a_{ij}) = 0$ satisface :*

$$a_{ij} \in \{a_{mj} \in b(X) : d(a_{mj}, X') \gneq 2\}. \tag{24}$$

Definición 21. *$a_{ij} \in X'$ con valor de pixel $p(a_{ij}) = 0$, es un **crecimiento** si $p(a_{ij+1}) = 0$.*

Nota 12. *La caracterización dada en la definición 21 esta ajustada, al movimiento y el marcador que desde el principio se están tomando. Cualquier otro crecimiento, obtenido de un tipo distinto de marcador se define de forma análoga.*

5.3. Eliminación de apéndices y crecimientos.

Eliminar apéndices y crecimientos, es la forma adecuada de obtener, la frontera definitiva X^n de una imagen binaria, obtenida del cifrador Isis.

5.3.1. Eliminacion de apéndices.

Si n_i, es el número de filas en el i-esimo bloque, L y C, son el conjunto de apéndices y crecimientos respectivamente, obtenidos por un movimiento. Entonces para cada $a_{ik} \in X'$, hacemos :

$$A(h_t) = \begin{cases} p(a_{ik}) & \text{si} \quad a_{ik} \notin L \cup C \\ p(a_{ik}) + 1 & \text{si} \quad a_{ik} \in L \end{cases}$$

Nota 13. *Si la posición de $h_t \in H$, genera un puente en el lugar a_{tj_0}, h_{t-1} está en el lugar, a_{t-1j_1} y h_{t+1}, en el lugar a_{t+1j_2}, $j_0 = Max\{j_1, j_2\}$, entonces $\{a_{t\pm1j} : Min\{j_1, j_2\} \leq j \leq j_0\}$, son apéndices.*

5.3.2. Podamiento (eliminación de crecimientos).

Si $h_t \in H$, es un crecimiento, entonces $A(h_t) = p(h_t) + 1$, siempre que $p(h_t) + 1$ no genere un apéndice, además si X'', es la frontera obtenida de X' por la eliminación de un crecimiento, entonces $Sop(X'') \leq Sop(X')$.

Una véz aplicado el algoritmo, es posible recuperar algunos crecimientos, que permitirían mayor asociación de la imagen obtenida por el algoritmo, con la imagen original.

5.4. Decodificación de los números perfectos.

A continuación se muestra el procedimiento, teniendo como entrada los primeros 5245 dígitos de los números perfectos.[11]

La permutación inicial produce que tengamos que escribir los números en la forma indicada por la tabla 1.

[11]Lo cual implica tratar las cifras hasta el perfecto 26.

6	2	4	8	3	8	...
.	8	9	1	3	5	...
.	.	6	2	5	8	...
.	.	.	8	5	9	...
.	.	.	.	0	8	...
.	.	.	.	3	6	...
.	.	.	.	3	9	...
.	.	.	.	6	0	...
.	.	.	.	.	5	...
.	.	.	.	.	6	...
.	.	.	.	.	⋮	⋮

Cuadro 1: Permutación inicial de los dígitos de los números perfectos.

En el segundo paso identificamos los tamaños de cada uno de los bloques en los que se dividen estas cifras. Estos tamaños son :

$26 \times 106, 14 \times 106, 14 \times 106$ y 66×13.

Una vez reconocidos los tamaños de los bloques, los cuales notamos $B_i, 1 \leq i \leq 4$, aplicamos el algoritmo de secuenciación y se encuentra la permutación final de los dígitos permutando adecuadamente los 4 bloques obtenidos. Este proceso produce como resultado el mapa de bits de la figura 5, conteniendo todas las imágenes binarias que le darán a la postre el nombre TIEMPO.

Nota 14. *Los bloques obtenidos son el código, de tal forma que distintas permutaciones de ellos, producirán, distintos mapas de bits conteniendo imágenes pertenecientes a bases de datos de imágenes que corresponden a un texto claro especifico. Además tal arreglo puede ser considerado como un texto cifrado, donde el desciframiento solo puede lograrse, con la lista de marcadores requeridos, para obtener cada imagen contenida en él.*

Algunos de los ideogramas mas utilizados para medir el paso del tiempo estarán contenidos en el mapa de bits. Por ejemplo tenemos el ideograma para indicar una época, que sabemos son:

antes de Cristo(A.C) y después de Cristo(D.C). El mapa contiene por tanto la imagen de cristo y algunas de las imagenes más conocidas de su vida a manera de autenticación del personaje.

Los ideogramas más usados para indicar un mes dado del año son los signos

del zodiaco, asi que en el mapa de bits encontraremos también, los ideogramas representando las constelaciones de capricornio,cáncer, etc.

Los ideogramas mas reconocidos para determinar un tipo de año son los usados por los chinos, en este caso el mapa de bits contiene los simbolos del horóscopo chino, por ejemplo, rata, perro, dragón, liebre, etc.

Nota 15. *Los marcadores y las correspondientes direcciones*[12] *necesarias para obtener la figura 28, deben ser aplicadas en el siguiente orden:*

En el primer bloque, $(63,16)^{\uparrow}, (24,5)^{\uparrow}, (63,16)^{\uparrow}, (89,14)^{\downarrow}$.

En el segundo y tercer bloques, $(89,1)^{\uparrow}_{2}, (68,1)^{\uparrow}_{3}, (55,5)^{\uparrow}_{3}, (62,4)^{\rightarrow}_{3}$.

En el cuarto bloque, $(78,1)^{\uparrow}_{4}, (66,14)^{\uparrow}_{4}, (33,5)^{\leftarrow}_{4}$.

Se finaliza con $(30,14)^{\downarrow}_{2}, (28,16)^{\downarrow}_{1}, (12,2)^{\downarrow}_{1}$.

El sentido de la flecha, indica la forma como debe ser aplicado el movimiento, por ejemplo $\leftarrow$ *en este caso, indica que los pixeles a la izquierda del marcador*[13]*, son eliminados mientras que la* P *en el subíindice indica que en ese lugar hay un puente.*

Los marcadores y puentes, que permiten obtener la figura 7, son :

$\underset{\leftarrow}{}(104,6)_{2}$, $\leftarrow(81,14)_{1}$, $\underset{\leftarrow}{}(68,7)_{1}$, $\underset{\leftarrow}{}(62,10)_{1}$, $\underset{\leftarrow}{}(56,12)_{1}$, $\underset{\leftarrow}{}(57,13)_{1P}$, $(30,5)_{\downarrow 1}$, $\underset{\leftarrow}{}(24,10)_{1}$, $\overset{\leftarrow}{}(27,15)_{1}$, $(42,15)_{1P}$.

$(32,1)_{\downarrow 2}$, $(49,8)_{\downarrow 2}$.

$(6,1)^{\longmapsto}_{3}$, $(16,3)_{3P}$, $(17,3)_{3P}$, $(18,3)_{\mapsto 3}$, $(28,9)_{3P}$, $(65,9)_{3P}$, $(66,9)_{3P}$.

$(64,8)_{4P}$, $(66,2)_{\downarrow 4}$, $(66,9)^{\mapsto}_{4}$, $(75,5)_{4P}$, $(77,5)_{4P}$, $(75,5)^{\mapsto}_{4}$.

A través del apropiado uso de los marcadores este tipo de información es redundante y de distinto tamaño, siempre se obtendrá la misma información[14]. En las paginas 23 hasta 41 se muestran algunas de las imágenes mas relevantes contenidas en el mapa de bits mostrado en la figura 5.

La figura 7, muestra el rostro de cristo, una vez aplicado el algoritmo comparado

[12]La posición original del arreglo, es rotada 180 grados y los bloques notados con subíndices, se cuentan de izquierda a derecha con las filas contadas desde abajo hacia arriba.

[13]En este caso el movimiento se realiza vertical hacia abajo.

[14]Esto no significa que necesariamente se repitan la mismas imágenes o ideogramas.

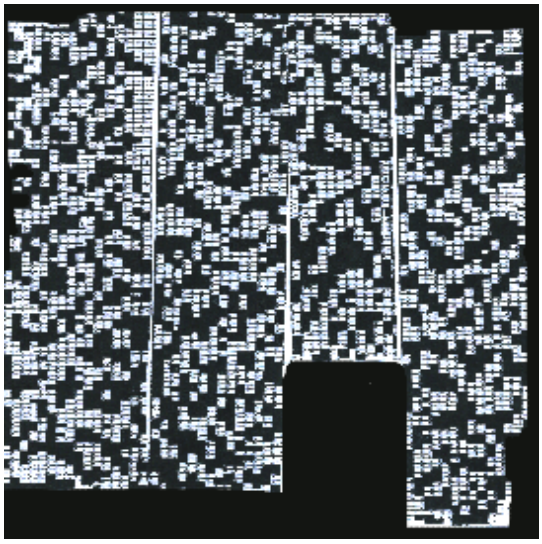

Figura 5: Mapa de bits TIEMPO obtenido de la permutación y secuenciación de las cifras de los números perfectos.

Figura 6: Rostros de Cristo en la base de datos.

con uno extraido de base de datos, sin embargo el código, autentica su identidad obteniendo otra imagen por rotación.

Otra forma de autenticación de una imagen en este contexto, se logra estableciendo que aquellos elementos de un texto representados en una base de datos de imágenes, forman parte del código decriptado, como una sucesión de imágenes satisfaciendo $I_1 \supset I_2, \ldots, I_K$. Donde I_1 es la mayor de las imágenes proporcionadas por el código y las imágenes, $I_2, \ldots, I_K$, son representaciones de objetos de la base de datos dadas por el código, relacionados todos con I_1.

Nota 16. *Las imágenes obtenidas por el proceso de decripción, al estar traslapadas, no están reconociendo fronteras de elementos de la base de datos de imágenes, son siempre aproximaciones de tales elementos*

La figura 8 muestra imágenes de Judas en la base de datos y la figura 10 muestra la serpiente en la base de datos y en el código. La serpiente en el código es obtenida de la imagen de Judas en la figura 9, por lo que como ideograma la imagen de Judas se lee "Aquel que traiciona". Note que las representaciones conocidas de este personaje, también utilizan elementos para identificarlo, por ejemplo la bolsa de dinero o el demonio en la obra de Giotto a la izquierda de la figura 8, todos de una forma u otra, mostrando el simbolismo de la traición.

Las figuras 12 a 17 representan, la ley de Herodes como elemento de la base de datos y como imágenes binarias en el código, en este caso el personaje en el código es autenticado, conteniendo la imagen de Salome.

Las figuras 18 a 23 muestran algunos aspectos de la natividad, tanto en la base de datos como en el código.

Las figuras 24 al 30, muestran las imágenes que corresponden al horóscopo chino, tanto en la base de datos, como en el código.

Las figuras 31 a 38, muestran las imágenes del zodiaco en la base de datos y en el código.

Las figuras 39 a 44 muestran el tipo de información que puede obtenerse, al decriptar usando el mismo procedimiento, tanto los primos de Mersenne como las cifras decimales de raíz de tres (un millón de ellas). En el caso de los primos de Mersenne todos contienen el mismo tipo de información por lo que se puede decir que la palabra CIRCO se representa a través de las imágenes contenidas en su código, o correspondiente mapa de bits. Mientras que los cuentos infantiles, son representados en el código que puede ser obtenido, a partir de las cifras de raíz de

tres con el procedimiento descrito en este articulo. Por lo que la estructura oculta en estos desarrollos binarios no serán más que conjuntos de imágenes cifradas.

6. Conclusiones.

La búsqueda de estructuras, visibles en las cifras de números irracionales, lleva a considerar tales números como textos cifrados. Esta consideración es valida tambień para cualquier sucesión de números naturales.

Los textos claros pueden leerse por medio de las imágenes que pueden obtenerse, desarrollando un procedimiento que consta de dos etapas, una de ellas es la secuenciación de las cifras, buscando la interrelación que existe entre ellas a través del problema de partición. La otra, consiste en determinar las permutaciones adecuadas que hacen de dicha secuenciación un mapa de bits. Siendo el objetivo primordial de todo el proceso determinar cual es la palabra o conjunto de palabras que identifican dicho mapa de bits, este consta de imágenes traslapadas dispuestas de tal forma, que puede hacerse una lectura de ellas como si fueran ideogramas. Estos ideogramas constan de representaciones de elementos que en la base de datos siempre permanenecen relacionados, de tal manera que una lectura no es más que la extracción de todas las imágenes contenidas en el más grande de los ideogramas.

Figura 7: Rostro de Cristo en el mapa de bits, proporcionado por el algoritmo a la derecha. A la izquierda, un rostro de Cristo tipico, extraido de la base de datos como mapa de bits monocromo. La diferencia entre una imagen obtenida por el algoritmo y una de la base de datos, consiste en que las imágenes de la base de datos no proveen de una autenticación mientras que las imágenes del código son autenticadas por otras imágenes obtenidas de ellas por una rotación.

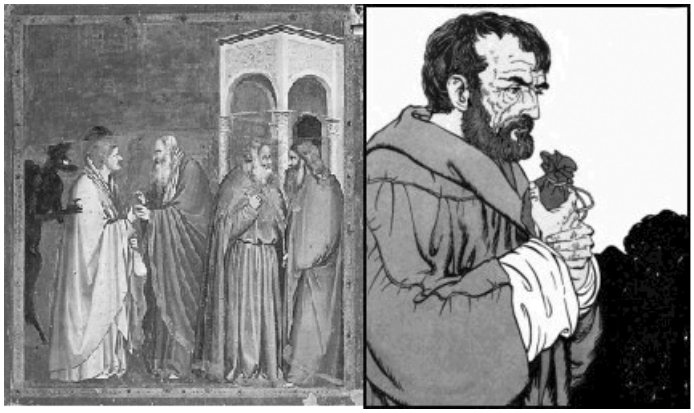

Figura 8: Judas en la base de datos.

Figura 9: Judas en el código.

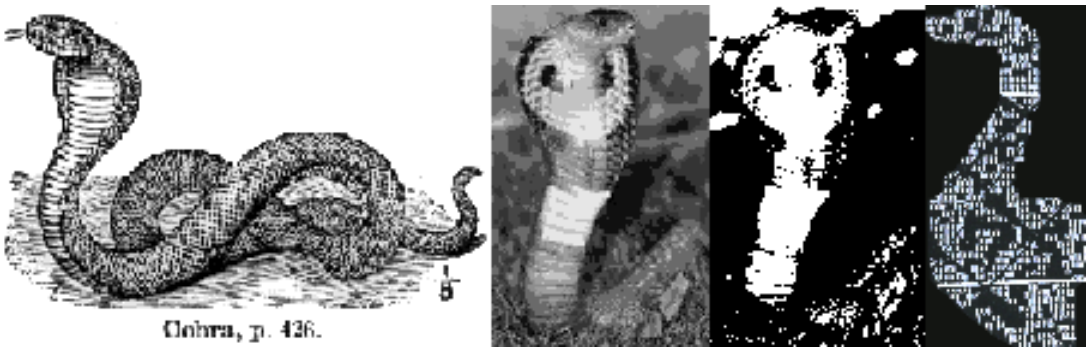

Figura 10: Serpientes en la base de datos y en el código (extremo derecho), obtenida de la imagen de Judas figura 9.

Figura 11: Ley de Herodes en la base de datos una obra de Duccio.

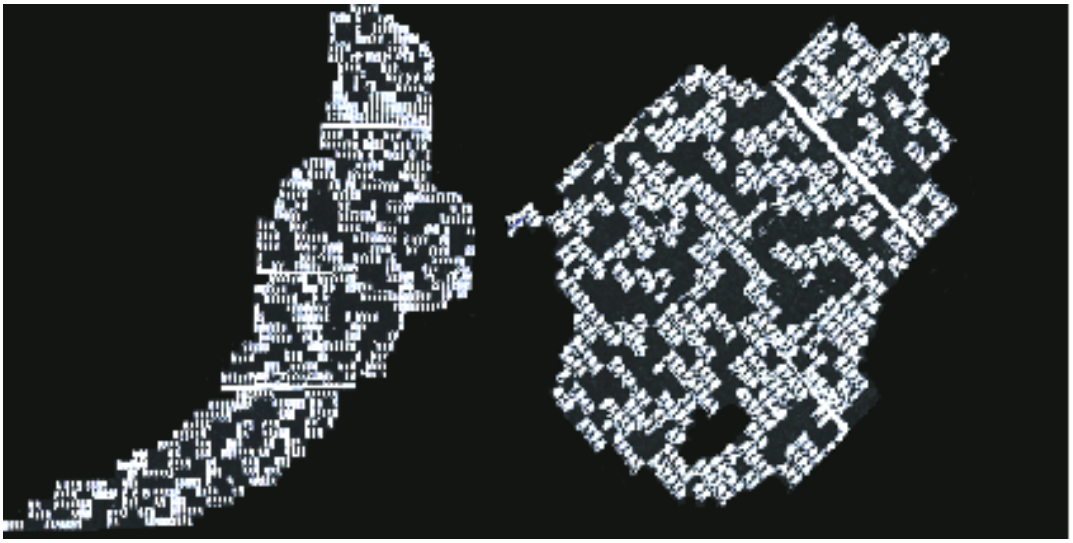

Figura 12: Todas las imágenes necesarias, para representar la ley de Herodes están contenidas en la espada a la izquierda. A la derecha el rostro de un niño

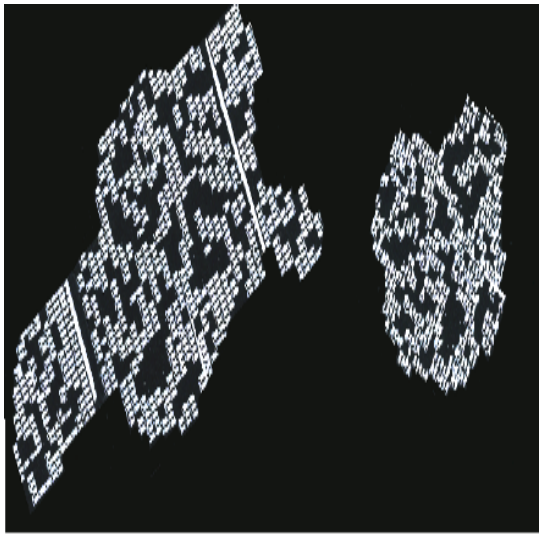

Figura 13: A la izquierda, la espada atraviesa el niño note que esta conjugación de imágenes produce la imagen de un cordero.

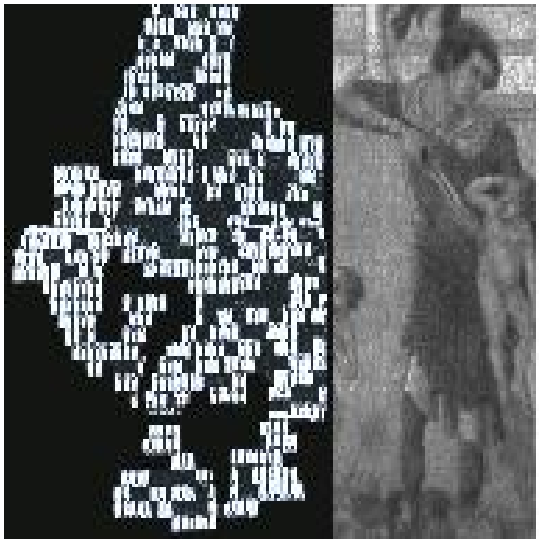

Figura 14: El genocidio en el código a la izquierda y en la base de datos a la derecha.

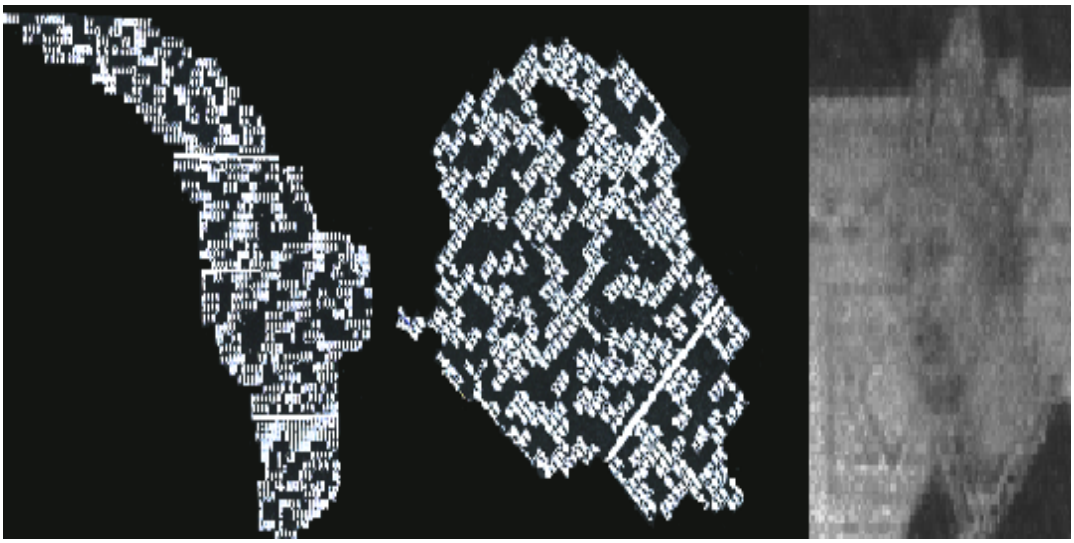

Figura 15: Rotación de la figura 13, representando un perfil de Herodes.

Figura 16: Ley de Herodes en la base de datos y el código.

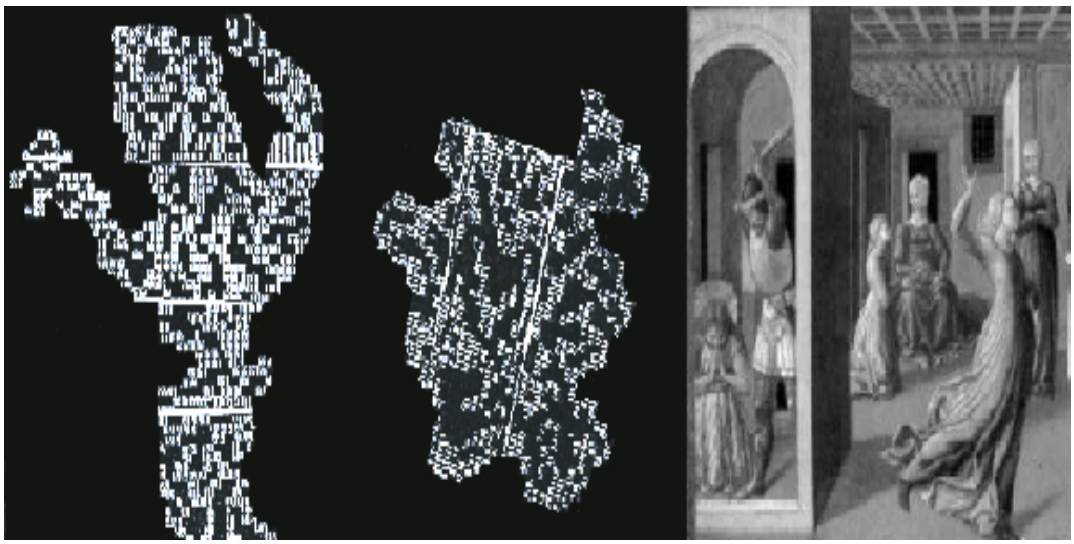

Figura 17: Salome en la base de datos y el código, la rotación 180 grados muestra en el código la cabeza de Juan Bautista.

Figura 18: La adoración de los magos(Giotto) como elemento de la base de datos.

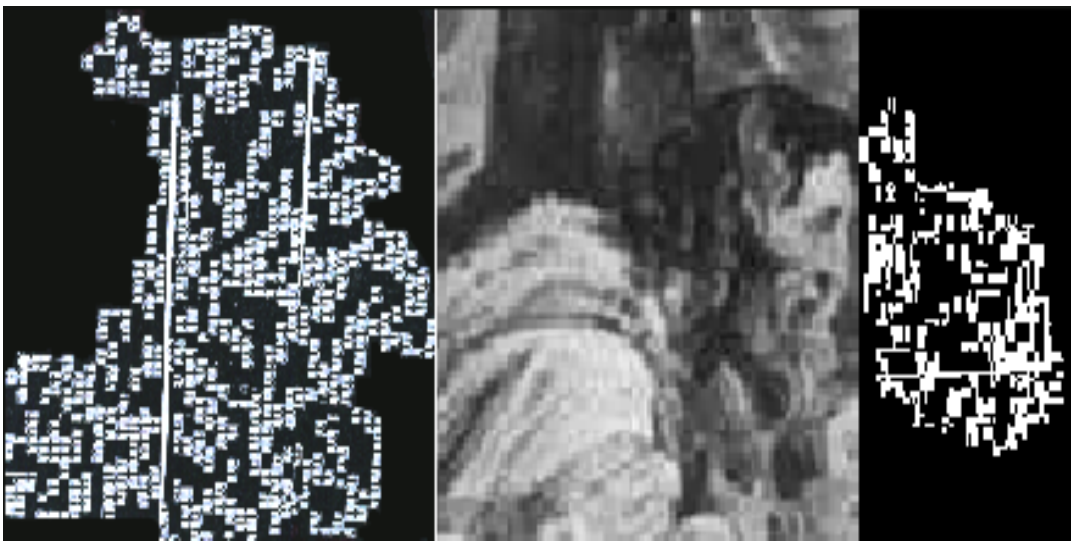

Figura 19: Detalle de la adoración de los magos de Giotto(centro) y representación con su respectiva autenticación en el código a los extremos.

Figura 20: Aplicación de la transformación de la distancia a la obra de Giotto, a un detalle (centro) y a la obra completa (extremo derecho), a la izquierda una representación de los magos en el código.

Figura 21: Reyes magos proporcionados por el código.

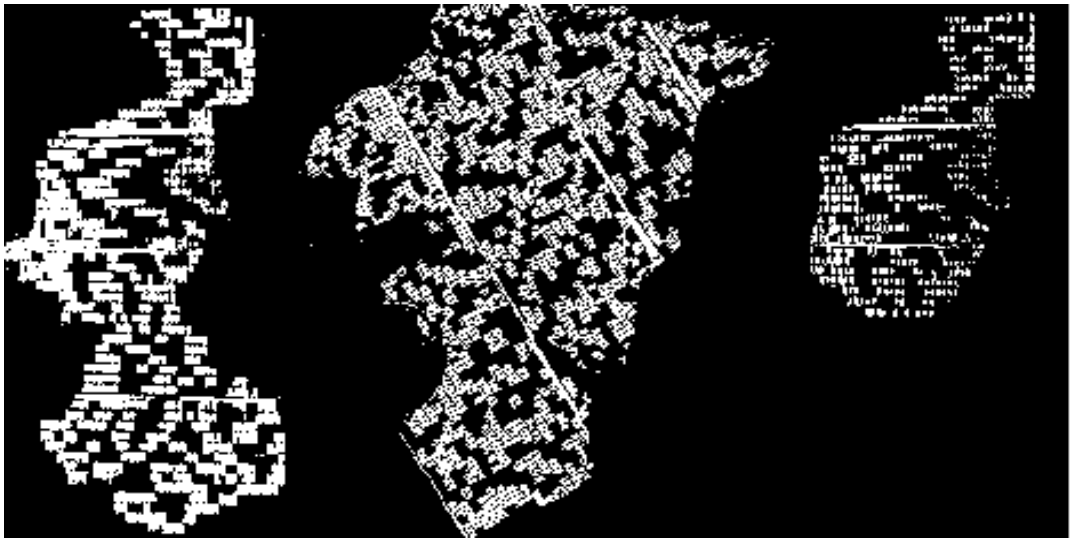

Figura 22: A la izquierda imagen del niño, al centro se muestra una rotación de la imagen izquierda de la figura 21, la cual representa otro mago y a la derecha un regalo como autenticación contenida en la imagen del niño. Todas ellas proporcionadas por el código.

Figura 23: La adoración de los magos(Duccio).

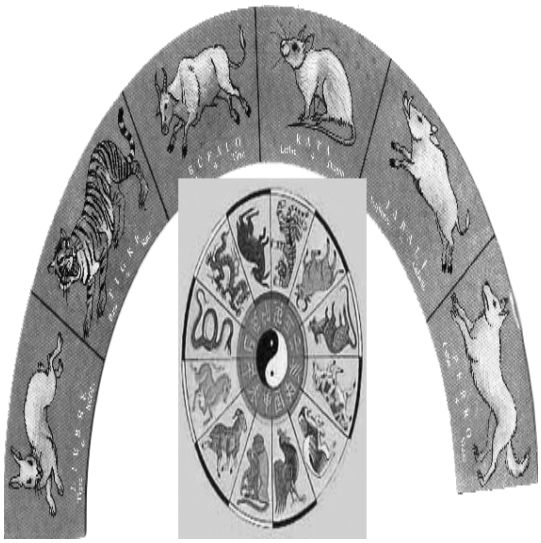

Figura 24: Horóscopo chino como base de datos.

Figura 25: Dragón y Conejo en el código.

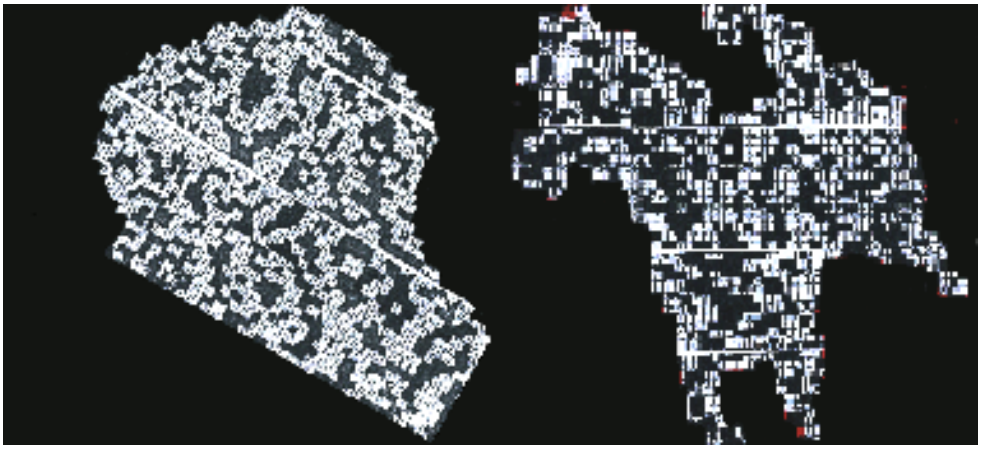

Figura 26: Mono y Caballo en el código note que la imagen Mono está contenida en la de Conejo.

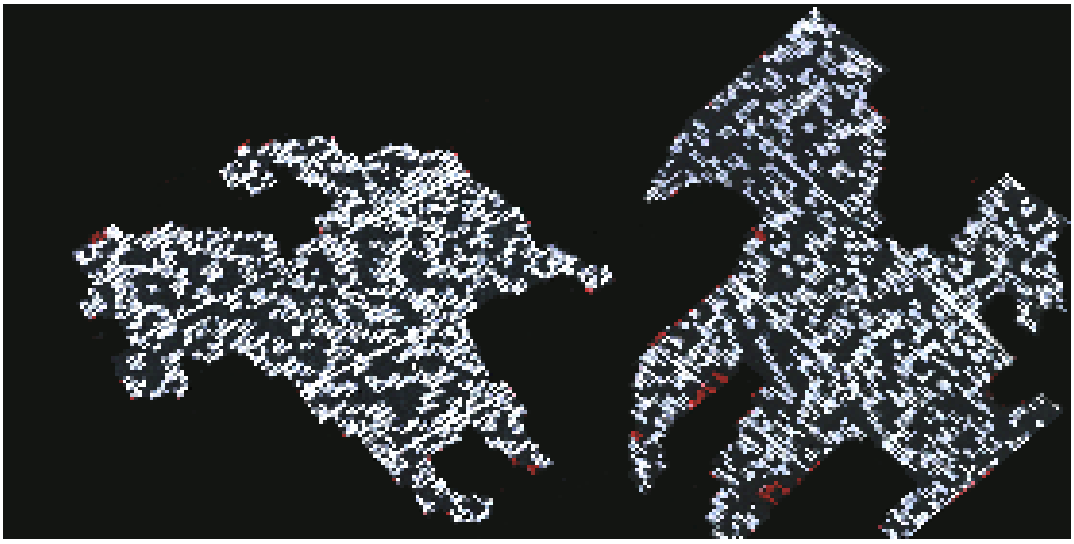

Figura 27: Perro obtenido de una rotación de Caballo en la figura(26) y Gallo en el código.

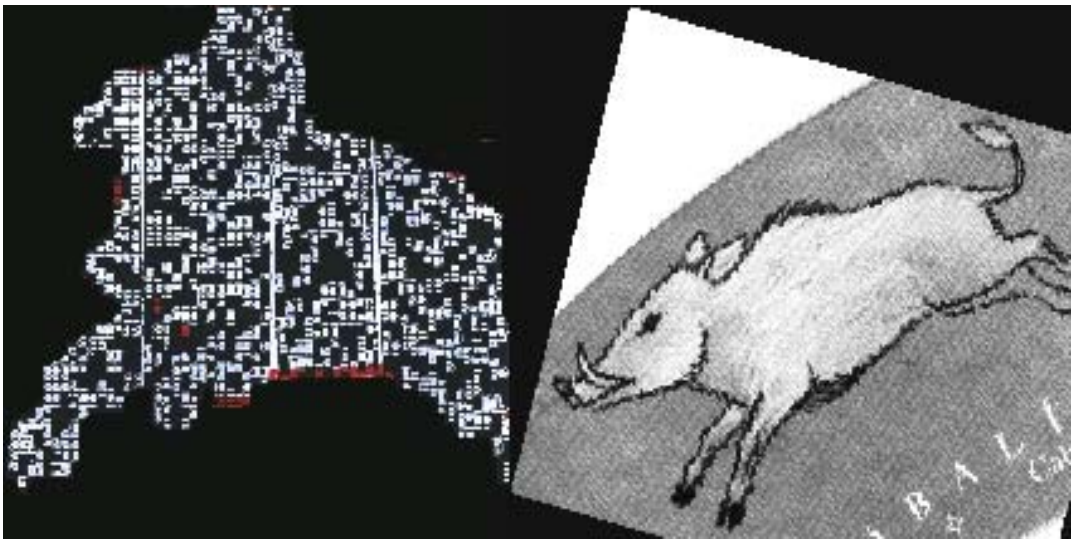

Figura 28: Cerdo en el código y en la base de datos.

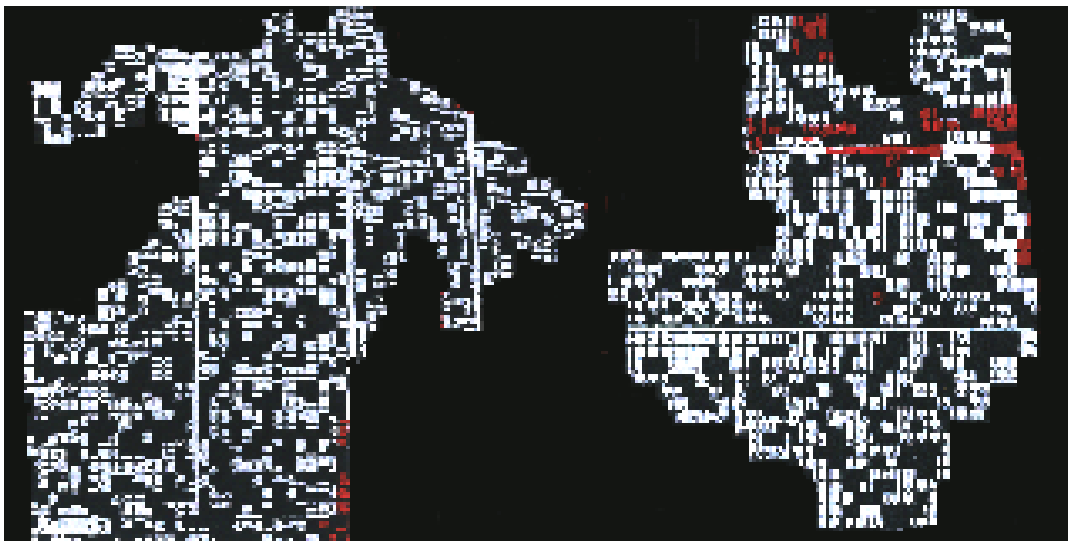

Figura 29: Cabra y Rata en el código.

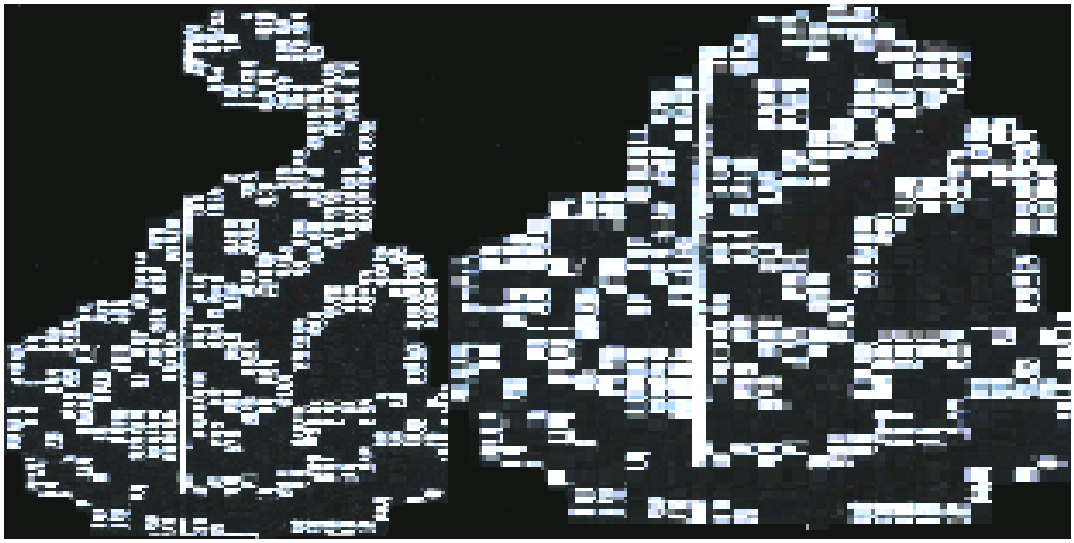

Figura 30: Serpiente y Mono, en esta figura se muestra como la imagen de Serpiente, contiene la de Mono.

Figura 31: Perro y Rata en el código.

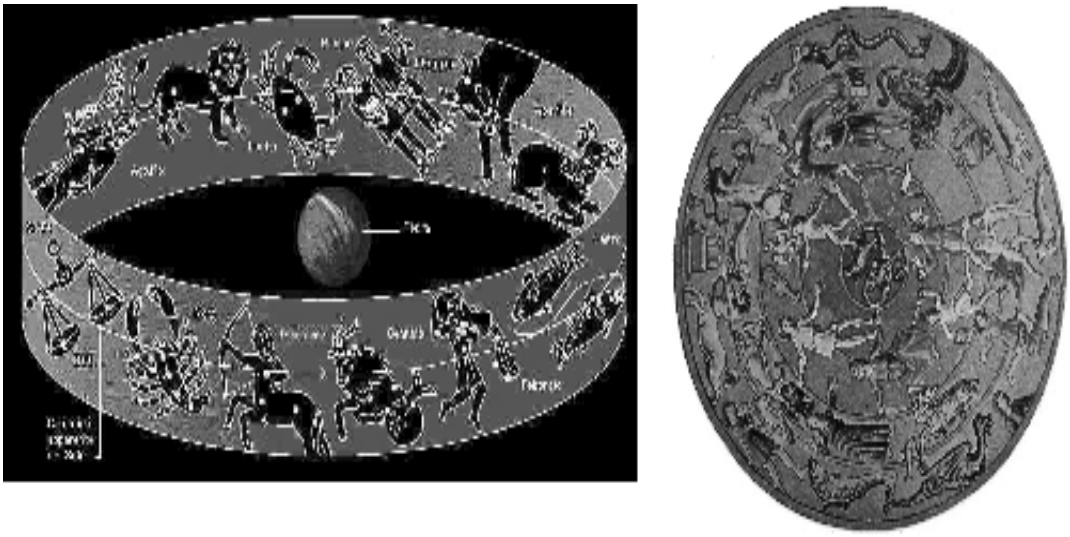

Figura 32: Zodiaco como base de datos.

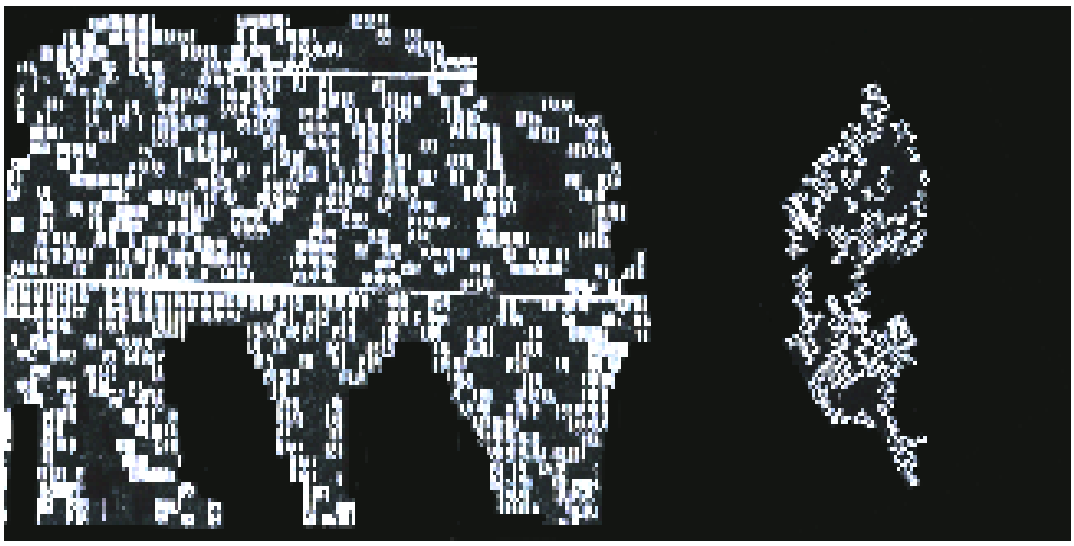

Figura 33: Tauro y Sagitario en el código.

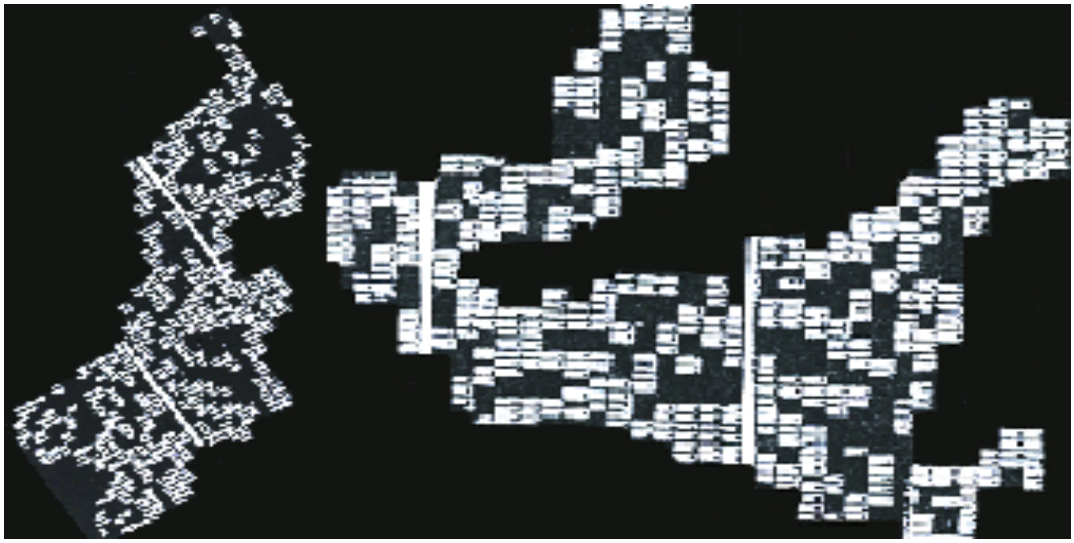

Figura 34: A la izquierda se muestran Capricornio y Sagitario en una sola imagen y a la derecha Escorpion en el código.

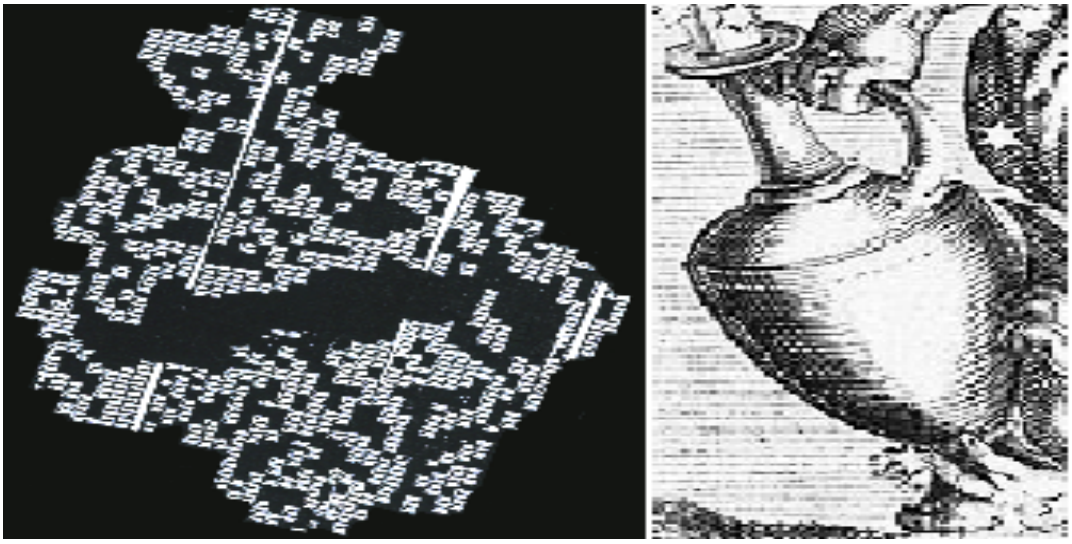

Figura 35: A la izquierda se muestran Como Cáncer y Leo(rotación) conforman Acuario en el código.

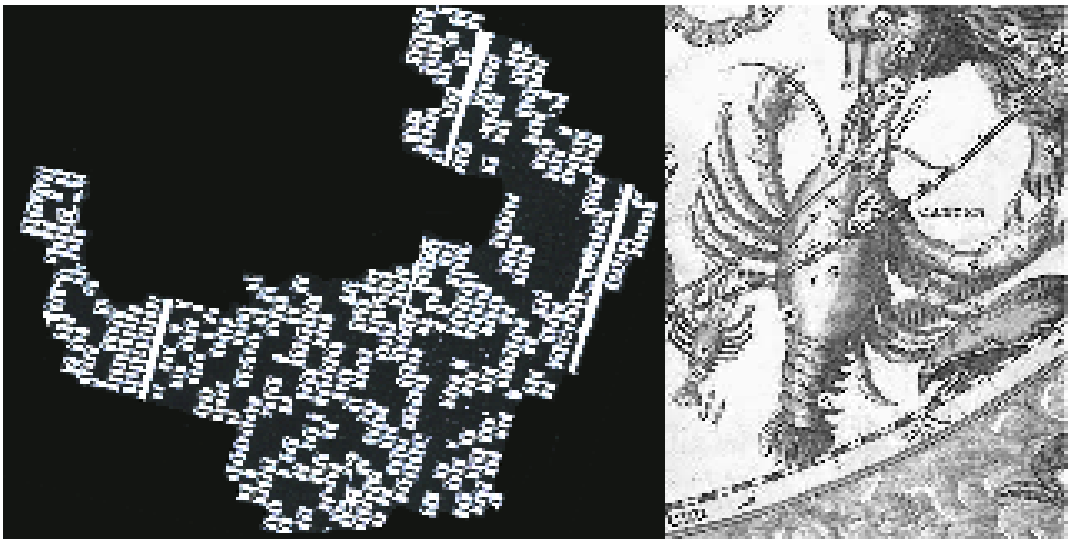

Figura 36: Cáncer en el código y en la base de datos.

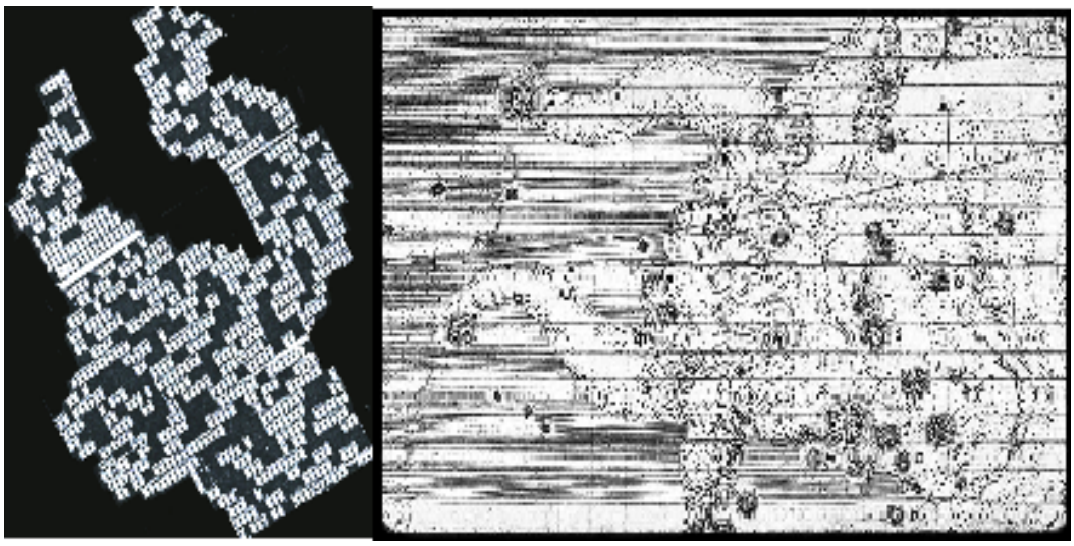

Figura 37: Tauro en el código y en la base de datos.

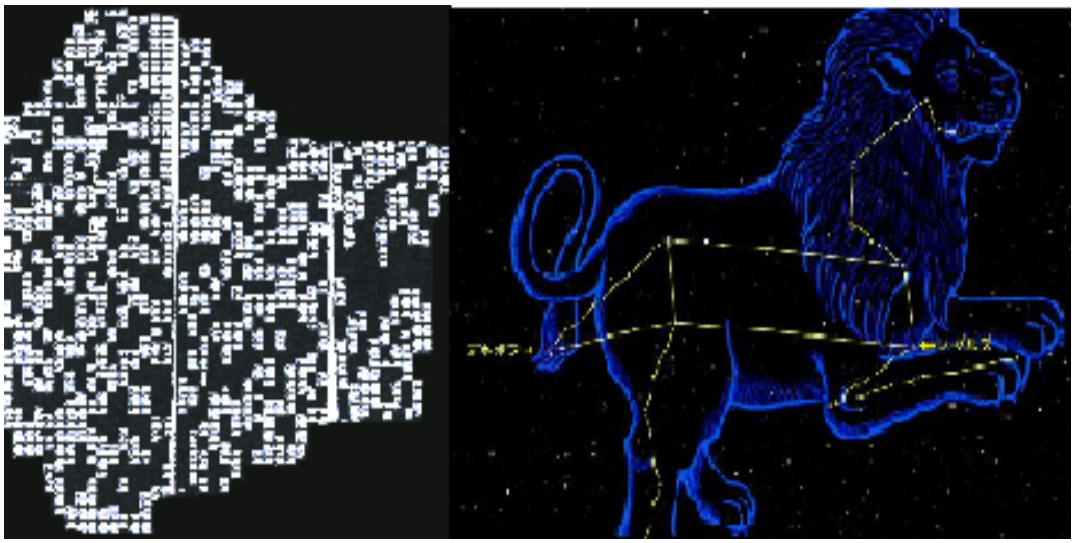

Figura 38: Leo en el código y en la base de datos, en este caso el código presenta la cabeza del León.

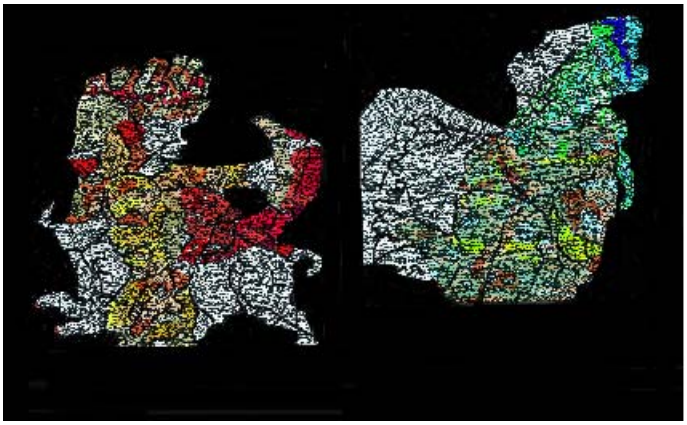

Figura 39: Ejemplo de imágenes obtenidas al decriptar con el mismo procedimiento el primo de Mersenne 34.

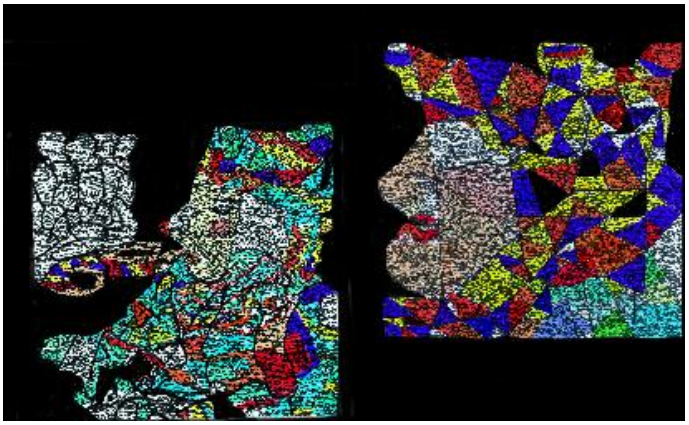

Figura 40: Ejemplo de imágenes obtenidas al decriptar con el mismo procedimiento el primo de Mersenne 34, todas representando números de circo.

Figura 41: Ejemplo de imágenes obtenidas al decriptar con el mismo procedimiento el primo de Mersenne 33, mostrando el mismo tipo de información decriptada en el Mersenne 34.

Figura 42: Ejemplo de imágenes obtenidas al decriptar con el mismo procedimiento el primo de Mersenne 33, mostrando el mismo tipo de información decriptada en el Mersenne 34.

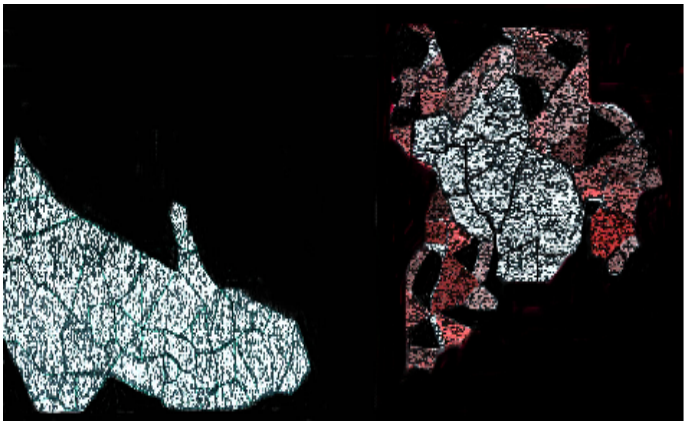

Figura 43: Ejemplo de imágenes obtenidas al decriptar con el mismo procedimiento las cifras de raíz de tres, en este caso el tema se relaciona con cuentos infantiles, aquí se muestra la zapatilla del cuento de la Cenicienta y la bruja de Blancanieves.

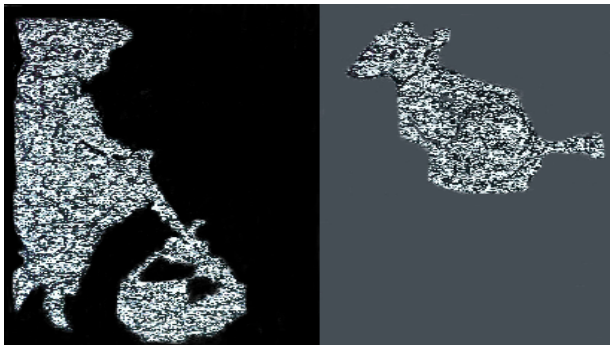

Figura 44: La zapatilla contiene las imágenes en esta figura, en este caso la escena de la medición de la zapatilla de cristal, la calabaza y los ratones de la historia.

Bibliografía

[1] Stinson Douglas R, Cryptography, Theory and Practice, CRC Press, Inc, 1995.

[2] Stallings William, Cryptography And Networks Security, second edition, Prentice Hall, 1999.

[3] Seberry Jennifer and Pieprzyk Josef, Cryptography An Introduction To Computer Security, Prentice Hall, 1989.

[4] Ritter Gerhard X, Wilson Joseph, Handbook of Computer Vision, Algorithms in Image Algebra, CRC Press, 2001.

[5] Apostol Tom M, Introduction to Analytic Number Theory, Springer, 1998.

[6] Moreno Agustín, Simetria De Los Números Perfectos, Encuentro de Geometria, Universidad Pedagogica Nacional,211-237, 2002.

[7] N Koblitz, A Course in Number Theory and Cryptography, Springer-Verlag, 1987.

[8] D.S Johnson, The NP-Completeness column: an ongoing guide. Journal of Algorithms,**9**(1988), 426-444

[9] WWW.GIMPS.COM.