Eng. & Tech. Journal ,Vol.27, No.16,2009

# Using Fractals in Information Hiding

**Dr. Nadia M. Al-Saidie*** **Thanaa A. Kadhim***

Received on: 29/4/2008
Accepted on:2/7/2009

**Abstract**

Fractals are generated by simple, recursive calculations. Encryption and decryption algorithms that take advantage of fractals are developed. This paper presents a new approach for information hiding using iterated function system (IFS) this approach exploits the main feature of fractals (generated by IFS). So that any individual that happens to find the transformed message, will not be able to understand it; without the correct method that will reverse the transformation, usually through some knowledge of key agreement, with the original encrypted. Also to make the encoding more difficult to introducers we use steganographic methods to hide the attractor image in another colored image 256X256 pixels size.

**Keywords:** fractals; IFS; cryptography; steganography ;collage theorem image decoding.

## إستخدام الكسوريات في إخفاء البيانات

**الخلاصة**

نطرح هنا طريقة جديدة لإخفاء البيانات , بالاستفادة من خاصية تشابه اجزاء في نماذج الهندسة الكسرية . او ما اصبح يعرف بالفراكتلات , منذ الدراسة التي قام بها ماندلبورت للهندسة الكسرية , حيث طرحنا الحقائق الرياضية لتشابه النماذج الكسورية الفراكتل . ثم اوجه الشبه والاختلاف بين التشفير واخفاء البيانات "Steganography" وبعض الانواع والطرق المستخدمة قدديما وحديثا , كاخفاء رسالة داخل صورة , بتغيير بسيط بقيم الالوان في البيان بحيث لا تسترعي انتباه المتلقي كونه الطرف الثالث بين المرسل والمرسل اليه المعلومة المخفية , وباستخدام نظم الدوال المتكررة "IFS" لاخفاء نص كرسالة يراد ايصالها الى المستلم كصورة خفية داخل صورة ملونة استخدمنا بذلك صورة ملونة , وحاولنا استخدام طريقة بارنسلي في نظرية الكولاج لاسترجاع حروف الرسالة من بين معلومات الصورة المستلمة . الهدف الرئيسي لهذه الرسالة اقتراح طريقة جديدة لتشفير رسالة مهما كان طولها باستخدام نظم الدوال المتكررة التي تولد النماذج المتشابهة من الفراكتلات والتي وفرت لنا ميزة مهمة جدا هو ان طول الرسالة لايشكل لنا عائقا في حشر الرسالة داخل الصورة الملونة مهما كان طولها , وذلك لان النتيجة ستكون صورة واحدة ستحشر داخل صورة ملونة.

## Introduction

Fractals where first described in the 1970's by IBM mathematician Beniot Mandelbrot. He found traditional geometry to be incomplete It had no formal representation of the appearance of a cloud. The new geometry that he developed could do all this. It was a description of the beautiful yet irregular and fragmented patterns around us.

* Applied science Department, University of Technology / Baghdad



https://doi.org/10.30684/etj.27.16.15
2412-0758/University of Technology-Iraq, Baghdad, Iraq

The term ‘fractal’, was coined by Mandelbrot from the Latin fractus, an adjective for the irregular and the fragmented. Essentially, they replicate themselves by fragmentation. [7].

Looking at a fractal pattern to see a form; then closely looking at a particular region of the pattern, to see the same form all over again, but only much smaller this time.

And so on it goes, from the largest scales to the smallest. So the fractals are the repetition of the same structural form [3].

Iterated Function Systems (IFSs) were born in mid eighties, as applications of the theory of discrete dynamical systems and as useful tools to build fractals and other similar sets. Some possible applications of IFSs can be found in image processing theory [18], in the theory of stochastic growth models and in the theory of random dynamical systems. In particular they come to a great popularity when they were been proposed as the ‘definitive’ method of images compression [8].

The idea behind the IFSs is the following: Suppose one wants to approximate an object f, i.e., a function, which is a point in some complete metric space (R, d). The aim is to construct a contractive operator W: R → R, with a unique fixed point x, i.e. W (x) = x, in such a way that d(x, W(x)) is minimum. In our case R will be the space of distance functions on a compact set [a, b] and d is the sup-norm distance.

**Definition of a fractal:**

Many definitions to the fractals as many mathematicians studied it, we will mention two of them:

**Benoit Mandelbrot** refers to the word "fractals" as objects that possess self-similarity. For the rest of this paper, we will adopt this as our definition of a fractal.figure 1 shows Mandelbrot's set.

He said:' I coined the word fractal in 1975 from the Latin ***fractus,*** which describes a broken stone-broken up and irregular. Fractals are geometrical shapes that, contrary to those of Euclid, are not regular at all. First, they are irregular all over. Secondly, they have the same degree of irregularity on all scales. A fractal object looks the same when examined from far away or nearby-it is self-similar. As you approach it, however, you find that small pieces of the whole, which seemed from a distance to be formless blobs, become well - defined objects whose shape is roughly that of the previously examined whole [1].

Nature provides many examples of fractals, for example, ferns, cauliflowers and broccoli, and many other plants, because each branch and twig are very like the whole. The rules governing growth ensure that small-scale features became translated into large-scale ones.

Figure 2 shows some of nature's fractals.

**Barnsley**

In his book" fractals every where" [3], started his introduction by:

"Fractal geometry will make you sees everything differently. You risk the loss of your childhood vision of clouds, forests, flowers, rocks, mountains, torrents of water and much else besides. Never again will your interpretation of these things be quite the same".

Fractals used as a model for studying complex natural phenomena. If you look closer and closer at a leaf of a tree,

you will eventually arrive at the cells that constitute the leaf. This property of self-similarity that natural objects possess, which maintained, as long as, we do not look too close at them. However, with fractals, there is no limit as to how far we can look before this property disappears.

Fractal geometry concerned with the description, classification, structure, analysis and observation of subsets of metric spaces (X, d). The metric spaces are usually, but not always, of simple geometrical characters. The subsets are typically geometrically "complicated". Fractal is a subset of a space. Whereas the space is simple, according to, the fractal subset may be complicated. There are number of general properties of subsets of metric spaces, which occur repeatedly. They are very basic, and form part of vocabulary for describing fractal sets and other subsets of metric spaces. Some of these properties, such as openness and closeness in a space denoted by X [3].

**The space of fractals**: **[3]**

The principle goal is to establish the construction of the element of the space of fractal denoted by H(X) the element of this space is the nonempty compact subsets of a complete metric space (X, d). There is a kind of interpolate between the space X and the space H(X), since a point A in H(X)corresponds to a compact subset in X.

In order to show that H(X) is a metric space we must define a distance function between any two compact sets called Hausdorff distance defined as d(A,B)= max {d(A, B), d(B,A) } which form with H(X)a metric space [3].

**Affine transformation [3]:**

Affine transformations in R are transformations of the form

$f(x) = a x + b;$

Where a, b are real constants.

Moreover, for the Euclidean plane the affine transformation defined as follows:

A transformation w: $R^2 \rightarrow R^2$ of the form:

$$\mathbf{w(x_1, x_2)=(a\, x_1 + b\, x_2 + e,\ c\, x_1 + d x_2 + f),}$$

where a ,b ,c ,d ,e ,and f, are real numbers ,is called a (two–dimensional ) affine transformation .We will often use the following notations:

$$w\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = A\ x + b.$$

Here $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is a 2 x 2 real matrix and b is the column vector $\begin{pmatrix} e \\ f \end{pmatrix}$, which we do not distinguish from the coordinate pair (e, f) $R^2$.

Such transformations have important geometrical and algebraic properties. Which they are:

1. Maps parallelograms into parallelograms.
2. Can turn a figure over (not shape preserving)
3. Affine = linear followed by translation.

**Contraction Mappings [3]**

f: X→X is a transformation on the metric space (X, d). f is a contraction if $\exists s \in R$ with $0 \leq s < 1$ such that $d(f(x),f(y)) \leq s\ ;\ d(x,y)\ \forall\ x, y \in X$. Any such number s, is called a contractivity factor of f.

Figure 3 shows the contraction.

**Iterated Function Systems[3]:**

A (hyperbolic) iterated function system consists of a complete metric space (X, d) together with a finite set of contraction mappings:

$w_n$: X→X, with respective contractivity factor $s_n$, for n =1,2,...N.

The abbreviation "IFS" is used for "iterated function systems".

The notation for the IFS just announced is {X; $w_i$ : i = 1,2,...n} and its contractivity factor is s = max {$s_i$ :i = 1,2,...n}.

**Generating the fractal [3]:**

How can we generate these fractals? When we repeat this scheme repeatedly, a global pattern will emerge. This global process will often have the self-similarity property that we were talking about. This technique is very similar to an iterated function system. Both of them are dynamical systems.

An Iterated Function Systems is a set of contraction mappings

W= {$w_1$, $w_2$...$w_n$} acting on a space X.

Associated with this set of mappings W, there existes a set of probabilities:

P= {$P_1$, $P_2$... $P_n$}. These probabilities are used to generate a random walk in the space X. If we start with any point in X and apply these maps iteratively, we will come arbitrarily close to a set of points A in X called the attractor of the IFS. These attractors are very often fractal. (For the most part, we will assume attractors are fractal sets, and thus, use the words interchangeably). This forms the basis for creating an algorithm that will approximate the attractor of IFS.

Increasing the number of times we apply the maps will give a more accurate picture of what the attractor looks like. The algorithms to generating the fractals are:

**1. The Deterministic Algorithm [3]**

Let {X; $w_1$, $w_2$...$w_n$) be an IFS. Choose any compact set $B_o \subset R^2$. Then compute successively $B_{n+1}$ = union ($w_i$ ($B_n$)) for i = 1... , n

This sequence {$B_n$} will converge to the attractor of the IFS.

**2.The Random Iteration Algorithm[3]**

Let {X ;$w_1$ ,$w_2$ ,...$w_N$} be an IFS, where the probability $P_i > 0$ has been assigned to $w_i$ for i = 1,2, ....n , where $\Sigma P_i = 1$. Choose $x_o \in X$ and then choose recursively and independently:

$X_n \in \{w_1(x_{n-1}), w_2(x_{n-1}), ...., w_N(x_{n-1})\}$ for n = 1,2,3... Where the probability of the event $x_n = w_i (x_{n-1})$ is $P_i$. Thus construct a sequence:

{$x_n$ : n = 0,1,2....} X.

This sequence of points will come arbitrarily close to every point in the attractor of the IFS.

**The algorithm to the random iteration:**

(i) Initialize: x =0, y =0.

(ii) For n=1 to 2500, do steps (iii)-(vii) .

(iii) Choose k to be one of the numbers 1, 2 , . . ., m, with probability P,

(iv) Apply the transformation W, to the point (x, y) to obtain

(v) Set (x, y) equal to the new point: x=z, y=y.

(vi) If n > 10, plot (x, y).

(vii) Loop.

**Examples of Fractals [3]:**

Fractals are geometric figures based on recursive expansion of a basic geometric figure.

Approximated fractals are easily found in nature. These objects display self-similar structure over an extended, but finite, scale range. Trees and ferns are fractal in nature and can be modeled on a computer by using a recursive algorithm. This recursive nature is obvious in these examples .A branch from a tree or a frond from a fern is a miniature replica of the whole:

not identical, but similar in nature.

**Example1: The fern[3]:**

The affine transformations are associated with, the simple fern fractal are:

$$w_1 = \begin{pmatrix} 0 & 0 \\ 0 & .16 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ .1 \end{pmatrix}$$

$$w_2 = \begin{pmatrix} .2 & -.26 \\ .23 & .22 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ .16 \end{pmatrix}.$$

$$w_3 = \begin{pmatrix} -.15 & .28 \\ .26 & .24 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ .44 \end{pmatrix}.$$

$$w_4 = \begin{pmatrix} .75 & .04 \\ -.04 & .85 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1.6 \end{pmatrix}.$$

The figure no.4 shows the iterations of the fern.

**Extracting the IFS by The collage theorem[10]:**

To understand how to find the IFS from the images, it is necessary to introduce the inverse problem or what Barnsley used to call the inverse problem by the collage theorem .The collage theorem provided the first stepping-stone toward solving the inverse problem. It states that a lazy tiling of an object out of smaller self-replicas yields a significantly more accurate IFS representation.

Although the collage theorem relaxed the accuracy of self-similar tiling needed for reasonable fractal modeling, the harder problem of forming an object out of smaller self-replicas remained unsolved. Automatic fractal modeling of objects based on heuristic search techniques of the IFS parameter space have yielded promising results, but have not yet become practical [10].

**Information hiding:**

Information hiding represents a class of processes used to embed data into various forms of media such as image, audio, or text. The embedded data should be invisible to human observer. Steganography and digital watermarking are two areas referred to as "information hiding" [15].

The word steganography comes from the Greek word "stegein", "grajein" which literally means "covered writing ". It conceals a message where that is the object of the communication. For example, sending an image hidden in other image.

In this paper we intended for a technical introduction to steganography for those unfamiliar with the field. Supported with a historical context for steganography; the emphasis is on digital applications.

**Cryptography:**

***Cryptography is the study of methods of sending information in disguised form so that only the intended recipients can remove the disguise and read the message [13].***

**Steganography:**

*Steganography is:" Hiding a secret message within a larger one in such a way that others can not discern the presence or contents of the hidden message". The purpose of steganography is to covert communication "to hide the existence of a message from a third party".* [16, 15]

**Application to Steganography:**

As an increasing amount of data is stored on computers and transmitted over networks, it is not surprising that Steganography has entered the digital age. On computers and networks, Steganography applications allow for someone to hide any type of binary file in any other binary file, although image and audio files are today's most common carriers. Steganography provides some very useful and

commercially important functions in the digital world, most notably digital watermarking [14].

**History of steganography:**

Steganography is a very old method of passing messages in secret. This method of message cloaking goes back to the time of the ancient Greeks. The historian Herodotus wrote about how an agent wrote a message warning of an invasion on the wood part of a wax tablet. Although the term steganography was only coined at the end of the 15th century, the use of steganography dates back several millennia. In ancient times, messages were hidden on the back of wax writing tables, since messages were normally inscribed in the wax and not the wood, the tablet appeared blank to a common observer.

Other use was by writing on the stomachs of rabbits, or tattooed on the scalp of slaves. Invisible ink has been in use for centuries-for fun by children and students and for serious espionage by spies and terrorists. Microdots and microfilm, a staple of war and spy movies, came about after the invention of photography [14].

There is also the story of a messenger during the Persian Wars who shaved his head and had a message tattooed on it. He waited until his hair grew back to make his journey. When he arrived at his destination, he shaved his head to reveal the message [13, 16].

During WWII spies on both sides used “invisible” inks. These inks were fluids such as milk, fruit juice, or urine that would darken when heated. They also sent messages with very small punctures above characters in a document that formed a message when combined [13].

**Cryptography and Steganography [13]:**

Many people lump ***steganography*** with ***cryptography***, and while they are in many cases means to the same ends ***"not letting unauthorized persons or the third party view data*** ", they are not the same thing. Although, they are often related processes and first encrypting a message then using a stego -tool to hide it is more effective in hiding a secret message than either method by itself.

Steganography differs from cryptography, the art of secret writing, which is intended to make a message unreadable by a third party but does not hide the existence of the secret communication. Although steganography is separate and distinct from cryptography, there are many analogies between the two, and some authors categorize steganography as a form of cryptography since hidden communication is a form of secret writing.

**Some types of steganography[7]:**

**1) Technical steganography:**

This type uses scientific methods, to hide a message such as, the use of invisible ink, or microdots, and other size- reduction methods.

**2) Linguistic steganography:**

This type hides the message in the carrier in some non_obvious ways and is further categorized as Semagrams or open codes.

**3) Semagrams:**

Semagrams hide information by the use of symbols or signs.

**4) Covered or concealment:**

Ciphers hide a message openly in the carrier medium so that it can be recovered by anyone who knows the secret for how it was concealed. A grille cipher employs a template that is

used to cover the carrier message. The words that appear in the openings of the template are the hidden message. A null cipher hides the message according to some prearranged set of rules, such as "read every fifth word" or "look at the third character in every word". The reader will learn nothing by looking at the word spacing or misspellings in the message; the zeroes and ones are encoded by the very choice of the words. The message received looks like the spam that most of us receive every day, which we ignore and discard.

**Example2:**

Historically, null ciphers are a way to hide a message in another without the use of a complicated algorithm. On the Internet, spam is a potential carrier medium for hidden messages. [15] One of the simplest null ciphers is shown in the classic examples below:

***"PRESIDENT'S EMBARGO RULING SHOULD HAVE IMMEDIATE NOTICE. GRAVE SITUATION AFFECTING INTERNATIONAL LAW. STATEMENT FORESHADOWS RUIN OF MANY NEUTRALS. YELLOW JOURNALS UNIFYING NATIONAL EXCITEMENT IMMENSELY."***

The German Embassy in Washington, DC, sent these messages in telegrams to their headquarters in Berlin during World War I [15].

By ***reading the first character of every word in the first message*** will yield the following hidden text:

***"PERSHING SAILS FROM N.Y. JUNE 1".***

Special tools or skills to hide messages in digital files using variances of a null cipher are not necessary. An image or text block can be hidden under another image in a PowerPoint file, for example. Messages can be hidden in the properties of a Word file. Messages can be hidden in comments in Web pages or in other formatting vagaries that are ignored by browsers. Text can be hidden as line art in a document by putting the text in the same color as the background and placing another drawing in the foreground. The recipient could retrieve the hidden text by changing its color. These are all decidedly low-tech mechanisms, but they can be very effective.

**Steganography's common Carriers:**

An image is a two-dimensional array of image points or pixels. In gray level images each pixel is described by one number corresponding to its brightness. In color images each pixel is described by three numbers corresponding to the brightness's of the three primary colors e.g. red, green, and blue. A typical image file has two parts, the header and the raster data. The header contains the magic number" identifying the format, the image dimensions, and other format-specific information that describes how the raster data relates to image points or pixels. ***The raster data is a sequence of numbers that contains specific information about colors and brightness's of image points*** [15].

These values represent the intensities of the three colors R(ed) G(reen) and B(lue), where a value for each of the three colors describes a pixel. Through varying the intensity of the RGB values, a finite set of colors spanning the full visible spectrum can be created. In an 8-bit gif image, there can be:

$2^8$ = 256 colors and in a 24-bit bitmap, there can be $2^{24}$ = 16777216 colors. Large images are most desirable for Steganography because they have the most space to hide data in. [15]

The cube in figure 5 is a common

mean of RGB. It's to represent a given

color by the relative intensity of its three components colors-red, green, and blue-each with their own axis. The absence of all colors yields black, shown as the intersection of the zero point at the three-color axes. The mixture of 100 percent red, 100 percent blue, and the absence of green form magenta; cyan is 100 percent green and 100 percent blue without any red; and 100 percent green and 100 percent red with no blue combine to form yellow. White is the presence of all three colors. Each RGB component is specified by a single byte, so that the values, for each, color intensity can vary from 0-255[15].

**Embedding the data [10]:**

Embedding data, which is to be hidden, into an image requires two files. The first is the innocent-looking image that will hold the hidden information, called the *cover image*. The second file is the message or the information to be hidden. A message may be plain text, cipher text, or images, or anything that can be embedded in a bit stream. When combined, the cover image and the embedded message make a *stegoimage*. Gray-scale images are preferred because the shades change very gradually from byte to byte, and the less the value changes between palette entries, the better they can hide information. Obviously, an image with large areas of solid colors is a poor choice, as variances created from the embedded message will be noticeable in the solid areas [10].

**Preliminaries and terminology [11]**

***Steganography terminology***

A general steganography system is shown in figure6.

In this system there are numbers of known terminologies, which are illustrated as follows:

1) Embedded <data-type>: something to be hidden in something else.

M: refers to the set of possible messages, m refers to the secret message, the length of m is defined by l (m), and the bits forming m by $m_i$,

$1 \leq i \leq l(m)$.

2) cover <data-type> an input to the stego- system, in which the embedded will be hidden.

C: refers to the set of possible covers used in the embedding step, c refers to a cover, it can be represented by a sequence of numbers $c_i$ of length l( c) ( i.e., $1 \leq i \leq l(c)$) . In the case of digital images a sequence can be obtained by vectorizing the image.

$c_i = 0$ or 1 for binary images

$0 \leq c_i \leq 255$, for 256 quantized images.

The index of all cover- elements $c_i$ will use the symbol j, for such an index. If the index itself is indexed by some set, the notation $j_i$ will be used. When referring to the $j_i^{th}$ cover- element it means $c_{ji}$ .

3) stego <data- type > the out put from the stego system: something that has the embedded message hidden in it.

S: refers to the set of possible stego-objects, s refers to the stego object; the length of s is defined by l(s).

4) stego-key or simply key. Additional, secret data, that may be needed in the stego system. The same key (or related one) is usually needed to extract the embedded message again.

K: refers to the set of stego –key. k refers to a stego-key.

5) The process of hiding the embedded message is called embedding.

E: refers to the embedding process

6) Getting the embedded message

out of the stego-message again is called extracting.

D: refers to the extracting process.

7) The party from the embedded message is hidden is called stego-analyst.

8) The key has been generated often depending on one or more security parameters. The standard case, where the same key is used in embedding and extracting is called, symmetric.

9) An entity or person that embeds and extracts is called an embeddor and an extractor.

**The least significant bit[11]:**

The least significant bit term comes from the numeric significance of the bits in a byte. The high-order or most significant bit is the one with the highest arithmetic value (i.e., $2^7$=128), whereas the low-order or least significant bit is the one with the lowest arithmetic value (i.e., $2^0$=1).

**Algorithm 1[11]:**

Embedding process: least significant bit substitution

for i= 1, …l(c ) do

$s_i \leftarrow c_i$

end for

for i= 1, …l( c ) do

compute index $j_i$ , where to store i $^{th}$ message bit

$s_i \leftarrow c_i \leftrightarrow m_i$

end for

**Algorithm 2[11]:**

extracting process: least significant bit substitution

for i= 1, …l(m) do

compute index $j_i$ , where to store $i^{th}$ message bit

$m_i \leftarrow LSB(c_{ji})$

end for

This can, however, lead to a serious security problem: the cover will have different statistical properties than the second part, where no modifications have been made.

**Example3**

***the carrier of the stego information is a jpg format picture for " Kadhum Al-Saher" and a message of four words" hi how are u". Using the program 2 to embed the message and view the comparison between before and after the embedment in the same image.*** And by comparing between the two images" the cover image and the stego – image" , pixel by pixel , and gathering the differences of the least significant bits of the pixels, gather them as a matrix. And the extracted message is in Figure6.

**Hiding information as fractals**

An IFS is a standard way to model natural objects. The intuitive key for deriving an IFS that models any given object is self-tiling (similarity). One can always view an object as the union of several sub-objects. Let the sub-objects be actually scaled-down copies of the original object. Each of these subjects is called a tile. In particular, each sub-object is obtained by applying an affine transformation to the entire object. Now consider the original object with two or more affinely transformed copies of itself. The tiling scheme should completely cover the object, even if this necessitates overlapping the tiles.

Each transformation used to “create” a tile corresponds exactly to one map in the IFS. If the self-tiling is accurate, the IFS can be used to generate exactly that object. Otherwise, the attractor (the object that is generated by IFS) will approximate the object and the degree of closeness is measured by utilizing the Collage Theorem [4].

**Constructing the IFS from the message:**

In order to create an IFS, one first

specifies a finite set of contractive affine transformations $\{W_i ; i=1,\ldots,n\}$ in $R^2$. In general, a contractive affine transformation W in $R^2$ is of the form:

$W(X) = AX + b$, which could be used as a secret key to encipher words of length K at a time in an p-letters alphabet if and only if:

as D the characters numeric value in the key agreement between the sender and the recipient, K is the number of characters in a block, and p the prime number as their characters alphabet.

Let's assume that there are two parties ( sender and receiver ) in two far places that need to communicate secretly in way that a third person (intruder) won't figure or recognize that they are exchanging information between them .

$$W\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}e\\f\end{bmatrix}=AX+b \quad .$$

The sender, and the receiver, has different possibilities to input the calculated characters. the message characters and their values as the key to the IFS, thus the IFS's forms , are to put a "1" in the place where there is a calculated block characters, and a "0" where a no characters calculated; to be distributed in the IFS, and a sequence if 0, 1, of eight digits ignoring the last two digits. Therefore how the affine transformations are organized are another key to the message encryption, thus the orders of the formation of these maps are as follows:

$$W\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}a_{11} & 0\\ 0 & 0\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 10000000=A.$$

$$W\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}0 & 0\\ a_{21} & 0\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 00100000=A.$$

$$W\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}0 & a_{12}\\ 0 & 0\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 01000000=A.$$

$$W\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}0 & 0\\ a_{21} & a_{22}\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 00110000=A.$$

$$W\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}0 & 0\\ 0 & a_{22}\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 00010000=A.$$

$$W\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}a_{11} & a_{12}\\ 0 & 0\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 11000000=A.$$

$$W\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}a_{11} & a_{12}\\ a_{21} & 0\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 11100000=ULA.$$

$$W\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}a_{11} & 0\\ a_{21} & a_{22}\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 10110000=LLA.$$

$$W\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}a_{11} & 0\\ 0 & a_{22}\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 10010000=DA.$$

$$W\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 11110000=ALLA.$$

$$W\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}0 & a_{12}\\ a_{21} & 0\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 01100000=SDA.$$

ULA, LLA, DA, ALLA, and SDA are upper A, lower A (11100000), diagonal A (10010000), all A (11110000), and second diagonal A (01100000), respectively. Are all the possible inputs as linear transformations, the first six orders are better to be ignored, since we are talking about transformations, the matrix A should have an ***inverse.*** Now for illustration if W is an upper with all the possibilities of the offsets are

$$W\begin{bmatrix}x\\y\end{bmatrix}=A\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}e\\0\end{bmatrix}=AX+b\rightarrow 11101000=AX.$$

$$W\begin{bmatrix}x\\y\end{bmatrix}=A\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\f\end{bmatrix}=AX+b\rightarrow 11100100=AY.$$

$$W\begin{bmatrix}x\\y\end{bmatrix}=A\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}e\\f\end{bmatrix}=AX+b\rightarrow 11101100=AXY.$$

The last linear transformations with just the contrast take A as one of linear transformations but with the offset.

**Algorithm for hiding messages by fractals:**

Suppose the sender wants to hide information inside a picture in a

language of N-alphabet with message characters blocks of length k. The sender can hide so much message characters by constructing an IFS W.

Notice that the sender can form an IFS, By modifying the entries of the members of IFS and considering the aforementioned order, receiver can decipher the message. For better input to the selected formations, the sender and the receiver must choose an alphabet of a prime number of characters.

However the above order must be agreed upon between sender and receiver. Upon the receipt, the receiver retrieves the IFS using "the collage theorem or what is better known as the inverse problem technique".

For the key equations of the numeric values of the characters, assume the following example:

**Example 4:**

If the sender and the receiver agreed on the prime "29" as the alphabet of their characters, and to hide the message "***Send me some money***.", Here A-Z takes the numerical values 0-25, '.', '?', and blank '$' take 26, 27, and 28 respectively .

Let Ali be the sender and Aqeel as the receiver.

So in this case p=29 and k=2, which means a block of two characters value.

Ali selects each block of message units as $D=C_i *p+C_{i+1}$ characters value of the order of appearance in the message.

Moreover, as a linear transformation Ali will use:

$C_1*29 +C_{2...}$

$C_3 *29 +C_4$ …for $C_i$ represents the characters in the message till the value of the last character in the message is computed.

And i=1, 2, 3…the last character in the message (message length).

Note that Se=526

S=18,e=4 so :

S*29+e=18*29+4=526…..

The same is for the rest of the blocks of characters.

The calculations of the numbers and the key agreement are arranged in table1:

Ali chooses {(10110000), (10011000), (11100100)}, to form the IFS, so the IFS are:

$$W = \{L\begin{bmatrix} x \\ y \end{bmatrix} = \begin{pmatrix} \frac{526}{29^2} & 0 \\ \frac{380}{29^2} & \frac{824}{29^2} \end{pmatrix},$$

$$D\begin{bmatrix} x \\ y \end{bmatrix} = \begin{pmatrix} \frac{144}{29^2} & 0 \\ 0 & \frac{536}{29^2} \end{pmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \frac{352}{29^2} \\ 0 \end{bmatrix},$$

$$U\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{pmatrix} \frac{822}{29^2} & \frac{377}{29^2} \\ 0 & \frac{140}{29^2} \end{pmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{838}{29^2} \end{bmatrix}\}.$$

***To get the attractor of the above affine transformations, applying the outputs to the program no.1 according to what the sender and the receiver agreed on , the program applies the random algorithm that mentioned before, and the image is the attractor in Figure8.***

***To make the encoding more difficult to intruders the sender Ali is going to hide the attractor image in a colored image of 256x256 pixels size as in Figure9.***

**Space of an Image and Decoding images [6]:**

Suppose we have a particular fractally compact set A that we wish to encode as the attractor of an IFS. Then we examine the set and try to discover parts of it that might resemble the whole set. That is, we find contractions $w_1$, $w_2$, $w_3$ such that each $w_i$ (A), is approximately a small piece of A. The union of all these pieces should be approximately A.

Now imagine the collection of all possible images: clouds, trees, dogs, random junk, the surface of Jupiter, etc. A grey scale image is defined as an image whose pixel values span the grey scale, i.e., $[0, 2^{N-1}]$. We want to find a map W which takes an input image and yields an output image, just as we did before with subsets of the plane [6].

An IFS is a standard way to model natural objects. The intuitive key for deriving an IFS that models any given object is self-tiling (similarity). One can always view an object as the union of several sub-objects. Let the sub-objects be actually scaled-down copies of the original object. Each of these subjects is called a tile (the blocks).

**Conclusions and future work:**

The fractal image generation through given parameters, needs a great amount of iterations to converge into an attractor, but at the same time, it provides non uniform randomness and it is independent of the image size.

Considering the coefficients of the affine transformations is each of two letters value multiplied by the prime number "that the sender and the receiver chose as their key to the message characters", and added to the value of the next character in the message. Till the message is finished to the last character and choosing the best form of the affine transformations to apply the random iteration algorithm to get the attractor of the given message. And to make the decoding more difficult, by embedding the attractor in a colored image using the LSB; and sending it to the recipient to decode the colored image and applying the key agreement to get back the message characters, by the collage method. This way to hide information is very useful cause even if the third party ( intruders ), recognized that there is a difference in the received image, wont figure what its , whether a lose in the information or just a rubbish data.

In addition, we suggest as future work:

1) The amount of the information embedded in the other media depends on the statistical properties of the cover media, where this amount is small the noise in the media is not perceptible .

2) If a message is encrypted and hidden with a steganographic method , it provides an additional layer of protection and reduces the chance of the hidden message being detected.

3) Another embedding method should be employed like wavelet transform or DCT based method, which can provide a high level of performance in terms of quality.

4) Improved system, dealing with video images and audio files as cover media to steganography.

5) Using another file format, as a cover media, such as (tiff), and (gif) format.

**References:**

[1] A. Jacquin. *"A Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding*". PhD thesis, Georgia Institute of Technology, August 1995.

[2] A. Westfeld and A. Pfitzmann. "*Attacks on steganographic systems*." Proc. Information Hiding Workshop, Springer-Verlag, New York, 1999.

[3] Barnsely. M, *"Fractals Everywhere"*, Academic Press, San Deigo, 1989.

[4] Forte, B. and Vrscay, E.R. *"Solving the inverse problem for function, image approximation using iterated function systems*", Theoretical basis, Fractal, 2, 3, 325-334. 1994.

[5] H. Farid. *"Detecting hidden messages, using higher-order statistical models*", Int. Conf. on Image Processing, Rochester, NY, 2002.

[6] Hossam Al-Deen, Said E. El-Khamy, "*A novel secure image coding scheme using fractal transformation*", IEEE, 2002.

[7] Fridrich, J., Goljan, M., and Du, R., *"Steganalysis based on JPEG Compatibility"*, SPIE multimedia sys. and appl. IV, Denver, CO, 2001.

[8] J. Fridrich, M. Goljan, and R. Du. *"Detecting LSB Steganography in color and gray-scale images"*, IEEE Multimedia Magazine, pp. 22, 28, October 2001.

[9] Jacquin.A, *"Image coding based on a Fractal Theory of Iterated Contractive Image Transformations*", IEEE trans. On Image Processing, pp.18-30, Jan 1992

[10] Kamal Gulati, "*Information Hiding Using Fractal Encoding"*, master degree in Technology, 2003.

[11] Lala Zareh Averdissian, " *Image in Image steganography system"*, PH.D. thesis , collage of science and information, University of technology, Page 3, 12, 20, 31, June 2000.

[12] Lisa A. Soberano, "*the mathematical foundation of image compression*", thesis, The University of North Carolina at Wilmington Wilmington, North Carolina ,May 2000

[13] M. R. Khadivi, *"IFS and its use in Cryptography and Steganography "*, Jackson State University, paper to be published in spriger verlag, 2003.

[14] N. Johnson and S. Jajodia. Exploring steganography: *"Seeing the unseen"*, "Steganalysis of images created using current steganography software." Proc. Int. Workshop on Information Hiding, Springer-Verlag, Berlin, pp. 273-289, 1998.

[15] N. Johnson, Z. Duric, and S. Jajodia. *"Information Hiding: Steganography and Watermarking"*, Attacks and Counter measures, Kluwer, Academic Publishers, Boston, 2000.

[16] N. Provos. *"Defending against statistical steganalysis."* Proc. 10th USENIX Security Symposium, pp, 323- 325, 2001.

[17] Nadia Mohammed Al-Sa'idi, *"on the multi – fuzzy fractal space"*, PH.D thesis, collage of science Saddam University, page 27-28. October 2002.

[18] William J. Gilbert. *"Modern algebra with application"*, University of Waterloo., page245, 1941.

[19] Yuval Fisher. *"Fractal Image Compression*": E-book, theory and application. Springer- Verlag, 1995.

The programs

The programs used in this paper are:

Program1

```
**********
Used to get the attractor of a fractal.
**********
For n=1to num
R= rnd
K=1
Sum =p(1)
While sum < r
 k=k+1
sum = sum +p(k)
end
newx =a( k) * x+ b( k) * y  + e(k)
newy =c( k) * x+ d( k) * y  + f(k)
x= newx
y=newy
picture 1 pset (x,y)
```

```
next n
Program 2
**********************
Used to hide an image in an image .
***************************
Function hide =imhide (org,text)
% hide – encoded image which have the text % data in its LSB.

Hide =zeros ( size ( org, 1) ,size (org, 2));
For i =1: size ( org, 1)
     For j =1: size (org, 2)
        If text (i, j )<= 128
%check the pixels gray value <=128
Hide ( i, j ) =bitset (org (i, j),1,0 );

% set LSB (1st bit ) bit as '0'
Else
Hide ( i, j ) =bitset (org (i, j),1);
% set LSB (1st bit ) bit as '1'
                End
              End
End
Program 3
*******************
Used to extract an image from an image
***************************
Function [ tx1, tx2] = imxtract(diim)
tx1 =zeros (size(diim,1),size (diim,2));
tx2 =zeros (size(diim,1),size (diim,2));
for x=1 :size (diim,1)
      for  y= 1: size (diim,2)
% extract the image for text image 1

If ( mod (x+y),2)= =0

% select the pixels in alternate manner
      tx1 (x, y,: ) = diim( x, y,: );
%select the pixels from even location of (x+y)
Else
      If x~ =size(diim,1) && y~= size (diim,1)
% check whether the index of a matrix %exceeds the size
 tx1 (x,y,: )=diim(x+y+1, : );
% select the pixel from the next (x+y+1) the location of (x+y)
          End
% extract the image for next image 2
If ( x~ = size (diim,1) && y~= size (diim ,2) && (mod((x+y),2) = =0))
   tx2 (x, y ,: )=diim (x+1,y ,: );
else
tx2 (x,y ,: )= diim (x,y ,: ) ;
  end
end
```

**Table (1) The Calculations Of The Message Characters**

| | | |
|---|---|---|
| **Se** | **18*29+4** | **526** |
| **nd** | **13*29+3** | **380** |
| **$m** | **28*29+12** | **824** |
| **e$** | **4*29+28** | **144** |
| **so** | **18*29+14** | **536** |
| **me** | **12*29+4** | **352** |
| **$m** | **28*29+12** | **824** |
| **on** | **14*29+13** | **419** |
| **ey** | **4*29+24** | **140** |
| **$.** | **28*29+26** | **838** |

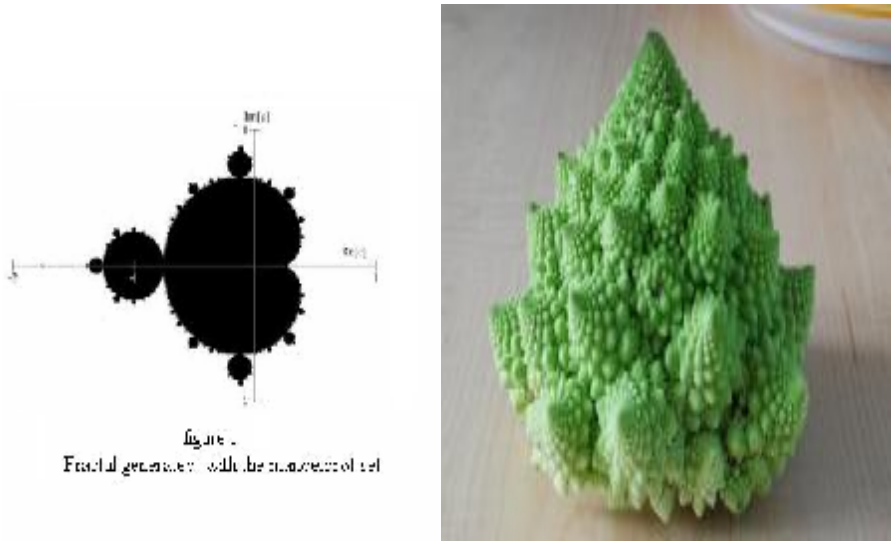

**Figure (2) A nature fractal ( a broccoli self similar structure).**

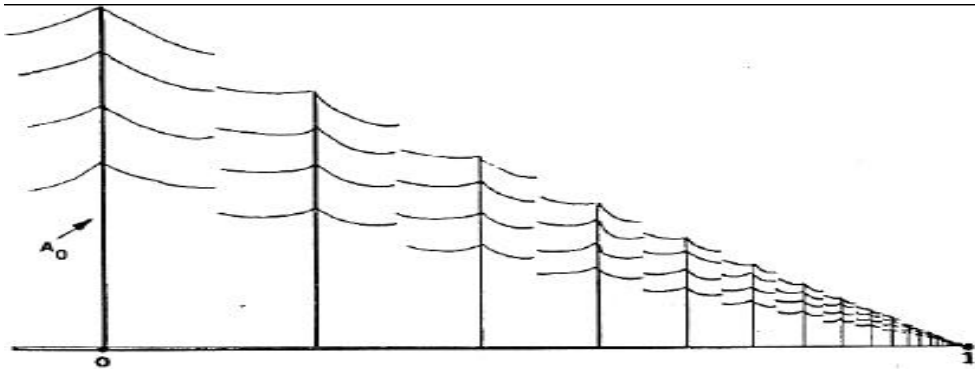


**Figure (3 )** an image explains the contraction mapping.

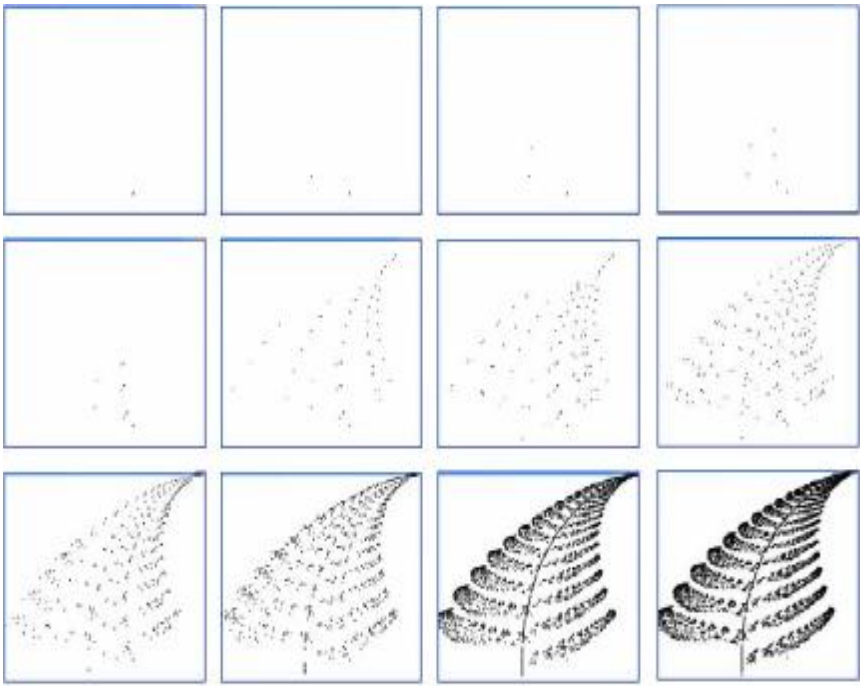

**Figure (4)** A fractal ferns 12 iterations

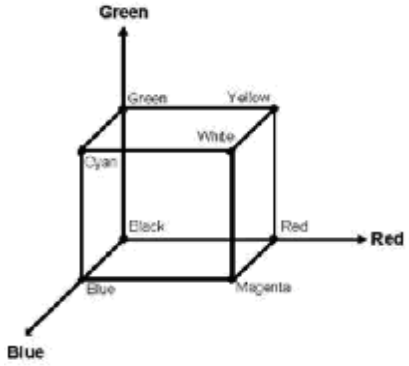


**Figure (5)** The RGB Color Cube.

**Figure (6) left Kadhum's image before embedment and to the right is the same image with the message "hi how are u"**


**Figure (7)** the message that is going to be embedded in the color image and the same message after the extraction .

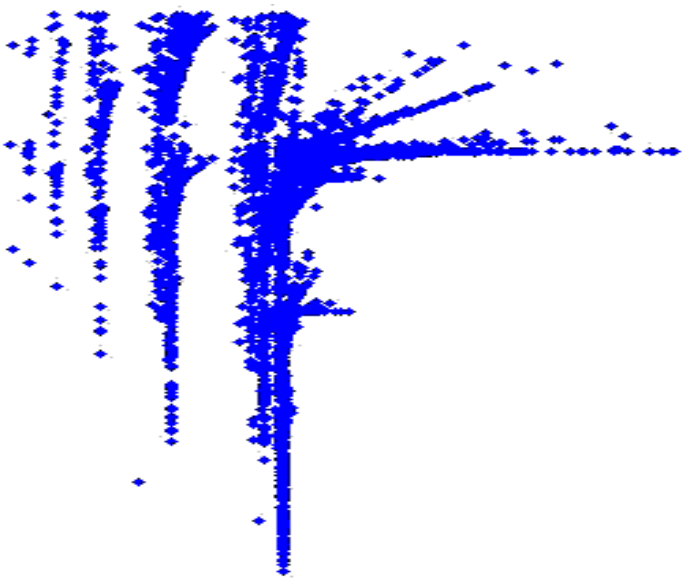

**Figure (8) The message attractor.**


**Figure (9) A colored image before and after the embedding the message of figure( 8).**