On the Security of Image Encoding Based on Fractal Functions

Nadia M. G. AL-Saidi

Applied Sciences Department-Applied Mathematics
University of Technology -Baghdad-Iraq
nadia_alsaidi@hotmail.com

***Abstract*— The information age brings some unique challenges to society. New technology and new applications bring new threats and force us to invent new protection mechanisms. So every few years, computer security needs to re-invent itself. In this paper we propose a new image encoding system utilizing fractal theories; this approach exploits the main feature of fractals generated by IFS techniques. Two levels of encryption and decryption methods performed to enhance the security of the system. The encrypted date represents the attractor generated by the IFS transformation, Collage theorem is used to find the IFS for decrypting data. The proposed method gives the possibility to hide maximum amount of data in an image that represent the attractor of the IFS without degrading its quality. Also to make the hidden data robust enough to withstand known cryptographic attacks and image processing techniques which do not change the appearance of image. The security level is high because the jointly coded images cannot be correctly reconstructed without all the required information.**

Keywords- Fractal, Iterated Function System (IFS), Attractor, collage theorem

I. INTRODUCTION

Information security is one of the important issues in the present information age; cryptography is the science of protecting the privacy of information during communication under hostile conditions. Internet changes our thinking method, life style, communication means, and others. Associated with this rapid development, there is a growing demand for cryptographic techniques, which has spurred a great deal of intensive research activities in the study of cryptography.

The fractals theory has proved to be suitable in many fields and particularly interesting in various applications of image processing. First important advances are due to M. F. Barnsley [6], who introduces for the first time the term of Iterated Function Systems (IFS) based on the self-similarity of fractal sets. Barnsley's work assumes that many objects can be closely approximated by self-similarity objects that might be generated by use of IFS simple transformations. From this assumption, the IFS can be seen as a relationship between the whole image and its parts, thus exploiting the similarities that exist between an image and its smaller parts. At that point, the main problem is how to find these transformations or, what is the same, how to define the IFS. There is, in fact, a version of the IFS theory, the Local Iterated Function Systems theory that minimizes the problem by stating that the image parts do not need to resemble the whole image but it is sufficient for them to be similar to some other bigger parts in it. It was Arnaud E. Jacquin [1], who developed an algorithm to automate the way to find a set of transformations giving a good quality to the decoded images

The outline of the paper is organized as follows; the theoretical concepts of iterated function systems are explained in Section 2, while a brief explanation of the Methodology part is provided in section 3. The core of this paper is section 4, which discusses the algorithm in addition to the software implementation with worked example. Section 5 deals with security analysis of the proposed system, followed by the conclusion in Section 6.

II. THEORATICAL BACKGROUND

This section is presented an overview of the major concepts and results of Iterated Function System (IFS) and their application. A more detailed review of the topics in this section are as in [3,4,6,7]. The theory of fractal sets is a modern domain of research. Iterated function systems have been used to define fractals. Such systems consist of sets of equations, which represent a rotation, a translation, and a scaling. These equations can generate complicated fractal images. Therefore, we need some information on dynamical systems.

Given a metric space (X,d), the space of all nonempty compact subset of X is called the Hausdorff space $H(X)$. The Hausdorff distance h is defined on $H(X)$ by,

$$h(P,Q)= \max\{\inf\{\varepsilon>0; Q\subset N_\varepsilon(P)\}, \inf\{\varepsilon>0; P\subset N_\varepsilon(Q)\}\} \quad (1)$$

Definition 1. For any two metric spaces (X,d_X) and (Y,d_Y), a transformation $\beta{:}X\rightarrow Y$ is said to be a contraction if and only if there exists a real number s, $0\leq s\leq 1$, such that $d_Y(\beta(x_i), \beta(x_j)) < sd_X(x_i,x_j)$, for any $x_i,x_j \in X$, where s is the *contractivity factor* for β.

The following theorem, known as the *contraction mapping theorem*, states an important property of contractive transformations of a complete metric space within itself.

Theorem 1. Let $\beta{:}X\rightarrow Y$ be a contraction on a complete metric space (X,d). Then, there exists a unique point $x_f \in X$ such that $\beta(x_f)=x_f$. Furthermore, for any $x\in X$, we have $\lim\limits_{n\rightarrow\infty}\beta^{\circ n}(x)=x_f$, where $\beta^{\circ n}$ denotes the n-fold composition of β.

A fractal is constructed from a collage of transformed copies of itself. It is inherently self-similar and infinitely scalable. The transformation is performed by a set of affine maps. An affine mapping of the plane is a combination of a rotation, scaling, a sheer and a translation in R^2.

Definition 2. Any affine transformation $\beta{:}R^2\rightarrow R^2$ of the plane has the form,

$$\begin{pmatrix} u \\ v \end{pmatrix} = \beta\begin{bmatrix} x \\ y \end{bmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} = A\vec{X}+b. \qquad (2)$$

where (u,v), $(x,y)\in R^2$, are any point on a plane.

By considering a metric space (X,d) and a finite set of contractive transformation $\beta_n : X\rightarrow X$, $1\leq n\leq N$, with respective contractivity factors s_n, we proceed to define a transformation B: $H(X)\rightarrow H(X)$, where $H(X)$ is the collection of nonempty, compact subsets of X, by,

$$A = B(A) = \bigcup_{i=1}^{N}\beta_i(Q) \text{ for any } Q\in H(X) \qquad (3)$$

It is easily shown that B is a contraction, with contractivity factor s=max $_{1\leq n\leq N}$ s_n. The mapping B is usually referred to as *Hutchinson operator*. It follows from the contraction mapping theorem that, if (X,d) is complete, B has a unique fixed point $A\in H(X)$, satisfying the remarkable self covering condition.

$$A = B(A) = \bigcup_{i=1}^{N}\beta_i(A) \qquad (4)$$

Definition 3. A hyperbolic IFS $\{X;\ \beta_1, \beta_2,\ldots, \beta_n\}$ consists of a complete metric space (X,d) and a finite set of contractive transformation $\beta_n{:}X\rightarrow X$ with contractivity factors s_n, for $n=1,\ldots,N$. The contractivity factor for the IFS is the maximum s among $\{s_1,\ldots,s_N\}$. The *attractor* of the IFS is the unique fixed point in $H(X)$ of the transformation B defined by (3).

However, given a set M, how to find a contractive transformation B such that its attractor A is close to M? To answer this question we have to apply to the *Collage Theorem.*

Theorem 2: For a set M and a contraction B with attractor A:

$$h(M,A) \leq \frac{h(M,B(M))}{1-s} \qquad (5)$$

where h is the *Hausdorff Distance*.

That is to say that M and A will be sufficiently close, if M and $B(M)$ are made close enough in terms of β_{i} , and combining the following two expressions:

$$B(M)=M, \quad B(M) = \bigcup_{i=1}^{N}\beta_i(M) \text{ which implies } \bigcup_{i=1}^{N}\beta_i(M) \approx M \qquad (6)$$

So, M can be partitioned as: $M = \bigcup\limits_{i=1}^{N} m_i$ and m_{i} can be closely approximated by applying a contractive affine transformation β_{i} on the whole M, where,

$$m_{\text{i}}= \beta_{\text{i}}(M) \qquad (7)$$

The theory of IFS was extended to local IFS where each part of the image is approximated by applying a contractive affine transformation on another part of the image:

$$m_{\text{i}}= \beta_{\text{i}}(D_{\text{i}}) \qquad (8)$$

D_i is the bigger part from which m_i is approximate.

III. Fractal Cryptosystem

A. Proposed Approach

There are many types of cryptography in which there are "double enciphering" and "double deciphering" processes that make the codes more difficult to crack and to analyses [12]. The proposed approach for enciphering and deciphering apply tow level method for each, for enciphering, firstly, one of the classical Cryptographic methods are used to convert message letter into integer numbers, secondly arranging the resulting code in a chosen manor of affine IFS transformation, and the resulting enciphering code is the attractor of the IFS system. For deciphering the receiver of the attractor *A* retrieves affine IFS transformation *B* using "Inverse Problems" techniques to perform the first level of deciphering method, then some algebraic calculation applied to obtain the plain text.

To illustrate the method some algebraic facts are recalls. Let *m* be a positive integer, the idea is to take *m* linear combination of the n alphabetic characters in one plaintext element thus producing the *m* alphabetic characters in one ciphertext element. An $m\times m$ matrix $K=(k_{i,j})$ is taken as a first key. Let $X=(x_1,x_2,\ldots,x_m)$ and $k\in K$ (the set of all $m\times m$ invertible matrices), compute $y=eK(X)=(y_1,y_2,\ldots,y_m)$, we say that the ciphertext is obtained from the plaintext by means of a linear transformation and K^{-1} is used for deciphering as $X=YK^{-1}$.

So we assume that our language has *n*-letter, *n* is prime, enciphering and deciphering *m* units of messages of length l at a time. *K* represents an $m\times m$ matrix whose entries belong to Z_t for which $t=n^m$, *D* represents the det(*K*). The relevant result for our purpose is that a matrix *K* has an inverse modulo *n* if and only if GCD(det(*K*),*n*)=1 [10].

Theorem. $\beta(X)=AX+b$ could be used as a secret key to encipher *p* messages of length *m* at a time in *n*-letter alphabet if and only if $GCD(D,n^m)=1$.

Proof. If *B* is secret key then *B* is one to one map from Z_t to Z_t where $t=n^m$ and hence onto and so invertible. Thus $GCD(D, n^m)=1$. Conversely if $GCD(D, n^m)=1$, then *A* is invertible and hence β is one to one.□

The sender arranges each unit of length *m* in entries with value one in the affine IFS transformation. The elements of the *B* maps are constructed from (C_{ij}/n^m) where $Cij=p_1\times n^m+p_2n^{m-1}+\ldots+p_m$.

B. Affine transformations

In order to create an IFS, one first specifies a finite set of contractive affine transformations $\{\beta_i\,;\ i=1,\ldots,n\}$ in R^2. In general, a contractive affine transformation β in R^2 is of the form: $\beta(X)=AX+b$, which could be used as a secret key to produce a encipher. There are different possibilities to arrange element in IFS invertible maps, therefore, for abbreviation, binary sequences of 0's and 1's used to represent all the possibility for element arranging in the β_i maps [9]. These possibilities are:

$$\beta\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}a_{11} & a_{12}\\ a_{21} & 0\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 111000.$$

$$\beta\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}a_{11} & 0\\ a_{21} & a_{22}\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 101100.$$

$$\beta\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}a_{11} & 0\\ 0 & a_{22}\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 100100.$$

$$\beta\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 111100.$$

$$\beta\begin{bmatrix}x\\y\end{bmatrix}=\begin{pmatrix}0 & a_{12}\\ a_{21} & 0\end{pmatrix}\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=AX\rightarrow 011000$$

All the above orders are for linear affine transformation. Now for non- linearity order each one of the above maps is extended to three forms by adding the translation part *b*. For example, for β=111000, we have;

$$\beta\begin{bmatrix}x\\y\end{bmatrix}=A\begin{bmatrix}x\\y\end{bmatrix}+\begin{bmatrix}e\\0\end{bmatrix}=AX+b\ \rightarrow 111010$$

$$\beta\begin{bmatrix} \mathrm{x} \\ \mathrm{y} \end{bmatrix} = \mathrm{A}\begin{bmatrix} \mathrm{x} \\ \mathrm{y} \end{bmatrix} + \begin{bmatrix} \mathrm{e} \\ \mathrm{f} \end{bmatrix} = \mathrm{AX} + \mathrm{b} \ \rightarrow 111011.$$

$$\beta\begin{bmatrix} x \\ y \end{bmatrix} = A\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ \mathrm{f} \end{bmatrix} = AX + b \ \rightarrow 111001.$$

IV. The Implementations.

Conversion of the plain-text message to the unreadable format is known as enciphering of the message. Similarly, conversion of the enciphered message back to the human readable form through the reversal of the encryption algorithm is known as deciphering of the message [2].

A. *The Encryption Method*

Let's assume that there are two parties(sender and receiver) in two far places that need to communicate secretly in a way that a third person (intruder) won't figure or recognize that they are exchanging information between them. However, the alphabetic, the classical encryption method, and the order of the affine IFS maps must be agreed upon between sender and receiver.

Table 1: English Alphabet used for encryption

English letters with integer values				
A=0	B=1	C=2	D=3	
E=4	F=5	G=6	H=7	
I=8	J=9	K=10	L=11	
M=12	N=13	O=14	P=15	
Q=16	R=17	S=18	T=19	
U=20	V=21	W=22	X=23	
Y=24	Z=25	$=26	.=27	?=28

Encryption Algorithm.

- The message characters are given a numbers as mentioned in Table 1, l represent the length of the message.
- Divide the message of length l into units with three characters $p_i p_{i+1} p_{i+2}$.
- Calculate the value of each unit depending on character value in Table 1 and using the polynomial $C=p_i n^2+p_{i+1}n+p_{\mathrm{i}+2}$, to perform first level of the proposed method.
- The contraction factor used is $r=1/n^m$, where m=3.
- The sender constructs the affine IFS transformations using the possibilities of the arranging mentioned above and agreed on between the sender and the receiver. The elements of β_i are calculated by $\beta_i=r*C$. Notice that $B = \{\beta_1, \beta_2,\dots \beta_{\mathrm{i}}\}$ is a (hyperbolic) IFS.
- The sender uses B to generate the attractor A using the techniques of fractal geometry.
- The sender sends the (picture) attractor A to the receiver.

B. *Decryption Method:*

An algorithm based on Jacquin's work is used as the second step of the deciphering technique. The main idea to automate the searching of a local IFS relies on a partition of the image in N non-overlapping blocks of a fixed size, called *Range Blocks*. Each range block R_i, for $i\in\{1,...,N\}$, is coded independently by matching it with a bigger block D_i in the image, called *Domain Blocks* [5]. This match defines a transformation τ_i, and the global fractal code is then given by the union $\tau=\cup\tau_i$ of local transforms as illustrated in Figure 1. Moreover, each local code τ_i restricted to consist of a reduction, a discrete isometric and an affine transformation on the luminance. Hence, τ_i can be modeled by (9):

$$\tau_i\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{pmatrix} \mathrm{a_i} & \mathrm{b_i} & 0 \\ \mathrm{c}_i & d_i & 0 \\ 0 & 0 & \mathrm{s_i} \end{pmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \mathrm{t_{i,1}} \\ \mathrm{t_{i,2}} \\ o_i \end{bmatrix}. \qquad (9)$$

where a_i, b_i, c_i, d_i, $t_{i,1}$, $t_{i,2}$ represent the geometric transforms and s_i, o_i the grey-levels transform ; *x, y* are the pixel coordinates and *z* the corresponding luminance value [14].
The search strategy used is block classification as described. Domain and range blocks are compared using rms metric (10). Given two squares containing *n* pixel intensities, $a_1, \ldots, a_i$ (from D_i) and $b_1, \ldots, b_n$ (from R_i), we can seek *s* and *o* to minimize the quantity of (11).
This will give us contrast and brightness settings that make the affinely transformed a_i values have the least squared distance from the b_i values. The minimum of *R* occurs when the partial derivatives with respect to *s* and *o* are zero, which occurs when the below equations satisfied:

$$rms = d(R,R') = \sqrt[2]{\frac{1}{B^2} \sum_{(x,y)\in R,R'} \left(R_i(x,y) - R'_i(x,y)\right)^2} \ . \tag{10}$$

$$s_i = \frac{n(\sum_{i=1}^{n} a_i\ b_i) - (\sum_{i=1}^{n} a_i)(\sum_{i=1}^{n} b_i)}{n(\sum_{i=1}^{n} a^2{}_i) - (\sum_{i=1}^{n} a_i)^2} \tag{11}$$

$$o_i = \frac{1}{n}\left(\sum_{i=1}^{n} b_i - s_i \sum_{i=1}^{n} a_i \right).$$

$$R = \left[\sum_{i=1}^{n} b_i^2 + s(\sum_{i=1}^{n} a_i^2 - 2(\sum_{i=1}^{n} a_i b_i) + 2o\sum_{i=1}^{n} a_i) + o(on^2 - 2\sum_{i=1}^{n} b_i) \right] \Big/ n^2$$

$$\text{if} \quad n^2 \sum_{i=1}^{n} a_i^2 - (\sum_{i=1}^{n} a_i\ )^2 = 0, \ \text{then} \ s = 0 \ \text{and} \ o = \sum_{i=1}^{n} b_i \big/ n^2 \ .$$

for *i = 1, 2, 3, ..., N*, here R_i and D_i are the intensities of the pixels of the range block and the decimated domain block respectively and *n = B x B* is the number of pixels in the range block. In practice, contrast s_i and brightness o_i are calculated in (10.)

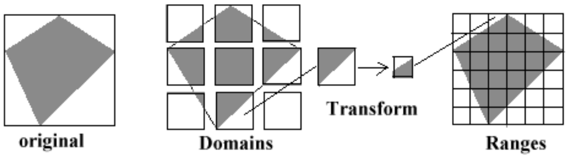

Figure.1 Domain block to range block transformation

Decryption Algorithm.

- Upon the receipt of the attractor (picture) *A*, the receiver retrieves *B* using "Inverse Problems" techniques. Let A denote the image we want to encode. Let also A_r denote a partition of *A* in nxn blocks referred to as Range Blocks (R_b). Similarly, *Ad* will denote another partition of *A*, this time in 2nx2n blocks or Domain Blocks (*Db*) in steps of nxn pixels.
- The goal of the encoding algorithm is to establish a relationship between *Ar* and *Ad* in such a way that any *Rb* can be expressed as a set of transformations to be applied on a particular *Db*. The receiver then modifies the entries of the retrieved IFS system *B* to get β_i as they agreed on.
- By multiplying each entry in the affine IFS map by n^m and rounding them to the nearest integer we perform the first level of decrypting method.
- Finally Apply some algebraic calculation to find p_1,p_2,p_3 in each cipher unit, as follows.

  p_1=int(C/n_2).

  R= C mod n_2.

  p_2=int(R/n).

  p_3=R mod n.

C. Software implementation.

The algorithm and its graphic user interface are carried out using Visual Basic. The message transforms to its corresponding as given in table 1, with a possibility to be read either from a file or direct input text. All the results have been obtained using a computer with these specifications: 3.0GHz Intel (Cor.2 Duo) CPU, and 2GB RAM.

Example: To hide the message ("hi how are you dear?"), the sender and the receiver agreed on an alphabet mentioned in table.1, to arrange the combinations between the places and the values of the characters used as inputs to the affine transformations we obtain the information as in table 2.

$$B = \begin{cases} \beta 1 = \Big\{ \quad \dfrac{1}{22*29}\begin{pmatrix} 2*10 & 6*24 \\ 12*26 & 0 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \quad \text{with prob.} = 0.2 \\ \beta 2 = \Big\{ \quad \dfrac{1}{22*29}\begin{pmatrix} 14*22 & 0 \\ 0 & 16*5 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 20*28 \\ 0 \end{pmatrix} \quad \text{with prob.} = 0.2 \\ \beta 3 = \Big\{ \quad \dfrac{1}{22*29}\begin{pmatrix} 0*0 & 1*9 \\ 5*16 & 0 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \quad \text{with prob.} 0.1 \\ \beta 4 = \Big\{ \quad \dfrac{1}{22*29}\begin{pmatrix} 8*2 & 0 \\ 0 & 9*19 \end{pmatrix} + \begin{pmatrix} 0 \\ 10*6 \end{pmatrix} \quad \text{with prob.} 0.1 \\ \beta 5 = \Big\{ \quad \dfrac{1}{22*29}\begin{pmatrix} 0*21 & 4*9 \\ 13*16 & 18*2 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 19*19 \\ 17*6 \end{pmatrix} \text{with prob.} \ 0.3 \\ \beta 6 = \Big\{ \quad \dfrac{1}{22*29}\begin{pmatrix} 3 & 7 \\ 0 & 11 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 15 \\ 0 \end{pmatrix} \quad \text{with prob.} \ 0.1 \end{cases}$$

Applying the random iterated algorithm, we get the attractor of the transformation as in the Figure 2.

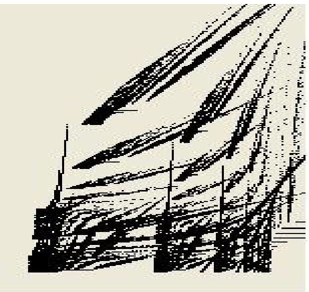

Figure.2 The attractor generated by the above iterated function systems.

V. Security And Performance Analysis.

Cryptography is an essential technology when there is need for protecting information. To make sure the system supports the law requirements that security has to be strong, the most important security attributes are the integrity and the confidentiality. Cryptography used in the right way can provide protection, confidentiality, authentication, integrity, and non-repudiation. When defining what security countermeasures the system should implement, it is important to know what threats exists, and how relevant they are [11].

To ensure fast attractor generation, the domain and the co-domain of fractal functions are defined within the infinite subfield (0,1). In this paper, a cryptosystem is formalized based on nonlinear fractal functions over (0,1). Fractal algorithm possesses sufficient security to resist some known attacks, applicable on finite field cryptosystems such as, ciphertext only attack, known plaintext attack, chosen plaintext attack, and chosen cipher text attack. The aforementioned attacks are considered as time consuming to be involved in solving non-linear systems numerically over the defined infinite subfield. As an example, a brute force attack strategy is based on explores all elements of the field in finding the secret values might be infeasible, and fail to break the system with open key space. Hence, some trial and error methods become impossible, and the adversary cannot recover the private key. He will find that the attempts along this line are meaningless as even if he can gain access to some secret parameters, because they are generated from random.

The fractal algorithm is able to resist the known attacks due the open key space and big key size. The "cumulative and truncation errors" accompanying the numerical solution of the non-linear system, pose a difficulty for the algorithm to obtain imprecise decimal numbers. Based on fractal properties, which ensure a sufficient level of randomness, introducing some of the blind signature techniques help to increase the security and randomization of the cryptosystem. (However, multiplying the message with some random reversible values, and then removing the randomization after decrypting using their inverse). The inclusion of these random values can helps to ensure a large number of unknown over number of equation, and helps to conceal the values of ciphertext through transmission. The adversary found that any attempt along this line is meaningless. Another approach is by adding different amount of noise into the image. Then attacks performed are change of brightness s_i and contrast o_i, and addition of noise.

For image encryption, security depends on two aspects, cryptographic security and perceptual security [13]. The former one denotes the encryption algorithms security against cryptographic attacks such as brute-force attack, statistical attack, differential attack, etc. The latter one denotes the unintelligibility of the encrypted image content. We will discuss the security with three different approaches:

- Which attack techniques are most likely to be used to break the cryptosystem?
- Is the cryptosystem capable of withstanding such attacks?
- Does the cryptosystem provide sufficient security in the given context?

Performance Analysis

The main objective of the fractal image coding is to find the minimum approximate error for a domain block through affine transformation to match a range block. This is done by minimizing the rms error (10) between them []. Suppose that we deal with 128x128 gray level image in which each pixel can be one of 256 levels of gray. In this case we have 32x32=1024 non-overlapping range block of size 4x44 pixels and 121x121 overlapping domain blocks of size 8x8 pixels. To find $D_i \in D$ we have to search through all D for each R_i which minimizes the rms error in (10). Since there are 8 ways to map one square into another, this mean we need to compare 8x121x121=117128 squares with each of the 1024 rang blocks. A choice of D_i along with a corresponding s_i and o_i as in (11) determine a map w_i.. The decoding is possible when we get the collection $w_1, w_2, \ldots, w_{1024}$. The original image requires 128x128 bytes of storage while the transformation requires only 3968 bytes. The IFS coding is suitable for this purpose because it offers the possibility to control the image quality during the reconstruction process.

VI. CONCLUSIONS

In this paper we have presented a new method to design a cryptographic system utilizing fractal theories. This approach employs two level methods for each of the encryption and decryption process; this is based on the fact that all fractal functions use real number to ensure satisfaction of contraction property. If the cryptosystem parameters are based on real numbers (a continuous infinite interval) then the search space is massive. Hence, many well known attacks fail to solve the nonlinear systems and find the imprecise secret key parameter from the given public one. Even if it is theoretically possible, it is computationally not feasible.

The fractal image generation through the 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 [4]. In the proposed method the IFS ($B(X)=AX+b$) could be used as a secret key to encipher p units messages of length m at a time in n-letter alphabet if and only if the GCD(D,n^m)=1, then generates the fractals associated with the IFS. The receiver can recover the message using the collage theorem and simple algebraic calculations.

REFERENCES.

[1] A. E. Jacquin, 1992, Image coding based on a fractal theory of iterated contractive image transformations. IEEE Trans. Image Processing, 1,1:18-30.

[2] A.J. Menezes, P.C.V. Oorschot, S.A Vanstone, 1997. Handbook of Applied Cryptography, Boca Raton, CRC Press.

[3] Daniel G. Piche, 2002. Complex Bases, Number Systems and Their Application to Fractal-Wavelet Image Coding. Ph.D. Thesis in applied Mathematics. Waterloo, Ontario, Canada.

[4] Forte B., Vrscay E.R., 1995. Theory of Generalized Fractal Transforms. Fractal Image Encoding and Analysis, July.

[5] J.L. Dugelay, E. Polidori and S. Roche, 1996. Iterated Function Systems for still Image Processing. IWISP-96, Manchester, UK, November.

[6] Indian Institute of Technology Bombay. Mumbai.

[7] M. Barnsley, 1993. Fractals Everywhere. Academic Press Professional, Inc., San Diego, CA, USA, second edition.

[8] M.F. Barnsley and S. Demko, 1985. Iterated function systems and the global construction of fractals, Proc. Roy. Soc. London A399, 243-275.

[9] N. Al-Saidi, Md. R. Muhammad Said, 2009. A new Approach in Cryptographic Systems using Fractal Image Coding, Journal of Mathematics and Statistics 5 (3): 183-189.

[10] N. Koblitz, 1994. A Course in Number Theory and Cryptography. Springer.

[11] Q.V. Lawande, B. R. Ivan and S. D. Dhodapkar. Chaos based cryptography: a new approach to secure communications. BARC NEWSLETTER, NUMB 258, pages 1-11,2005.

[12] R. Stinson, 2006. Cryptography: Theory and Practice. CRC Press.

[13] Shiguo Lian. Secure Fractal Image Coding. CoRR/0711.3500,2007.

[14] Y. Fisher, 1995. Fractal Image Compression: theory and application. Springer-Verlag. New York, USA.

AUTHORS PROFILE

Associate Prof. Dr. Nadia Mohammed Ghanim Al-Saidi . Applied Sciences Department-Applied Mathematics University of Technology -Baghdad-Iraq.

Table .2 the combinations of (v, cp), and their probabilities.

$\mathscr{B}$	a	b	c	d	e	f	Prob.
β_1	I= (2,10)	W= (6,24)	Y= (12,26)	0	0	0	0.2
β_2	U=(14,22)	0	0	D=(16,5)	?=(20,28)	0	0.2
β_3	"= (0,0),	H= (1,9),	O=(5,16),	0	0	0	0.1
β_4	A= (8,2),	0	0	R=(9,19)	0	E=(10,6)	0.1
β_5	"=(0,21)	H=(4,9)	O=(13,16)	A=(18,2)	R=(19,19)	E=(17,6)	0.3
β_6	$=(3,1)	$= (7,1)	0	$=(11,1)	$=(15,1)	0	0.1