Data Hiding Technique with Fractal Image Generation Method using Chaos Theory and Watermarking

K. Thamizhchelvy\(^1\) and G. Geetha\(^2\)

\(^1\)Department of Computer Applications, Sathyabama University, Chennai, India; thamizhchelvy@gmail.com
\(^2\)Division of Research and Development, Lovely Professional University, Phagwara; gitaskumar@yahoo.com

Abstract

We propose a fractal image generation method for data hiding using chaos theory. Chaos is an unpredictable behavior which arises on the dynamical system in turns explains the non-linearity and randomness. The initial condition acts as a key which is generated by the PRNG. The fractal image generation method is used to generate the fractals; while generating the fractal itself we have to hide the data. It uses the difference equation i.e., Fibonacci series for data hiding. If the initial value is identified wrongly then chaotic behavior arises in the system that may lead to chaos. It is very difficult for the steganalysis to break the data. The generated fractal images are watermarked and it can be used as a digital signature for any online applications.

Keywords: Chaos, Fractals, Pseudo Random Number Generator (PRNG)

1. Introduction

1.1 Fractals

The geometric shapes of the fractals are very complex and it is infinitely detailed. The small sections of them are recursively defined and it is similar to large ones. The Function of the fractals is \(f(x)\) is to consider \(x, f(x), f(f(x)), f(f(f(x))), \) etc. The definite properties of the complex systems of the fractals are closely related to Chaos.

1.1.1 Image Encoding using Fractals

In nature everywhere the fractals are seen. The fractals have the elements of chaos and it is first imagined by Julia and Mandelbrot, but the essence is based in mathematics. This is not only the way to represents the fractals. Using Iterated Function System (IFS) technique the Sierpinski gasket is drawn. Sierpinski thought the fractal originally.

It is originally from the normal triangle, choose any vertices from the triangle, and start from the middle to draw the triangle, repeat this process infinitely for number of iterations.

The simple and complex fractal images like koch snowflake, sierpinski triangle, etc are given shown in Figure 1.

In this paper we proposed a new Image Generation Algorithm using chaos theory and watermarking. The paper is organized as follows - Section 2 deals with previous works on fractals, Section 3 explains the techniques of Chaos Theory and Pseudo Random Number Generator (PRNG), in section 4 discusses about the Image Generation Algorithm processes, Algorithm, Flowchart Representation and experimental data results, and in section 5 we bring out the conclusion and future works.

\[\text{Figure 1. Fractal images of different shapes.}\]
2. Previous Works

Bilal et al.\(^1\) proposed an algorithm hides the payload based on certain relationship between the cover image, chaotic sequence and the payload, instead of directly embedding payload into the cover image which often leaves telltale signs of steganography.

Tayel\(^2\) proposed an algorithm based on coordinate the data in the image dimensions using chaos distribution arrangement. The data is embedded with the original image in the pixels least significant bits, so can’t appears within the image. When the image received the embedded data is separated and rearranged using the initial condition of the chaos coordination.

Wu and Noonan\(^3\) proposed an algorithm uses a fractal image as the cover image, takes a random-like sequence generated by a chaotic map as the reference for embedded positions, and employs a wavelet transform to realize the embedding procedure.

Thamizhchelvy and Geetha\(^4,5\) proposed a new Message Authentication Image (MAI) Algorithm that provides confidentiality, authentication and digital signature. It uses Cryptographic and Steganographic ideas to conceal the data in the image and to protect against e-banking fraud and the latest e-banking fraud techniques, such as Phishing, Trojans and man-in-the-middle attacks. MAI generates fractals and embeds the password using chaos technique.

Kiani et al.\(^6\) a new approach for embedding the authentication data using fractal and chaos technique is developed in this paper. This approach is to meet the need for authentication, including fragility and inalterability. Using chaos technique and the keyword extracted from the image itself, it is so unlikely to predict the blocks used for embedding the data.

El-Khamy et al.\(^7\) proposed a new steganography technique for hiding images. It adopts both fractal and wavelet image processing techniques. The idea of the presented scheme is to hide the fractal codebook of to be-hidden image in the wavelet domain of a host or hiding image. The presented technique had been tested and proved its robustness against additive white Gaussian noise AWGN.

Geetha showed that non-linearity plays a vital role in cryptographic algorithms by appealing to chaos and quantum chaos\(^8\).

In Shannon’s paper “Communication theory of secrecy systems”, he mentioned that the idea of using chaos in cryptography\(^9\) and this transformation are used in secrecy system. Some more researches also pointed out that there is the tight relationship between chaos and cryptography\(^9\)–\(^13\),\(^15\). The future behavior of chaos is unpredictable, if two close starting points diverge exponentially and it is arises in the dynamical system, based on this idea, Geetha et al.\(^16\) introduced chaos in cryptography as follows. The embedded Markov chain explains the stochastic matrix. With the help of this the sequence of Fibonacci is generated. We plan to introduce chaos in steganography and it could be embedded in fractals, it explains the non-linear system.

3. Chaos Theory

The chaos theory is an unpredictable behavior arises in the deterministic system. The behavior of the system is identified by the initial condition. They set the initial condition as the “key”. This initial condition is generated by the Pseudo Random Number Generator (PRNG). If the initial state of the systems are known in advance then the future state of the system could predicted theoretically. The great sensitivity to the initial condition of the chaos is hard to break it by the steganalyst. The precision limits the future state and initial state can be measured, chaotic systems are examined by the strong dependence on the initial conditions.

3.1 Pseudo Random Number Generator (PRNG)

The evolution of the Pseudo Random Number Generator (PRNG) is deliberately made hard to predict even though it is a deterministic algorithm. The numerical sequences are produced by PRNG using deterministic algorithm. The statistical Pattern tests are passed by the pseudorandom sequences for randomness. By make use of this algorithm, conditions are used to initialize it, called the “seed”, the output can be predicted. Because the sequence of the numbers which is generated by the PRNG is predictable. But the Hardware random number generators produce the sequences of numbers are no in a predictable way, and it provide the greatest security for date encryption. It is completely deterministic system. In higher levels, a seed state of a PRNG sometimes called as a “key” and is used as arguments to a method of an algorithm to produce the random numbers. The deterministic part is the “key”. If you run the algorithm with the same “key” it will produce the same random result but this is not truly random.
4. Image Generation
Algorithm Processes

The Image Generation Algorithm composed of two processes
1. Forward Process
2. Reverse Process

4.1 The Process of Image Generation Algorithm

4.1.1 Forward Process

1. Generate a Fractal Image - While generating an image itself, we have to hide a hashed data using Chaos Theory.
   • Change the Input data into hashed data.
   • Hashed Data -> Use SHA2 algorithm for Hashing.
   • Image -> Fractal Image using IFS Technique.
   • Hide -> Use Chaos Theory and apply generalized Fibonacci sequence.
2. Generated fractal Image serves as watermark image.
3. Embed it in any photographs.

4.1.2 Reverse Process

1. Scan the photograph.
2. Extract Photo and Fractal image separately.
3. Extract hashed data from fractal image.
4. At last compare both the hashed values.

4.2 Algorithm
Assumptions:
Hv - Hashed Value
FImg - Fractal Image
GFImg - Generate Fractal Image
FEdImg - Fractal Embedded data Image
SPh - Scanned Photo

4.2.1 Forward Process

Step 1: Get (M) // Read a Data.
Step 2: Hv=H(M) // Use SHA2 Algorithm to generate the Hashed value.
Step 3: GFImg=Generate (FImg) // Generate the fractal image by using Iterated Function System (IFS) Technique.
Step 4: FEdImg=Generate (Chaos (GFImg , Hv)) // While generating the image itself, we have to hide the hashed value using chaos theory and we will get the Fractal with embedded data image.
Step 5: Watermarking Process
   Step 5.1: Sph=scan (Photo) // Scan the Photograph
   Step 5.2: Image=Embed (SPh, FEdImg) // Apply the 3 properties of Watermarking and embed the watermark (FEdImg) Image.
   Step 5.3: Image generation algorithm generated the final secured image.

4.2.2 Reverse Process

Step 6: Scan (Image) // Scan the image
Step 7: Extract (SPh, FEdImg) -> Image // Extract the photo and Fractal with embedded Data (FEdImg) from the scanned photo.
Step 8: Hv1= Extract (Hv, FEdImg) // Use Hashing algorithm to separate the Hashed value from fractal Embedded data image.
Step 9: Compare (Hv, Hv1) // Compare both the hashed values, if both the values are same then it proves that they are original otherwise, it is duplicate.
Step 10: Stop

4.3 Algorithm Description
Assumptions:
Hv - Hashed Value
FImg - Fractal Image
GFImg - Generate Fractal Image
FEdImg - Fractal Embedded data Image
SPh - Scanned Photo
4.3.1 Forward Process

Step 1: Read the Data.

Example:

Data: K.Thamizhchelvy 30 F Chennai

Step 2: Use SHA2 Algorithm to generate the Hashed value.

Code: Hv=H(M)

The SHA (Secure Hash Algorithm) is one of a number of cryptographic hash functions. A cryptographic hash is like a signature for a text or a data file. SHA-256 algorithm generates an almost-unique, fixed size 256-bit (32-byte) hash. Hash is a one way function – it cannot be decrypted back. This makes it suitable for password validation, challenge hash authentication; anti-tamper, digital signatures. SHA-256 is one of the successor hash functions to SHA-1 and is one of the strongest hash functions available.

Example:

The given data (K.Thamizhchelvy 30 F Chennai) is converted into hashed value (5c027cf0365696ef224f3eba81467b01aac0535c471208187fed3b8dc26f981c) by using SHA2 algorithm. Then the hashed value is inserted into the fractal image.

SHA-256 hash: 5c027cf0365696ef224f3eba81467b01aac0535c471208187fed3b8dc26f981c

Step 3: Generate the fractal image by using Iterated Function System (IFS) Technique.

Code: GFImg=Generate (FImg)

Example: Spleenwort fern (Figure 2)

For the two dimensional case the affine transformations have the following forms:

Equations:

\[ x_{n+1} = ax_n + by_n + e \]
\[ y_{n+1} = cx_n + dy_n + f \]

The coefficients a, b, c, d, e and f are the IFS "code". A given image will normally require multiple transformations, each with their own set of coefficients. In the random iteration algorithm each transformation is assigned a probability p. With each round of iteration one of the transformations is chosen randomly, using the probability as factor in the choice, and the transformed point is plotted on the graphic plane. As the points are plotted the image emerges. The most famous of IFS images is probably the spleenwort fern it uses four affine transformations.

The Table 1 has the IFS code for producing the image.

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.16</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>0.85</td>
<td>0.04</td>
<td>−0.04</td>
<td>0.85</td>
<td>0.00</td>
<td>1.60</td>
<td>0.85</td>
</tr>
<tr>
<td>0.20</td>
<td>−0.26</td>
<td>0.23</td>
<td>0.22</td>
<td>0.00</td>
<td>1.60</td>
<td>0.07</td>
</tr>
<tr>
<td>−0.15</td>
<td>0.28</td>
<td>0.26</td>
<td>0.24</td>
<td>0.00</td>
<td>0.44</td>
<td>0.07</td>
</tr>
</tbody>
</table>

Step 4: While generating the image itself, we have to hide the hashed value using chaos theory and we will get the Fractal with embedded data image (Figure 3).

Code: FEdImg=Generate(Chaos(GFImg, Hv))

![Figure 2. Spleenwort fern.](image)

![Figure 3. Generated fractal image.](image)
Step 5: Watermarking Process (Figures 4 and 5)

5.1 Scan the Photograph
Code: $\text{Sph} = \text{scan (Photo)}$

Figure 4. Original image.

4.3.2 Reverse Process

Step 1: Scan the Final Secured output image (Figure 7)
Code: $\text{Scan (Image)}$

Figure 7. Final Secured Image

5.2 Apply the 3 properties of watermarking and embed the watermark ($\text{FEdImg}$).

Code: $\text{Image} = \text{Embed (SPh, FEdImg)}$

Figure 5. Watermarking system.

5.3: Image generation algorithm generated the final secured image (Figure 6).

Figure 6. Final secured image.
Step 2: Extract the photo and Fractal with embedded data image (FEdImg) from the scanned photo (Figure 8).

Code: $\text{Extract(SPh, FEdImg)}$ -> Image

Step 3: Use an algorithm to extract the Haashed value from fractal Embedded data image (Figure 9).

Code: $\text{Hv1 = Extract(Hv, FEdImg)}$

Step 4: Compare both the hashed values; if both the values are same then it proves that they are original user so it allows the user. Otherwise, it is duplicate user so it rejects the user.

Code: $\text{compare(Hv, Hv1)}$

\[
\text{If (Hv == Hv1) then}
\]
\[\text{// It is original, Allow the User}\]
\[
\text{Else}
\]
\[\text{// Duplicate, Reject the User}\]
\[
\text{Endif}
\]

4.4 Flow Representation

Figure 8. Extraction

Figure 9. Extracted hash code.
4.5 Experimental Data Result Comparison

4.5.1 Result 1
Input: HV and HV1
Final Result: ORIGINAL USER

Compare (HV, HV1)
Result:
ItProves that it is Original User, so it Allows the User.

4.5.2 Result 2
Input: HV and HV1
Final Result: DUPLICATE USER

Compare (HV, HV1)
Result:
ItProves that it is Duplicate User, so it Rejects the User.
5. Conclusion

We proposed a fractal image generation method to hide the data using chaos theory. In fractal generation method we generate the fractal, while generating the fractal itself we have to hide the data. We use the difference equation i.e., Fibonacci series for hiding the data. The initial value is a key which is set by the PRNG. If the initial value is identified wrongly then chaotic behavior arises. So it is very difficult for the steganalysis to break the data. Once the image is generated, we watermark this image and use that image as digital signature for any online applications.

6. References

2. Tayel M. A new chaos steganography algorithm for hiding multimedia data, 14th International Conference on Advanced Communication Technology (ICACT); 2012 Feb 19–22; Korea (South).