

A 3D hybrid chaotic system for image steganography based on framelet and SVD

Sepideh Nikseresht^a, Ahmad Yousefian Darani^{a,*}, Hadi Seyedarabi^b, Yaver Khedmati Yengejeh^a

^aDepartment of Mathematics, Faculty of Mathematical Sciences, University of Mohaghegh Ardabili, 5619911367, Ardabil, Iran

^bFaculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

(Communicated by Haydar Akca)

Abstract

In this paper, we introduce an innovative steganography technique that employs discrete framelet transform coupled with singular value decomposition (SVD) for the purpose of embedding a color secret image inside a fractal-based cover image. The process utilizes an alpha scale factor, a crucial parameter that adjusts the level of embedding to ensure the secret image blends smoothly without compromising the visual quality of the fractal cover. To enhance resistance against possible targets, we have also designed a new shuffle operator to scramble pixels of the cover image. In addition, we have incorporated a 3D hybrid chaotic system which combines Tinkerbell, Gingerbreadman and Duffing chaos maps. The results of our experiments clearly show that our approach not only produces stego images with suitable quality, but also guarantees the security and resilience of the hidden image against different types of attacks.

Keywords: Steganography, SVD, Framelet, Image, Chaotic system
2020 MSC: Primary 94A08; Secondary 68P30

1 Introduction

In recent years, with the progress of science and the increase of knowledge, the transfer of information and messages with different content has also increased and so data security on the internet is a big challenge [11]. The transmission of information is required to maintain security and not reveal it to other people and hackers. Encryption is one of the ways to increase security, but it demonstrates secret information to a third party and causes curiosity and low security. Steganography is another method which is more secure than cryptography, because its cover security layer makes the third party unaware of secret information. In image steganography, the security of secret image against attacks, cover image capacity, and stego image quality are essential. Steganography encompasses a range of techniques used to conceal information within various mediums. In [1] encrypted sensitive data using a digital dictionary and the AES algorithm, and then embeds the encrypted data within an image. Reference [28], proposed a wavelet-based neural network for side-information estimation in JPEG steganography. The method utilizes wavelet transforms on the network input and employs a combined Charbonnier, wavelet, and perceptual loss function to enhance embedding direction adjustment. In [5] introduced two steganography methods based on IPVDS (Indicator-based Pixel-Value

*Corresponding author

Email addresses: ssnikseresht@uma.ac.ir (Sepideh Nikseresht), yousefian@uma.ac.ir (Ahmad Yousefian Darani), seyedarabi@tabrizu.ac.ir (Hadi Seyedarabi), khedmati.y@uma.ac.ir (Yaver Khedmati Yengejeh)

Differencing Steganography) and optimized by a Genetic Algorithm (GA). In [9] used particle swarm optimization and chaos theory to select optimal pixel locations in a cover image for embedding secret data. The host and secret images are divided into blocks, allowing each block to store a relevant amount of secret bits. In reference [15], the system hid secret data within a cover image by detecting edges using the xDoG algorithm and replacing the 4 Least Significant Bits (LSB) of pixels in those areas. Subsequently, the hash value of the stego image was recorded on the block chain to ensure its authenticity and integrity. In reference [10], the ACO-LSB method, which utilizes Ant Colony Optimization, is employed to enhance image steganography by embedding additional bits in gray-scale images. This technique increases data capacity by up to 30. In [14], a steganography method is described for concealing a user's message within an image using the LSB technique. In this approach, the message is first converted into cipher text and then embedded within a cover image. Additionally, both visible and invisible watermarking are utilized to enhance security and identification. In [4], a full and blind color image steganography algorithm is proposed using fractal, SVD, DWT, and integer wavelet transform (IWT). This algorithm ingeniously embeds the secret color image within a color fractal, providing a substantial capacity for concealing extensive amounts of secret data. The use of fractal images as covers serves as an efficient method for masking information, ensuring the preservation of the original image's quality while simultaneously protecting the embedded data from potential security breaches. Reference [22] utilizes framelet transform along with SVD for embedding a secret gray-scale image within a cover gray-scale image. This technique ensures that the resulting steganographic images, or stego images, not only retain high visual quality but also demonstrate resilience against a range of attacks. In [13], the authors introduced various shift operators and proposed a novel 2D hybrid chaos map for image secure transforms based on framelet. In [24], the authors proposed an algorithm based on haar DWT using alpha blending and reference [3] proposed a new BESOPS-CE based image steganography technique by IWT and chaotic system. In [12] introduced an algorithm that integrates a hybrid 1D and 2D chaotic map with optimization and LSB techniques to improve embedding efficiency. This study introduces an innovative steganography technique that conceals a color secret image within a fractal cover image. This approach employs a complex hybrid method involving a novel 3D chaotic system, a proposed shuffle operator, framelet transform, and SVD. Shuffling pixels enhances security by providing an additional layer of protection, making it considerably more challenging for potential attackers to detect concealed data[16] Furthermore, in our proposed algorithm, the shuffle of pixels and the dependency of the shuffle amount on the output of the chaotic function enhance the algorithm's security compared to similar methods. Given the sensitivity of the chaotic function to initial values, where even the slightest change in input can cause significant variations in the function's output, a portion of the required key is considered as an input to the function. This minor alteration in the key modifies the shuffle amount, posing a challenge to image extraction. Additionally, the use of the framelet transform and SVD, combined with a 3D chaotic system, plays a crucial role in our algorithm. In proposed algorithm fractal images are used as cover images, and SVD is employed as a tool for feature extraction. One significant advantage of this method is that there is no need to send the cover image to the extractor. Instead, the extractor can reconstruct the fractal cover image using the necessary parameters provided as a key. This reconstructed cover image is then utilized for extracting the secret image. Framelets offer complete reconstruction, scattering, and stability, effectively overcoming the limitations of traditional wavelet transforms, such as instability and poor directional sensitivity. For this reason, we used framelets instead of wavelets, resulting in significantly better outcomes in our analysis and applications. Although concealing color images in a 3D context presents more challenges than working with gray images, the results achieved were satisfactory. This technique greatly improves the creation of high-quality stego images, outperforming modern Least Significant Bit (LSB) methods. The strong embedding process reduces image distortion and maintains visual clarity. As a result, our approach achieves better outcomes in terms of being unnoticed and overall image quality, making it more effective for data hiding. The high quality of the resulting stego images ensures that the hidden data remains undetectable, thus enhancing both the security and practical usefulness of the method. While embedding color images within a 3D context presents greater complexity compared to gray-scale images, the achieved outcomes are notably satisfactory. This methodology significantly surpasses contemporary Least Significant Bit (LSB) techniques by producing stego images of superior quality. The robust data embedding process effectively minimizes image distortion while preserving visual clarity. Consequently, our approach yields enhanced results in terms of both imperceptibility and overall image quality, thereby rendering it more efficient for data concealment. The high caliber of the resultant stego images guarantees that the embedded data remains undetectable, thus augmenting both the security and practical applicability of the method. This provides an effective means for inconspicuous data transmission. This makes the technique a strong choice for applications where both image quality and data stealth are of paramount importance, ensuring that sensitive information can be transmitted securely without compromising the visual integrity of the cover image. In conclusion, we assessed the quality of the stego images and then evaluated the security of the secret image through various attacks, achieving satisfactory results.

1.1 Mathematics concepts

In this section, we introduce basic mathematical concepts such as fractal, framelet transform, and SVD.

1.1.1 Fractal

One of the primary sources of human inspiration, directly and indirectly, has been nature, and the order governing its core is a fractal. Fractals are shapes that differ from Euclidean geometric shapes in that they lack regularity and exhibit the same level of irregularity across all scales. Fractals are shapes that, unlike Euclidean geometric shapes, are not normal in any way and are inconsistent in all dimensions and express order in disorder. Euclidean shapes are produced using static functions, but fractal shapes are made by dynamic processes. Dynamic processes have memory and their behavior depends on the past. Fractal sets can mathematically describe many complex and irregular shapes in nature. In other words, a fractal is a structure in which each part is the same as the whole, and this characteristic of fractals is called self-similarity. Not every repetitive and dynamic process creates complex fractal structures. The mechanism for creating such dynamic form is chaos, and a fractal is a mathematical representation of chaos. For example, a pine tree is a self-similar object because each branch is very similar to a pine tree, but on a much smaller scale. Of course, there are many examples of fractal objects in nature. For example, snowflakes, clouds, rivers, ferns, and even cauliflower are fractal objects. Silicon chips or the Sierpinski triangle can also be examples of artificial fractal [17, 8]. The Julia set results from the singular quadratic function $f(z) = z^2 + c$, where c is a complex number. The Julia set has complex structures, and self-similar features. By zooming in on an area of the boundary of the Julia set, an image similar to the Julia set is observed. This set has properties such as infinity and chaos, and for this reason, it is susceptible to small disturbances and changes. That is, it is prudent to initial conditions. A small perturbation can lead to punctual changes in the repeating sequence f . We have given one of these forms in Figure 1a. The Mandelbrot set is a kind of fractal taken from Julia's collection, but it has some differences. This set is the function of the Julia set with complex parameter c , but in this set, the pixels change in the whole image [23]. In the study, the utilization of Ultra Fractal 6.05 plays a pivotal role in the creation of fractal visuals, necessitating specific parameters to fabricate the imagery. These parameters include color space, type, formula, location, etc [21]. An example of this is given in Figure 1b.

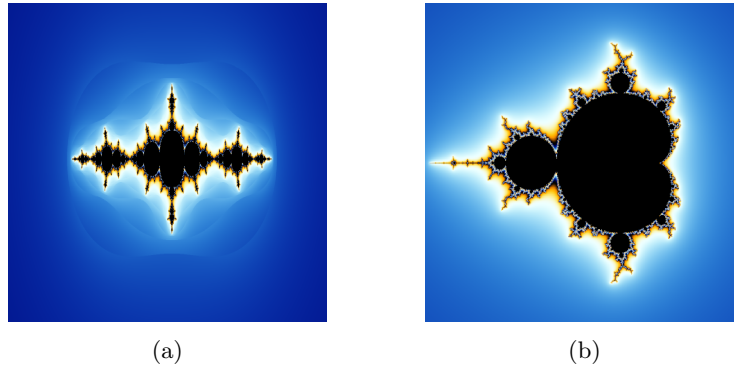


Figure 1: (a) Julia fractal, (b) Mandelbrot fractal.

1.1.2 Framelet Transform

The DWT employs two main operators: the transfer operator, for breaking down the data's frequency into lower resolution levels, and the subdivision operator, for reconstructing and improving the data to higher resolution levels. Framelets have advantages over traditional wavelet methods due to their unique properties. In wavelets, small changes in the input signal can significantly alter the distribution of coefficients across scales. However, in framelet sub-bands, coefficients within one sub-band are correlated with those in other sub-bands, allowing changes in a coefficient after decomposition to be compensated by adjustments in other coefficients during reconstruction, significantly decreasing noise in the final image. Adding extra data into framelets generally has a negligible impact, facilitating accurate stego image reconstruction. This is due to framelets' lower sensitivity to changes compared to the higher sensitivity found in wavelets.

The Framelet transform is constructed using a basis composed of one scale function, $\varphi(t)$, and two wavelet functions, $\Psi_1(t)$ and $\Psi_2(t)$. These functions are generated using a low-pass filter (scaling) $h_0(n)$ and two high-pass (wavelet) filters $[h_1(n), h_2(n)]$.

$$\varphi(t) = \sqrt{2} \sum_n h_0(n) \varphi(2t - n) \quad (1.1)$$

$$\Psi_i(t) = \sqrt{2} \sum_n h_i(n) \varphi(2t - n), \quad i = 1, 2. \quad (1.2)$$

So, the signal $f(t)$ can be expressed as an expansion using these framelet functions:

$$f(t) = \sum_{k=-\infty}^{\infty} c(k) \varphi_k(t) + \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} d_1(j, k) \Psi_{1,j,k}(t) + d_2(j, k) \Psi_{2,j,k}(t) \quad (1.3)$$

where the decomposition coefficients are given by:

$$c(k) = \int f(t) \varphi_k(t) dt \quad (1.4)$$

The coefficients $c(k)$ are the approximation coefficients.

$$d_i(j, k) = \int f(t) \Psi_{i,j,k}(t) dt, \quad i = 1, 2. \quad (1.5)$$

The coefficients $d_i(j, k)$ are the detail coefficients.

To preserve the quality of the secret image, this feature is utilized for embedding the secret image. The framelet transform is an enhanced form of a tight frame that ensures energy conservation and maintains an isometric relationship between the input signal and the output coefficients. This property enables us to utilize its convolution as an inverse operation. As a result, even if some data is lost, it can be tolerated without any negative consequences. Ultimately, this leads to the creation of high-quality stego images [2].

1.1.3 Singular value decomposition

SVD is a matrix factorization that has gained much attention since an efficient algorithm for its computation was developed in the 1970s and is still widely used. A real-valued matrix $A \in \mathbb{R}^{m \times n}$ can be factored into a SVD (1.6),

$$A = USV^T, \quad (1.6)$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$, are orthogonal matrices, $V^T \in \mathbb{R}^{n \times n}$ is the transpose of V and $S \in \mathbb{R}^{m \times n}$ is diagonal with $r = \text{rank}(A)$. In the proposed scheme, the diagonal values in matrix S are singular values of original image matrix A . So, the coefficients of A are wavelet coefficients of image subbands. The columns of matrix V and matrix U are called right-singular vectors and left-singular vectors of matrix A , respectively.

2 A 3D hybrid chaotic system

In this section, after the preliminary definitions of chaos and introduction of some chaos functions, we introduce the proposed chaos function. Chaotic maps are mathematical functions that represent chaotic behavior. Chaotic maps are a set of dynamic systems with unique properties, such as periodicity, unpredictability, sensitivity to initial values, and control parameters. These features make them useful in various fields such as steganography and cryptography. Chaotic systems are used for algorithms to increase security. The unique part of chaos in hiding information is its extreme delicacy in the initial position and the expansion of circuits in the whole space. These best features make cluttered maps a great way to hide data [19, 13]. The primary motivations for employing chaotic maps include ensuring the secure transfer of digital images, the robustness of these images against slight pixel modifications, and the pronounced correlation among adjacent pixels. Of late, cryptographic algorithms that leverage chaotic systems have gained popularity. The dynamic system has two types, 1-D and n-D ($n \geq 2$), which depend on the dimensions. In the 1-D scale of calculations, simplification is more straightforward, and processing is faster, and in general, it has

a simple structure. Encryption systems are generally designed using low-dimensional anomaly maps such as 1-D. Such maps are less secure and fragile against attacks, and their output can be easily predicted. However, the second type of system, which has higher dimensions, is more secure[7]. Although these mappings are more complex and have complex calculations with more errors, their sequence can be used in various areas. Among its applications is the design of unbreakable encryption algorithms, which have recently been studied [25, 31]. Chaotic maps have problems and we try to solve them using hybrid chaotic maps. We propose a novel chaotic system obtained using the Tinkerbell map, Gingerbreadman map, Duffing map, trigonometric and exponential functions. This proposed map is a 3-D dynamic system and enhances security and quality. The used chaotic maps explain as follows:

2.1 Tinkerbell map

The Tinkerbell map is defined as a 2D discrete-time chaotic system that settles as (2.1).

$$\begin{cases} x_{n+1} = x_n^2 + y_n^2 + ax_n + by_n, \\ y_{n+1} = 2x_n y_n + cx_n + dy_n. \end{cases} \quad (2.1)$$

In this equation, parameters a, b, c, and d dictate the system's behavior, while n denotes the step in the discrete iteration process. Visually, this system's dynamics resemble Tinkerbell's flight path around the Cinderella Castle, famously depicted in the opening sequence of Disney animations. When the parameter increases, it moves from a fixed circle to two selected coexisting circles and then thirteen small turbulent coexisting sets to unattractive turbulent sets. It is also observed that by changing parameter b, there are turbulent gravitations, ten sets of coexistence perturbations, periodic circuits, fixed circles, and two sets of absorption perturbations with non-attractive gravitational sets. Observed periodicity and rough behaviors. With d changes, there are paths from a periodic circuit to a fixed circle, then to unattractive turbulent settings, and finally to an attractive turbulent settings [29].

2.2 Gingerbreadman map

In the theory of dynamical systems, the Gingerbreadman map is a 2D chaotic map. This map is chaotic for definite primary conditions and parameters. The graphical picture of the system is similar to the gingerbread man which is a biscuit or cookie made of gingerbread [6]. This map is defined as (2.2).

$$\begin{cases} x_{n+1} = 1 - y_n + |x_n|, \\ y_{n+1} = x_n. \end{cases} \quad (2.2)$$

2.3 Duffing map

The Duffing map represents a 2D discrete-time dynamical system known for exhibiting chaotic dynamics. This map is chaotic for definite primary constants a, and b [20]. The Duffing map is defined as (2.3).

$$\begin{cases} x_{n+1} = y_n, \\ y_{n+1} = -bx_n + ay_n - y_n^3. \end{cases} \quad (2.3)$$

2.4 Proposed 3D chaotic system

In this section, we introduce a novel three-dimensional chaotic mapping. The proposed system is constructed by integrating the core dynamics of the Tinkerbell map with the nonlinearities of the Gingerbread and Duffing maps, supplemented by trigonometric and exponential modulations. The architecture of this mapping is designed to enhance sensitivity to initial conditions and expand the complexity of the phase space. The complete mathematical formulation

of the proposed three-dimensional chaotic mapping is given by (2.4):

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} x_{n+1} = 10^9 \exp(10^{-3} z_n) f(gin(x_n, y_n), duf(x_n, y_n), \\ \sin(y_n) \cosh(x_n)), \quad \text{mod } 1; \quad z_n < x_n, \\ x_{n+1} = 10^9 \exp(10^{-3} y_n) g(gin(x_n, y_n), duf(x_n, z_n), \\ \cos(y_n) \sinh(x_n)) \quad \text{mod } 1; \quad z_n \geq x_n, \end{array} \right. \\ \left\{ \begin{array}{l} y_{n+1} = 10^9 \exp(10^{-3} x_n) f(gin(x_{n+1}, y_n), duf(x_{n+1}, y_n), \\ \cos(x_n) \sinh(y_n)), \quad \text{mod } 1; \quad y_n < \frac{1}{2}, \\ y_{n+1} = 10^9 \exp(10^{-3} z_n) g(gin(x_{n+1}, y_n), duf(x_n, y_n), \\ \sin(z_n) \cosh(y_n)), \quad \text{mod } 1; \quad y_n \geq \frac{1}{2}, \end{array} \right. \\ \left\{ \begin{array}{l} z_{n+1} = 10^9 \exp(10^{-3} y_n) f(gin(x_{n+1}, y_n), duf(x_n, y_n, 2a, b), \\ \sin(x_{n+1}) \cosh(z_n), a, 2b), \quad \text{mod } 1; \quad z_n < \frac{1}{2}, \\ z_{n+1} = 10^9 \exp(10^{-3} x_n) g(gin(x_{n+1}, y_{n+1}), duf(x_{n+1}, y_{n+1}), \\ \cos(y_{n+1}) \sinh(z_n)), \quad \text{mod } 1; \quad z_n \geq \frac{1}{2}, \end{array} \right. \end{array} \right. \quad (2.4)$$

In the proposed three-dimensional mapping, two nonlinear functions, $f, g : \mathbb{R}^3 \rightarrow \mathbb{R}$, serve as the core components of the system's dynamics. These functions represent an extended version of the Tinkerbell map, which incorporates an additional dimension and control coefficients to induce a more complex and chaotic structure. The functions are defined as:

$$f(x, y, z) = x^2 - y^2 + r_1 x + r_2 y + \sinh(z), \quad (2.5)$$

$$g(x, y, z) = 2xy + r_3 x + r_4(y + \cosh(z)), \quad (2.6)$$

where the coefficients (r_1, r_2, r_3, r_4) act as control parameters. Depending on the active branch of the mapping, these parameters are selected from the following set:

$$(r_1, r_2, r_3, r_4) \in \{(a, 2a, a, 2a), (b, 2b, b, 2b)\}. \quad (2.7)$$

The primary parameters of the system are determined by the initial conditions as follows:

$$a = \frac{x_0 + y_0}{2}, \quad b = \frac{y_0 + z_0}{2}, \quad (2.8)$$

where x_0, y_0 , and z_0 denote the initial values of the system, playing a decisive role in the overall mapping behavior. To further enrich the dynamics and enhance the nonlinearity within the three-dimensional mapping, two auxiliary maps, namely the Gingerbread map and the Duffing map, are incorporated. These auxiliary maps are generally expressed as $gin(x, y; \alpha_g, \beta_g)$ and $duf(x, y; \alpha_d, \beta_d)$, respectively. Here, the coefficients α_g, β_g and α_d, β_d constitute the internal parameters of the Gingerbread and Duffing maps, respectively, playing a crucial role in shaping the nonlinear and oscillatory behavior of the system. For the Gingerbread map, the parameters α_g and β_g control its nonlinear nature and the intensity of its dynamical complexity. These parameters are tuned based on the primary parameters a and b according to (2.8). This direct dependence of α_g and β_g on the primary parameters ensures that the nonlinearity of the Gingerbread map is explicitly influenced by the initial conditions, thereby intensifying the system's sensitivity to initial states.

Similarly, in the Duffing map, the parameters α_d and β_d determine its oscillatory and damped behavior. These parameters are also defined as functions of a and b via (2.8) to establish a coherent link between the oscillatory structure and the initial conditions. Consequently, any variation in the coefficients—stemming from changes in the initial values—directly impacts the oscillatory structure and damping characteristics. In conclusion, by defining the coefficients of the Gingerbread and Duffing maps in terms of the primary parameters a and b (which are themselves extracted from the initial conditions), the proposed three-dimensional mapping exhibits a structure that is highly dependent on initial conditions. The integration of the main maps f and g with these two auxiliary maps, combined with the dynamic selection of coefficients, leads to the emergence of a chaotic system characterized by intense sensitivity

to initial conditions and complex dynamical behavior in the three-dimensional space. Some of the characteristics of the suggested chaotic map are presented in the following plots in Figure 2-5. Lyapunov exponent is a quantity that indicates the sensitivity of chaotic maps to the initial value and is suitable for examining the degree of separation of infinitely relative paths, so they are important for studying the behavior of chaotic systems. One of the proposed methods for calculating Lyapunov's exponent is the method presented in [30], which is applied, and the result is depicted in Figure 2. This Figure illustrates the Lyapunov exponent with the initial condition $(x_0, y_0, z_0) = (0.55, 0.12, 0.04)$. In steganography, for the original image information to withstand statistical attacks, the stego image must bear no statistical resemblance to the secret image, and the stego image histogram must be uniform as security increases, and the attacker has difficulty extracting hidden information from the stego image. The histogram plot is shown in Figure 3. The histogram plot is flat enough. Because one of the visual designs for the qualitative behavior of chaos is a cobweb, we investigated this scheme and the result is shown in Figure 4. A cobweb plot for 500 points obtained from 3DS is shown, which denotes chaotic behavior. Regarding the bifurcation diagram that is given in Figure 5, we can observe complex 3D behavior that leads to the generation of secure algorithm.

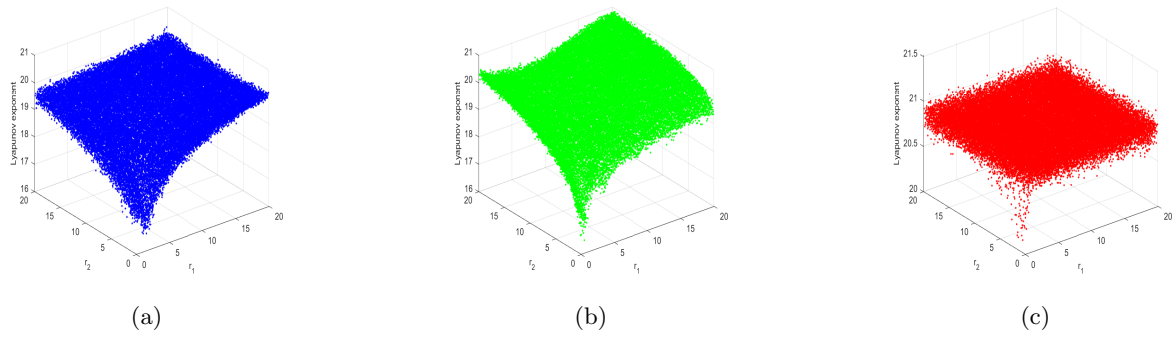


Figure 2: Lyapunov exponent diagrams of 3DS.

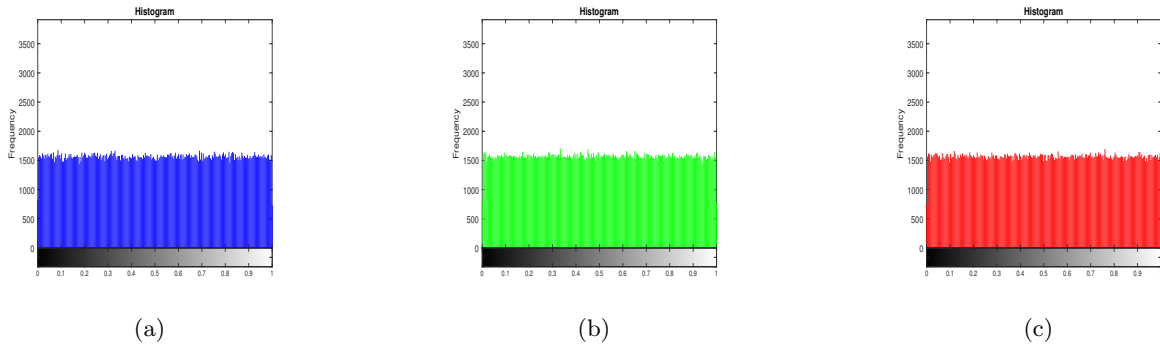


Figure 3: Histogram diagrams of 3DS.

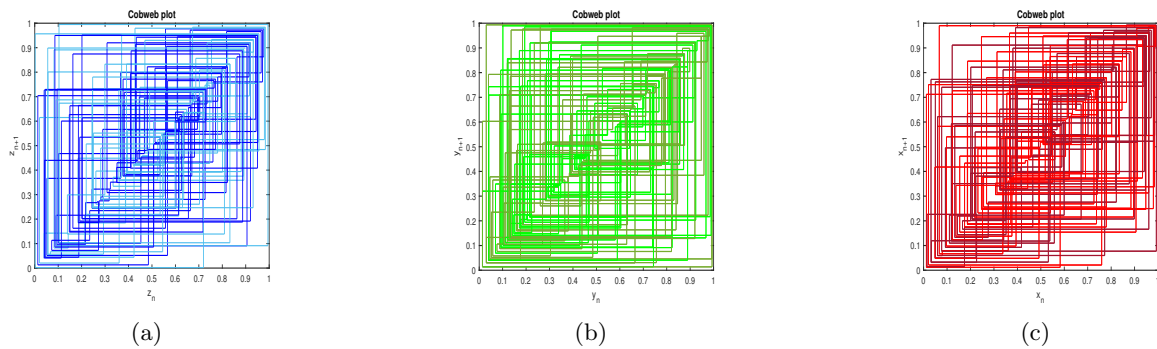


Figure 4: Cobweb plots of 3DS.

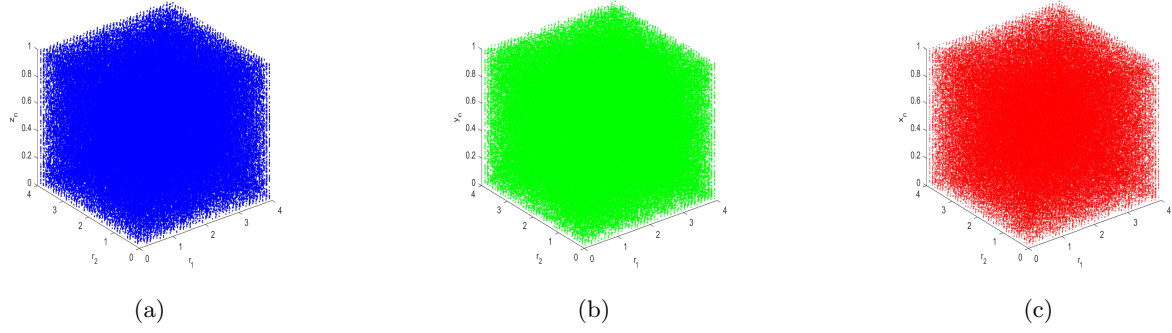


Figure 5: Bifurcation diagrams of 3DS.

3 Shuffle operator

In this section, we introduce a new shuffle operator called the snake shuffle named Shf. This operator shifts the pixels of an image in a way that resembles the movements of a snake, hence its name. As color images are made up of three channels - red, green, and blue (RGB) - the shuffle process is applied to each channel separately, causing the location of the pixels to change. Here, m is a positive scalar and A, B, C are $n \times n$ matrices where n is even. The start point of the snake shuffle operator is $A(\frac{n}{2}, \frac{n}{2} + 1)$ and it rotates counterclockwise two times and then enters $B(1, 1)$ and rotates two times around the matrix clockwise, next enters $C(\frac{n}{2}, \frac{n}{2} + 1)$ and like the first matrix, it rotates counterclockwise two times, then enters the pixel $A(\frac{n}{2} - 2, \frac{n}{2} + 2)$ and repeats this process until it has gone through all pixels. Note that the shuffling of pixels occurs in the number of m each time. Figure 6 illustrates the pixel movement process. Figure 7 demonstrate the variations in the visual representation of images for values of m equal to 250, 500, and 1000. To better understand, we give a simple example by considering matrices of size 4×4 in Figure 8.

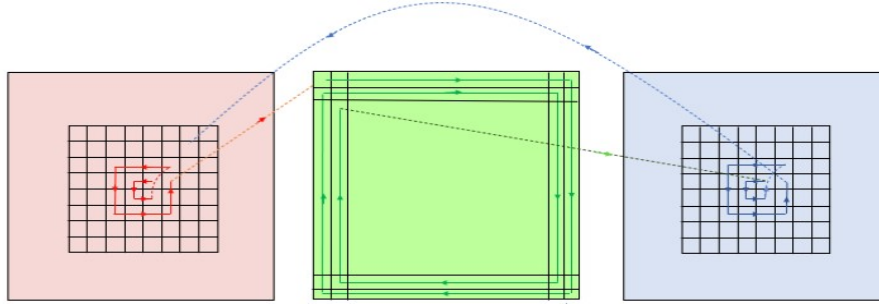


Figure 6: Snake shuffle.

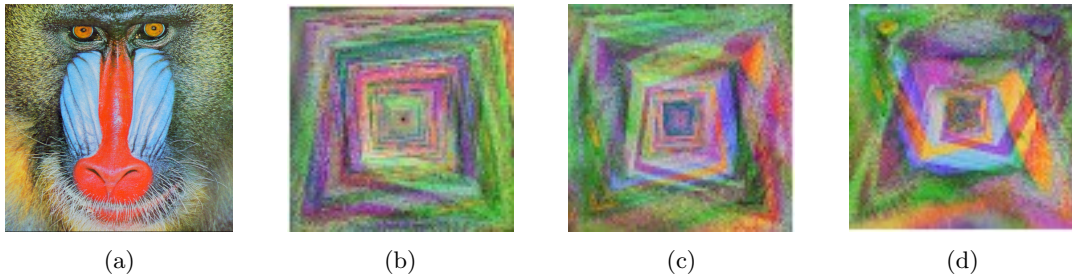


Figure 7: Output images after applying snake shuffle operator (a) original image, (b) after 1000 shifts, (c) after 500 shifts, and (d) after 250 shifts.

4 Steganography

In this section, we outline our proposed image steganography algorithm, detailing the process of embedding a secret image within a cover image and subsequently extracting it from the stego image. In this method, 3D hybrid chaos

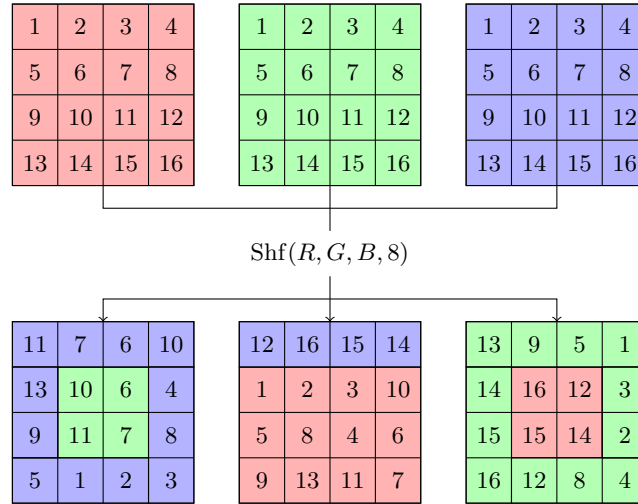


Figure 8: Example of snake shuffle.

function, shuffle operator, framelet, and SVD analysis are used, and we used fractal images as a cover image. Two images of Julia and Mandelbrot were selected from the fractal collection. Fractal images are sensitive to minor changes in their productive parameters. These parameters are a series key and are required to recover a fractal image sent to the recipient for extracting a secret image. The message receiver can recover this image without having a cover image. Fractal images are generated through Ultra Fractal 6.05, incorporating parameters like seed value, location, exponent, color space, bailout value, and rotation angle. In this algorithm, inputs serve as the keys for the chaos function, and the rate of pixel alteration in the cover image is determined by the chaos function's output. The embedding and extraction method is shown in Figure 9.

4.1 Embedding algorithm

The following steps are done to embed a secret image in a fractal cover image:

1. Generate a fractal cover image C of size 512×512 using primary parameters obtained from the key of the algorithm. Additionally, read an RGB secret image S with a size of 512×512 .
2. Perform the proposed chaotic system using inputs obtained from the key to generate relevant outputs for the introduced shuffle operator to scramble pixels of C . We show channels of the resulting image by C_k , $k = r, g, b$.
3. Apply framelet transform on all C_k and S_k and select LL_c^k and LL_s^k sub-bands, where S_k , $k = r, g, b$, are channels of S .
4. Apply SVD decomposition on LL_c^k to obtain the singular coefficients U_c^k , S_c^k , and V_c^k , $k = r, g, b$.
5. Apply SVD decomposition on LL_s^k to obtain the singular coefficients U_s^k , S_s^k , and V_s^k , $k = r, g, b$, which left-singular and right-singular vectors are needed as keys to extract the secret image.
6. Use embedding factor α to achieve the singular coefficients of the stego image ST as (4.1) .

$$S_{st}^k = S_c^k + \alpha * S_s^k. \quad (4.1)$$

7. Perform the inverse of SVD on $[U_c^k, S_{st}^k, V_c^k]$ to get the LL_{st}^k .
8. Apply Inverse of the framelet transform and proposed the shuffle function to generate the ST .

4.2 Extracting algorithm

1. Generate a reconstructed cover image C' of size 512×512 by received keys and the Ultra Fractal 6.05 and read the stego image ST .
2. Perform the proposed chaotic system using inputs obtained from the key to generate relevant outputs for the introduced shuffle operator to scramble pixels of C' and ST where channels of the resulting image are C'_k , and ST_k , $k = r, g, b$.
3. Apply framelet transform on all C'_k and ST_k and select $LL_{c'}^k$ and LL_{st}^k sub-bands.

4. Apply SVD decomposition on $LL_{c'}^k$ to obtain the singular coefficients $U_{c'}^k$, $S_{c'}^k$, and $V_{c'}^k$, $k = r, g, b$.
5. Apply SVD decomposition on LL_{st}^k to obtain the singular coefficients U_{st}^k , S_{st}^k , and V_{st}^k , $k = r, g, b$.
6. Use the factor α to obtain the singular coefficients of the extracted image as (4.2).

$$S_{ex}^k = (S_{st}^k - S_{c'}^k)/\alpha. \quad (4.2)$$

7. Perform the inverse of SVD on $[U_s^k, S_{ex}^k, V_s^k]$ to get the LL_{ex}^k .
8. Apply Inverse of the framelet transform to generate the extracted image.

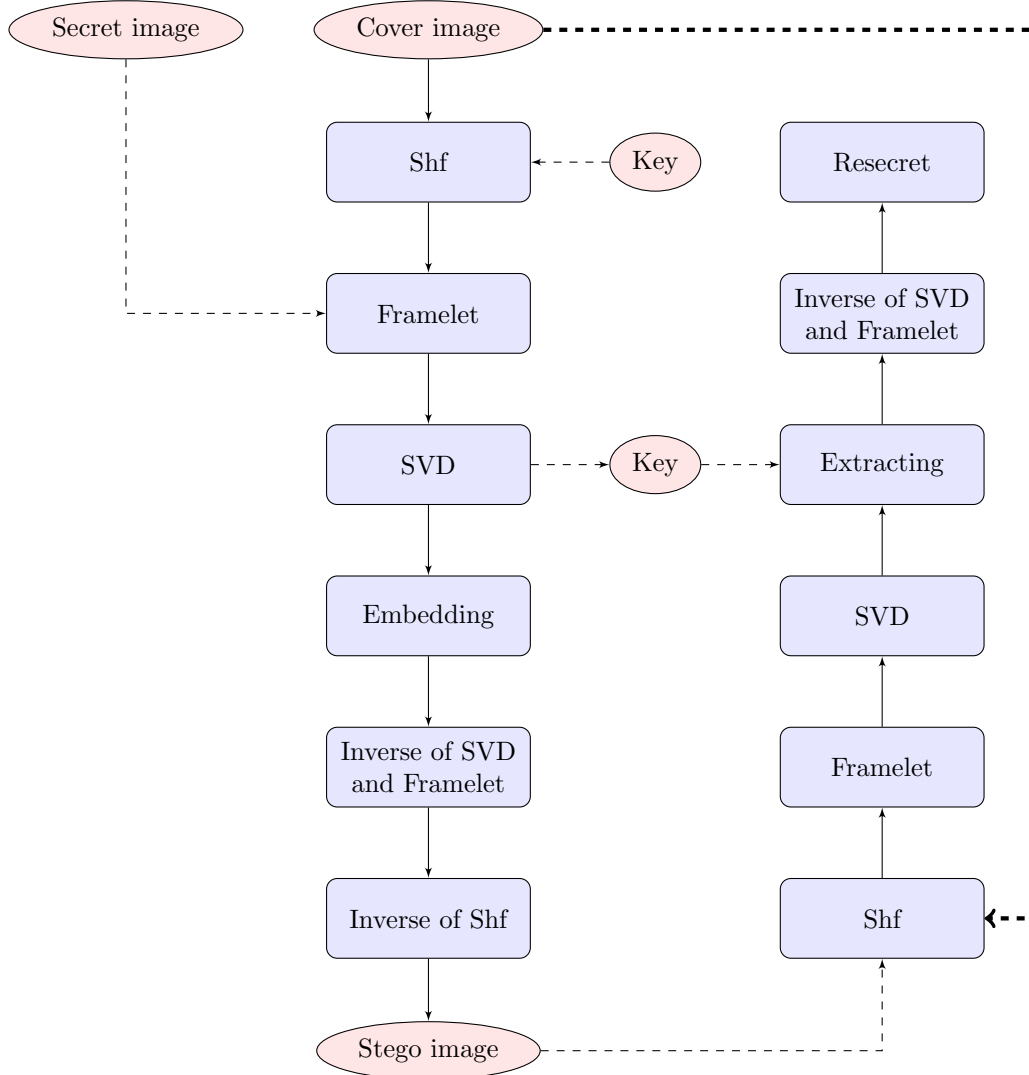


Figure 9: Flowchart of proposed steganography algorithm.

In this method, a technique known as image shuffling is utilized, which plays an important role in the security and quality of the extracted image. The degree of shuffle applied to images is dependent on the output of a chaotic function. The chaotic function exhibits extreme sensitivity to initial conditions; even minute alterations in the input can significantly impact the output. The input to this function, which acts as the key, means that any alteration in the input or key can lead to completely different outputs of the chaotic function and, ultimately, significant changes in the shuffling of images, affecting their quality and recognizability. Thus, the chaotic function directly and profoundly influences the degree of shuffling and, consequently, the extracted images. Without access to the correct key, it is likely that the extracted image will be adversely affected by an inappropriate shuffling, resulting in the production of unintelligible and unrecognizable images. At best, the extracted image may have low quality and be unacceptable, highlighting a vulnerability in the system's inability to accurately recognize data. For this reason, the comparison

between the extracted images with the correct key and appropriate shuffling and those extracted without a key (or with an incorrect key) clearly demonstrates the importance of the key and the chaotic function, as well as the necessity of security in the data extraction process. In Figure 10, a secret image extracted from the stego image is presented, where a minor alteration in the input of the chaotic function leads to a lack of correct shuffling information for the extractor, resulting in the creation of a significantly degraded image.



Figure 10: (a) Stego image, (b) Extracted image with the correct chaotic function key, (c) Extracted image with a slight change in the chaotic function key

5 Evaluation Criteria

In this section, we introduce the necessary metrics for checking the quality and evaluating the similarity between stego and cover images, secret and extracted images.

5.1 Mean Squared Error (MSE)

Mean Squared Error (MSE) is a critical measure for evaluating image quality, quantifying the average of the squares of the differences between actual and estimated values. The closer the value of this criterion is to zero, the better. MSE measures the mean squared errors and has units of squared quantitative measurement, calculated as a variance. An evaluator refers to a technique used to quantify a not directly observable attribute within an image. Error is the difference between the evaluator and the evaluated result. It is a risk function that considers the expected value of the squared error [18]. MSE between two images, such as $f(m,n)$ and $g(m,n)$, is defined as (5.1).

$$MSE = \frac{1}{MN} \sum_{m=0}^N \sum_{n=1}^M [g(m,n) - f(m,n)]^2. \quad (5.1)$$

5.2 Peak signal to noise ratio (PSNR)

The PSNR metric computes the peak signal-to-noise ratio between two images, usually expressed in logarithmic scale as decibels. MSE measures the average of the squares of the differences between the original and the compressed images. We utilize MSE to compute PSNR, as it represents the peak error measure. A lower MSE value and higher PSNR value indicate less error and better quality in the reconstructed image. The selected signal-to-noise ratio is defined as (5.2).

$$PSNR = 10 \log_{10} \frac{255}{MSE}. \quad (5.2)$$

5.3 Structural similarity index (SSIM)

The SSIM method is a perception-based model that assesses image modifications as changes in structural information, accounting for aspects like luminance masking and contrast masking. The Structural Similarity Index (SSIM) is an advanced metric designed to evaluate the visual effects of three key image attributes: luminance, contrast, and structural integrity. By quantitatively assessing these aspects, SSIM facilitates the calculation of the similarity between a pair of images. A score approaching 1 within this index indicates a strong resemblance between the reconstructed and the original images, signifying a superior fidelity in image reconstruction and a high degree of visual similarity. The structural similarity index method can be expressed as (5.3)

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2\mu_y^2 + c_1)(\sigma_x^2\sigma_y^2 + c_2)}, \quad (5.3)$$

where μ_x is the pixel sample mean of x , μ_y is the pixel sample mean of y , σ_x^2 the variance of x , σ_y^2 the variance of y , σ_{xy} the covariance of x and y , $c_1 = (0.01 \times L_1^2)$, $c_2 = (0.03 \times L_1^2)$, and L is dynamic range of the pixel-values.

6 Proposed algorithm experimental results

Several algorithms have been proposed in the past for image steganography, each having its own set of advantages and disadvantages. Continuous research and development have led to significant advancements in the field of steganography. To enhance the performance of the proposed algorithms for embedding and retrieving secret images within stego images, it is essential to focus on the quality of the images, safeguard the confidentiality of the secret image, and balance the capacity and integrity of the cover image. In this section, we will examine the experimental results obtained and present the quality and security of the proposed algorithm using defined metrics. The proposed algorithm was implemented using Matlab R2021b on an Intel Core i5 processor with 8 GB RAM. The cover images used were produced by Ultra Fractal 6.05 program with a size of 512×512 , while the three standard secret images (Baboon, Lena, and Peppers) with a size of 512×512 were taken from USC-SIPI image database [26]. Figure 11 and Figure 12 display the cover and secret images used in this study. The stego images obtained are shown in Figure 13 (a)-(b). These images have high visual quality and are not significantly different from their corresponding cover images. The level of inconspicuousness of the stego image is greater when the image has good quality, with a PSNR value of more than 30 dB and an SSIM value close to 1. Table 1 presents the results related to the quality of stego images compared to cover images using PSNR and SSIM metrics, which were obtained with different alpha values and Table 2 present PSNR value of extracted images.

In this study, various types of cover images, excluding fractal images, were tested using the USC-SIPI image database to evaluate the proposed method's efficiency. In these experiments, the cover image was directly presented to the recipient. This approach yielded promising results, as detailed in Table 2 and Figure 14, which showcase the high quality of the evaluated non-fractal images. A significant advantage of using fractal images is that the extractor does not require access to the original cover image to retrieve the secret image; instead, the cover image can be accurately reconstructed using the necessary parameters as a key. This feature provides a crucial security advantage, offering greater security and flexibility compared to traditional cover images. Reducing the dependency on the original cover image minimizes the risk of unauthorized individuals identifying the hidden information. Figure 13 (c)-(h) displays the secret images extracted from two stego images. Furthermore, we compare the histogram of cover and stego images. An image histogram is a two-dimensional graph that indicates the frequency of brightness levels of image pixels. The x-axis specifies the range of brightness changes, and the y-axis defines the frequency of brightness levels. The histogram provides valuable information about an image such as contrast and brightness, which can be used in image processing. In Figure 15, we show the histograms for cover images and their corresponding stegos along with different secret images. The proposed method's cover diagram and stego are very similar to each other, making it difficult to recognize the stego image using this algorithm. The Chi-squared distance between the histograms was calculated, yielding a value of 2×10^{-10} . This extremely small value indicates a high degree of similarity between the histograms of the cover and stego images. A low Chi-squared distance suggests that the embedding process has minimally altered the statistical distribution of pixel values, making the presence of hidden data difficult to detect through histogram analysis. In simpler terms, this means the stego image's pixel distribution is almost identical to the original cover image, indicating good security against basic steganalysis techniques that rely on histogram comparison. RS analysis is an advanced statistical technique designed to detect the presence of steganography in digital images. Steganography refers to the practice of hiding information within the data of an image, facilitating the secret and undetectable transmission of that information. This method examines statistical variations in the pixels of an image to uncover the effects caused by hidden data. Typically, RS analysis begins with changes in the least significant bits of pixels. Although these changes may have a negligible effect on the visual appearance of the image, they can provide significant clues regarding the presence of concealed information. To evaluate this issue, two functions R and S are computed, which represent the smoothness and roughness of the image, respectively. The R function typically reflects the level of homogeneity within the image, while the S function indicates the degree of roughness and variations across the image surface. In this study, RS analysis was used to examine the steganographic images obtained. The results of the analysis indicate a high level of security in the stego images, suggesting that the computed values demonstrate that no concealed image has been identified among the stego images. Figure 16 presents the results of R-S analysis of stego images.

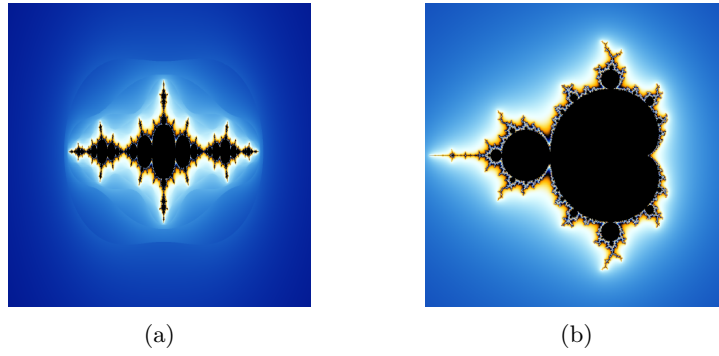


Figure 11: images: (a) Julia, (b) Mandelbrot.



Figure 12: Secret images: (a) Baboon, (b) Lena, and (c) Peppers.

Table 1: PSNR and SSIM values of a proposed steganography algorithm

Cover Image	Secret Image	alpha	PSNR	SSIM
Julia	Lena	0.1	25.52	0.7258
		0.01	45.09	0.9930
		0.0015	74.97	1
		0.001	104.07	1
Julia	Peppers	0.1	26.45	0.7428
		0.01	45.86	0.9931
		0.0015	70.73	1
		0.001	104.07	1
Julia	Baboon	0.1	25.84	0.7523
		0.01	45.32	0.9928
		0.0015	75.36	1
		0.001	104.07	1
Mandelbrot	Lena	0.1	25.96	0.7285
		0.01	44.95	0.9850
		0.0015	104.07	1
		0.001	104.07	1
Mandelbrot	Peppers	0.1	26.71	0.7153
		0.01	46.10	0.9852
		0.0015	77.15	1
		0.001	104.07	1
Mandelbrot	Baboon	0.1	26.05	0.7881
		0.01	45.70	0.9818
		0.0015	104.07	1
		0.001	104.07	1

Table 2: PSNR values of the proposed steganography algorithm.

Cover Image	Secret Image	α	PSNR (Stego-Cover)	PSNR (Extracted-Secret)
Julia	Lena	0.1	25.52	29.45
		0.01	45.09	29.45
		0.0015	74.97	29.45
		0.001	104.07	29.45

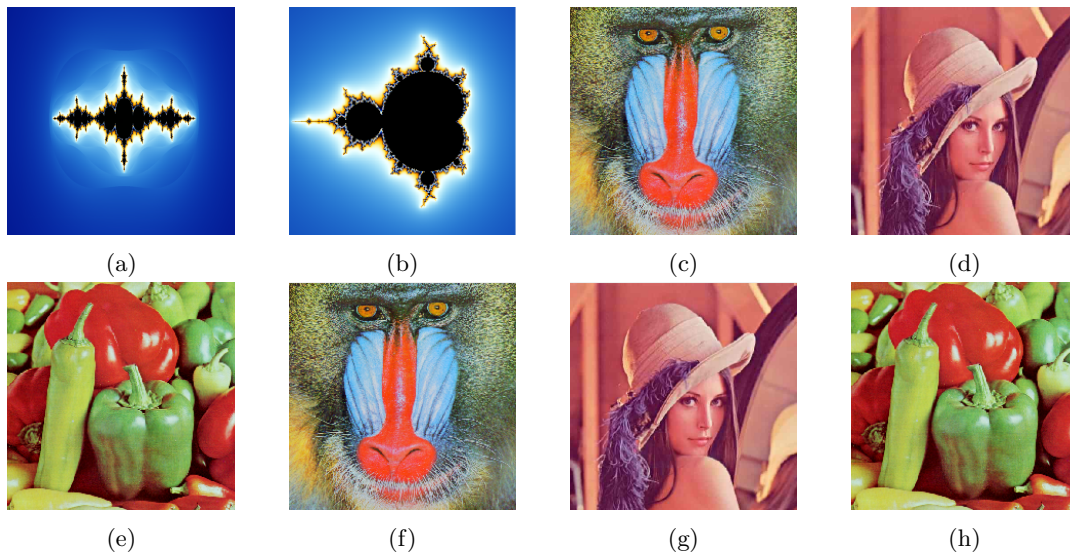


Figure 13: (a)-(b) Stego images, (c)-(e) extracted images from Julia stego, and (f)-(h) extracted images from Mandelbrot stego.

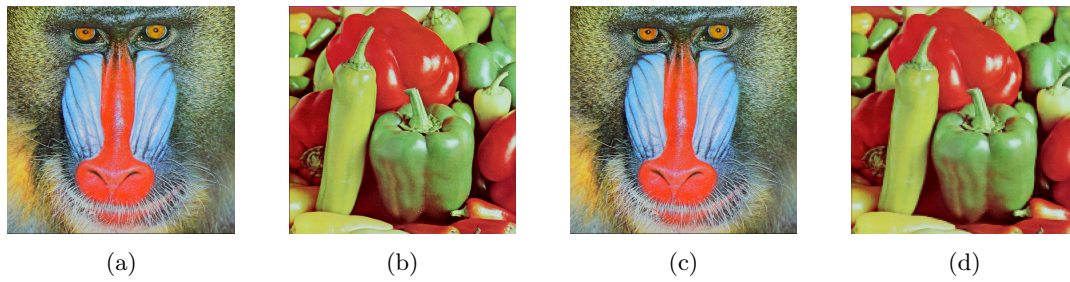


Figure 14: (a) Non-fractal cover image, (b) Secret image, (c) Stego image, and (d) Extracted image

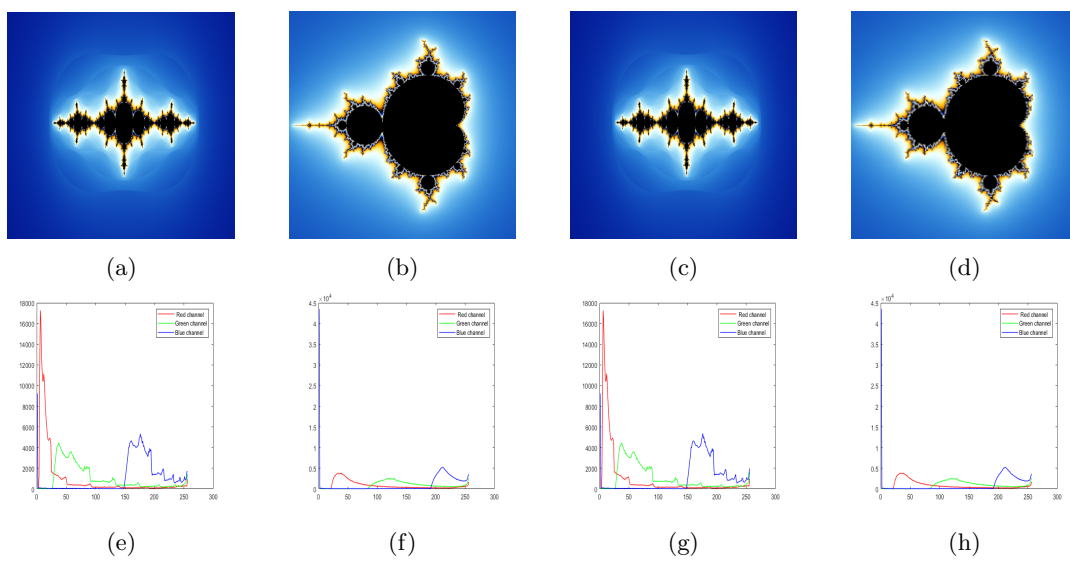


Figure 15: (a)-(b) Cover images, (c)-(d) stego images, and (e)-(h) histogram of cover and stego images.

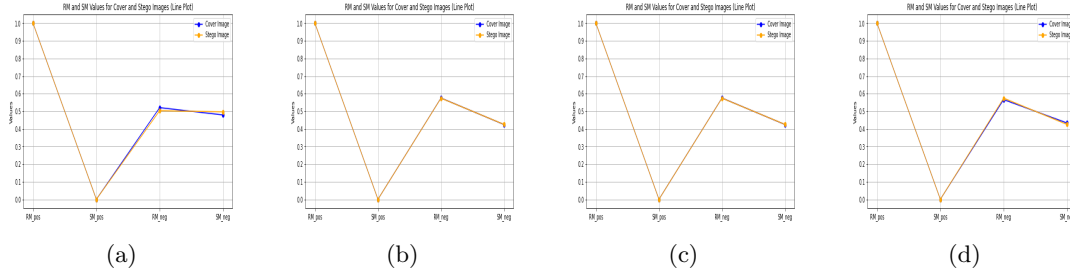


Figure 16: (a)-(d) R-S analysis for several different stego images.

7 Comparative analysis of the proposed algorithm's performance against existing methods

In this section, we compare the performance of the proposed method with several existing steganography algorithms. As shown in Table 3 and 2, PSNR and SSIM values for the proposed method are significantly higher than those of other methods. The proposed method utilizes a three-dimensional chaos function and a shift function, which enhance its security. Additionally, the extractor's capability to reconstruct the cover image using a key is a major advantage of this method. Furthermore, the quality of the stego images is very high due to the techniques employed for information hiding and optimization in cover image reconstruction. Complete reconstruction, scattering, and stability are three essential features of framelets. Framelets are particularly effective in overcoming the limitations of traditional wavelet transforms. In wavelets, small changes in the input signal can lead to significant changes in the energy distribution of the wavelet coefficients across different scales, resulting in instability in signal analysis. Moreover, the Discrete Wavelet Transform has poor directional sensitivity for diagonal features, which can hinder the accurate representation of certain characteristics of the signal. Framelets address these shortcomings by providing improved directional sensitivity and stability against variations, making them more suitable for applications in image processing, signal reconstruction, and more effective analysis. The proposed method executes at a high speed, indicating its high efficiency. However, it is important to note that the actual execution speed may depend on various factors such as the software and hardware used. Choosing optimized software for complex computations and data manipulation, along with hardware specifications like processor speed, memory capacity, and graphical capabilities, all have a significant impact on execution speed. Additionally, the process of generating a fractal image with desired parameters and reconstructing it using key parameters requires time. Table 4 presents the execution times of the proposed method, providing a clearer comparison with other methods. While embedding color images within a 3D context presents greater complexity compared to gray-scale images, the achieved outcomes are notably satisfactory. This methodology significantly surpasses contemporary LSB techniques by producing stego images of superior quality. The robust data embedding process effectively minimizes image distortion while preserving visual clarity. Consequently, our approach yields enhanced results in terms of both imperceptibility and overall image quality, thereby rendering it more efficient for data concealment. The high caliber of the resultant stego images guarantees that the embedded data remains undetectable, thus augmenting both the security and practical applicability of the method. This provides an effective means for inconspicuous data transmission. This makes the technique a strong choice for applications where both image quality and data stealth are of paramount importance, ensuring that sensitive information can be transmitted securely without compromising the visual integrity of the cover image.

8 Stability against attacks

One of the factors that puts secret images at risk is the possibility of attacks by hackers attempting to destroy them. These attacks can prevent the secret images from reaching their intended destination or arriving with reduced quality, making restoration impossible [27]. In this article, we conducted various attacks on fractal steganographic images to evaluate the resilience of hidden images against them. These attacks included left upper cropping (150×150 by white), salt and pepper noise with density 0.01, and tree fractal attacks. Figure 17 shows stego images with different alpha values and Figure 18 shows attacked images, and extracted secret images from this images. Figures 18(a)-(f), show stego images after a upper cropping and extracted secret images. The extracted images can be identified, and these attacks could not prevent the secret image from reaching its destination. Figures 18(g)-(l) show salt and pepper noise applied to stego images and extracted images. These stego images have 0.01 noise, and the extracted images are recognizable and in 18(m)-(r), show the tree fractal crop applied to stego images. Although a fractal tree attack has

Table 3: Comparison of the proposed algorithm with some steganography algorithms.

Algorithm	Secret image	Cover image	PSNR	SSIM
Proposed (512 × 512)	Lena (424 KB)	Julia (196 KB)	74.97	0.9999
	Baboon (769 KB)	Julia (196 KB)	75.36	1
	Peppers (769 KB)	Julia (196 KB)	70.73	1
	Lena (424 KB)	Mandelbrot (231 KB)	104.07	1
	Baboon (769 KB)	Mandelbrot (231 KB)	104.07	1
	Peppers (769 KB)	Mandelbrot (231 KB)	77.15	1
[4] (512 × 512)	SVD-DWT			
	Baboon (833 KB)	Julia ₁ (62.5 KB)	61.82	0.9130
	Peppers (426 KB)	Julia ₂ (38.6 KB)	64.17	0.9716
	Couple (151 KB)	Mandelbrot ₁ (34.8 KB)	68.60	0.9383
	Lena (783 KB)	Mandelbrot ₂ (27.1 KB)	63.43	0.9724
	SVD-IWT			
	Baboon (833 KB)	Julia ₁ (62.5 KB)	63.03	0.981
	Peppers (426 KB)	Julia ₂ (38.6 KB)	65.28	0.980
	Couple (151 KB)	Mandelbrot ₁ (34.8 KB)	70.48	0.993
	Lena (783 KB)	Mandelbrot ₂ (27.1 KB)	65.03	0.941
[22] (512 × 512)	Cameraman (256 KB)	Lena-gray	69.44	0.9999
	Cameraman (256 KB)	Peppers-gray	69.53	0.9999
	Cameraman (256 KB)	Airplane-gray	69.61	0.9999
	Cameraman (256 KB)	Splash-gray	69.59	0.9999
[3] (512 × 512)	Image-1	Baboon	55.31	0.9992
	Image-2	Baboon	55.94	0.9983
	Image-3	Baboon	55.51	0.9984
	Image-4	Baboon	55.90	0.9984
	Image-5	Baboon	56.64	0.9982
[24]	Lena (512 × 512)	Gantrycrane (400 × 256)	55.74	-
	Kids (318 × 400)	Onion (198 × 135)	61.21	-
	Fruits (512 × 512)	Peppers (512 × 384)	62.52	-
	Baboon (512 × 512)	Trees (350 × 258)	64.28	-
	Peppers (512 × 384)	Onion (198 × 135)	66.50	-
[9] (128 × 128)	Baboon-gray	Lena-gray	60.15	-
	Baboon-gray	Peppers-gray	60.2	-
	Baboon-gray	Boat-gray	63.08	-

Table 4: Comparison of execution time

Techniques Used	Average Execution Time (in Seconds)
[4] DWT-SVD	1.83
[4] IWT-SVD	1.48
[3] BESOPS-CE	23.96
Proposed Technique	1.23

dramatically affected the quality and clarity of the secret image, it could not prevent the extraction and recognition of the secret image. Note that the alpha coefficient plays a crucial role in determining the quality of the secret image that is extracted. Increasing the value of the alpha coefficient enhances the resistance of the secret image against various attacks. However, it is important to ensure that this increase does not compromise the quality of the stego image.

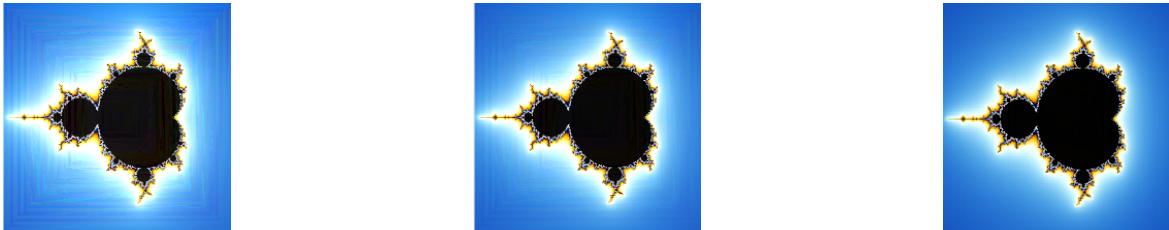


Figure 17: Stego images (alpha = 0.2, 0.1 and 0.02)

9 Conclusion

The proposed steganographic algorithm utilizes framelet transform and SVD decomposition to enhance security. Furthermore, a new combination of 3D chaos function and shuffle operator has been incorporated to improve resistance

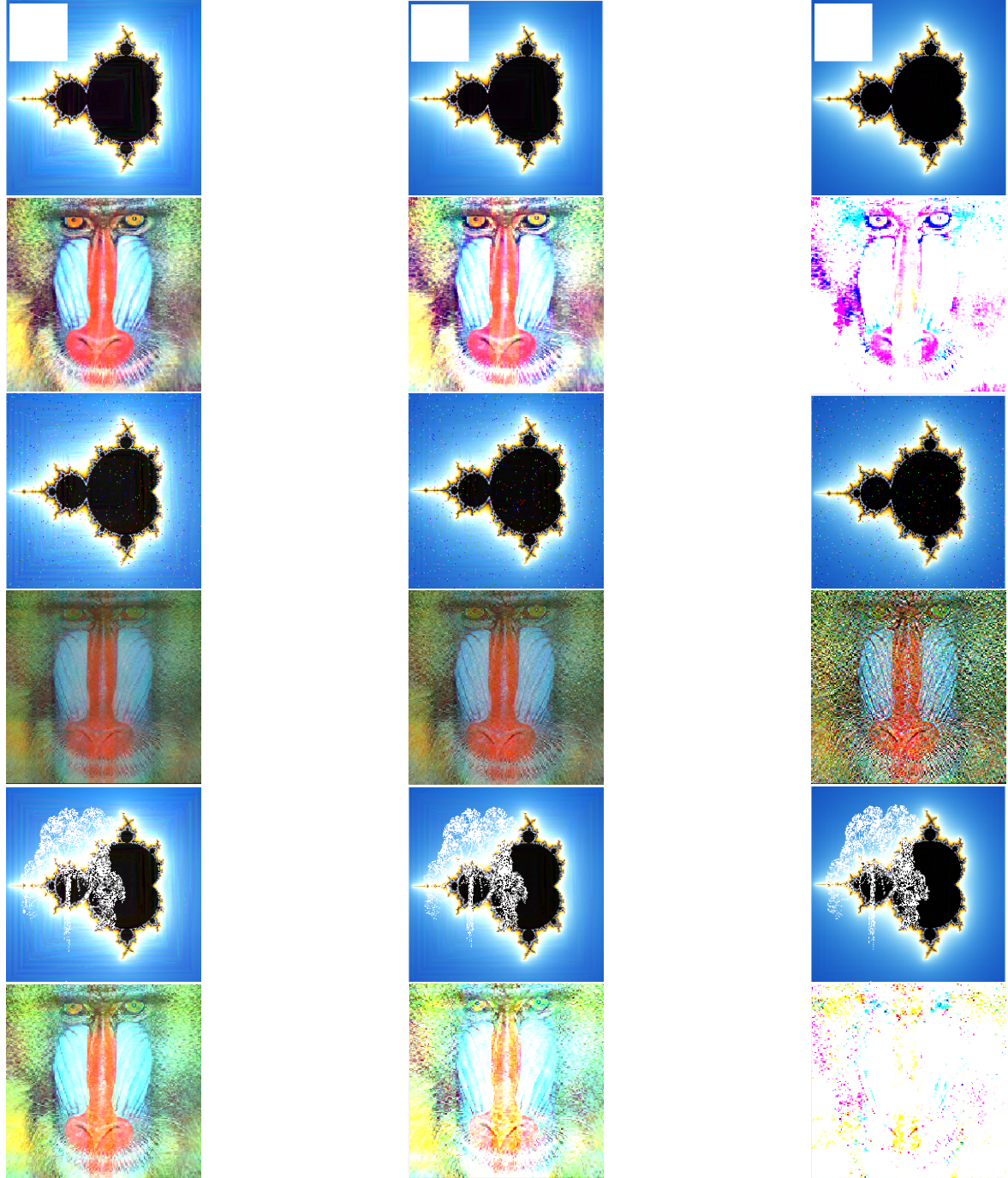


Figure 18: (a)-(c) Attacked image (cropping 150×150 by white), (d)-(f) extracted secret images, (g)-(i) attacked images (salt and pepper noise with variance 0.01), (j)-(l) extracted secret images. (m)-(o) attacked image (tree fractal) (p)-(r) extracted secret images.

and safety. To evaluate the level of protection of the secret image against potential attacks, we conducted several types of attacks and obtained satisfactory results. The proposed method demonstrates appropriate values of PSNR and SSIM, and its use of the defined shuffle and chaos function ensures superior security and robustness compared to similar methods.

References

- [1] O. Abd Qasim and S. Golshannavaz, *Data protection enhancement in smart grid communication: An efficient multi-layer encrypting approach based on chaotic techniques and steganography*, e-Prime Adv. Electr. Eng. Electron. Energy **10** (2024), 100834.
- [2] H. N. Al-Taai, *A novel fast computing method for framelet coefficients*, Am. J. Appl. Sci. **5** (2008), no. 11, 1522–1527.
- [3] A. A. Bahaddad, K. A. Almarhabi, and S. Abdel-Khalek, *Image steganography technique based on bald eagle search optimal pixel selection with chaotic encryption*, Alex. Eng. J. **75** (2023), 41–54.
- [4] A. Durafe and V. Patidar, *Development & analysis of IWT-SVD and DWT-SVD steganography using fractal cover*, J. King Saud Univ. Comput. Inf. Sci. **34** (2022), no. 7, 4483–4498.
- [5] A. Fahim and Y. Raslan, *Optimized steganography techniques based on PVDs and genetic algorithm*, Alex. Eng. J. **85** (2023), 245–260.
- [6] A. Gaffar, A. Joshi, and D. Kumar, *Image encryption using gingerbreadman map and RC4A stream cipher*, Appl. Appl. Math. **15** (2020), no. 2, 894–908.
- [7] X. Gao, *Image encryption algorithm based on 2D hyperchaotic map*, Opt. Laser Technol. **142** (2021), 107252.
- [8] A. Ibaida, D. Al-Shammary, and I. Khalil, *Cloud enabled fractal based ECG compression in wireless body sensor networks*, Future Gener. Comput. Syst. **35** (2014), 91–101.
- [9] A. Jaradat, E. Taqieddin, and M. Mowafi, *A high-capacity image steganography method using chaotic particle swarm optimization*, Secur. Commun. Netw. **2021** (2021), 6679284.
- [10] Z. K. J. Jasim and S. Kurnaz, *An improved image steganography security and capacity using ant colony algorithm optimization*, Comput. Mater. Contin. **80** (2024), no. 3, 16459–16475.
- [11] H. Kavak, J. J. Padilla, D. Vernon-Bido, S. Y. Diallo, R. Gore, and S. Shetty, *Simulation for cybersecurity: State of the art and future directions*, J. Cybersecur. **7** (2021), no. 1, tyab005.
- [12] N. Khalil, A. Sarhan, and M. A. Alshewimy, *A secure image steganography based on LSB technique and 2D chaotic maps*, Comput. Electr. Eng. **119** (2024), 109566.
- [13] Y. Khedmati, R. Parvaz, and Y. Behroo, *2D hybrid chaos map for image security transform based on framelet and cellular automata*, Inform. Sci. **512** (2020), 855–879.
- [14] R. Panigrahi and N. Padhy, *An effective steganographic technique for hiding the image data using the LSB technique*, Cyber Secur. Appl. **3** (2025), 100069.
- [15] B. Patwari, U. Nandi, S. Dey, S. Dhar, S. K. Ghosal, and T. Si, *Blockchain-integrated image steganography using XDoG edge detection for authentication*, Cyber Secur. Appl. **2** (2025), 100091.
- [16] J. G. Rajan and R. S. Ganesh, *Dynamic pixel shuffling and hash LSB steganography with RC4 encryption: A robust data security framework*, Expert Syst. Appl. **260** (2025), 127403.
- [17] C. Rupa, *A digital image steganography using Sierpinski gasket fractal and PLSB*, J. Inst. Eng. (India) Ser. B **94** (2013), no. 3, 147–151.
- [18] U. Sara, M. Akter, and M. S. Uddin, *Image quality assessment through FSIM, SSIM, MSE and PSNR – a comparative study*, J. Comput. Commun. **7** (2019), no. 3, 8–18.
- [19] J. Sharafi, Y. Khedmati, and M. M. Shabani, *Image steganography based on a new hybrid chaos map and discrete transforms*, Optik **226** (2021), 165492.
- [20] S. Singh, R. Parida, and C. Pradhan, *Comparative analysis of image encryption using 2D and 3D variations of Duffing map*, Proc. Int. Conf. Signal Process. Commun. (ICCSP), 2018, pp. 0751–0754.
- [21] F. Slijkerman, *Ultra fractal 5*, 2023.
- [22] M. S. Subhedar and V. H. Mankar, *Secure image steganography using framelet transform and bidiagonal SVD*, Multimed. Tools Appl. **79** (2020), no. 3, 1865–1886.

- [23] Y. Sun, L. Chen, R. Xu, and R. Kong, *An image encryption algorithm utilizing Julia sets and Hilbert curves*, PLoS One **9** (2014), no. 1, e84655.
- [24] S. S. Tevaramani and J. Ravi, *Image steganography performance analysis using discrete wavelet transform and alpha blending for secure communication*, Glob. Transitions Proc. **3** (2022), no. 1, 208–214.
- [25] N. Tsafack, S. Sankar, B. Abd-El-Atty, J. Kengne, K. C. Jithin, A. Belazi, I. Mehmood, A. K. Bashir, O. Y. Song, and A. A. Abd El-Latif, *A new chaotic map with dynamic analysis and encryption application in internet of health things*, IEEE Access **8** (2020), 137731–137744.
- [26] USC-SIPI, *Database*, 1997.
- [27] R. Wang, G. Yang, X. Yan, S. Luo, and Q. Han, *Secret image sharing in the encrypted domain*, J. Vis. Commun. Image Represent. **98** (2024), 104013.
- [28] T. Wu, X. Hu, C. Liu, Y. Wang, and Y. Zhu, *An efficient steganography scheme based on wavelet transformation for side-information estimation*, J. King Saud Univ. Comput. Inf. Sci. **36** (2024), no. 6, 102109.
- [29] S. Yuan, T. Jiang, and Z. Jing, *Bifurcation and chaos in the Tinkerbell map*, Int. J. Bifur. Chaos **21** (2011), no. 11, 3137–3156.
- [30] P. Zhou, X. Hu, Z. Zhu, and J. Ma, *What is the most suitable Lyapunov function?*, Chaos Solitons Fractals **150** (2021), 111154.
- [31] H. Zhu, J. Ge, W. Qi, X. Zhang, and X. Lu, *Dynamic analysis and cryptographic application of a sinusoidal-polynomial composite chaotic system*, Res. Sci. (RS) **1** (2021).