Image steganography using least significant bit and secret map techniques

Ashwak ALabaichi[1], Maisa'a Abid Ali K. Al-Dabbas[2], Adnan Salih[3]
[1]Department of Biomedical Engineering, Engineering College, University of Kerbala, Iraq
[2]Department of Computer Science, University of Technology, Iraq
[3]Department of Computer Science, Science College, University of Kirkuk, Iraq

Article Info

Article history:

Received Aug 6, 2019
Revised Oct 2, 2019
Accepted Oct 11, 2019

Keywords:

3D Chebyshev
3D logistic map
LSB
Secret message
Steganography

ABSTRACT

In steganography, secret data are invisible in cover media, such as text, audio, video and image. Hence, attackers have no knowledge of the original message contained in the media or which algorithm is used to embed or extract such message. Image steganography is a branch of steganography in which secret data are hidden in host images. In this study, image steganography using least significant bit and secret map techniques is performed by applying 3D chaotic maps, namely, 3D Chebyshev and 3D logistic maps, to obtain high security. This technique is based on the concept of performing random insertion and selecting a pixel from a host image. The proposed algorithm is comprehensively evaluated on the basis of different criteria, such as correlation coefficient, information entropy, homogeneity, contrast, image, histogram, key sensitivity, hiding capacity, quality index, mean square error (MSE), peak signal-to-noise ratio (PSNR) and image fidelity. Results show that the proposed algorithm satisfies all the aforementioned criteria and is superior to other previous methods. Hence, it is efficient in hiding secret data and preserving the good visual quality of stego images. The proposed algorithm is resistant to different attacks, such as differential and statistical attacks, and yields good results in terms of key sensitivity, hiding capacity, quality index, MSE, PSNR and image fidelity.

Corresponding Author:

Ashwak ALabaichi,
Department of Biomedical Engineering,
Engineering College,
University of Kerbala, Iraq.
Email: ashwaq.alabaichi@gmail.com

1. INTRODUCTION

The Communication is vital in the modern world. During communication, information is transmitted through different data channels. This process is prone to serious security problems. Increasing attention has thus been paid to the discovery of ways to protect valuable information during its transmission. Cryptography is a technique used to secure communication secrecy, and several methods have been proposed to encrypt and decrypt data and thereby ensure message secrecy. However, maintaining the secrecy of message contents is not always sufficient, hence the use of ciphertext. Ciphertext is easy to notice, but it informs others when communication channels are monitored. Thus, the delivery of secret messages by exchanging plaintext has been widely investigated in the past two decades. Maintaining message secrecy is required and is realised through steganography. Steganography involves hiding information in a way that no information appears to be hidden, whereas cryptography involves encrypting information by using a key and sending this information through a specific channel. A user or process can observe the communication process, but they cannot steal the relevant information unless they possess the key. In steganography, the person or process is unaware of the transmission of secret information. Therefore, no attempt is made to extract information [1-7].

The term steganography is derived from the Greek words stegos, which means cover, and grapha, which means writing. It is defined as covered writing that hides the existence of the actual message. Steganography is used to hide information inside other media. Its two main steps are embedding a secret message inside a cover using a stego key and extracting the secret message from the cover using the stego key. The combination of embedded message and cover creates stego media. Stego keys are utilised to hide and extract secret messages. Only the holders of stego keys can correctly retrieve hidden secret messages. Steganography can be described with the following formula: stego media = cover media + embedded message + stego key.

Steganography is classified into linguistic steganography and technical steganography. Linguistic steganography involves the use of natural language as a carrier for hiding secret data. Technical steganography employs a multimedia carrier. Most digital file formats are characterised by a high degree of redundancy that benefits steganographic techniques. Common steganographic techniques are steganography in texts, images, audio and videos. Among these varieties of file formats, digital images are the most popular because of their frequency on the internet and high capacity for data transmission while minimising image quality degradation [6-10]. Steganographic methods may be in the form of spatial domain embedding or frequency domain embedding. Frequency domain embedding involves the transformation of images into frequency components through discrete cosine transform, fast Fourier transform and discrete wavelet transform (DWT). Messages are embedded at the bit or block level. In spatial domain embedding, information is directly hidden depending on the intensity of pixels. Frequency domain procedures are robust and are commonly used for watermarking, whereas spatial domain methods provide high capacity and are widely used in steganography. Steganography and its usefulness are influenced by three aspects, namely, 1) capacity, which refers to the number of data bits that can be hidden in cover media; 2) visual quality of stego images, which must remain unchanged (imperceptibility); and 3) robustness, which refers to the resistance to modification or destruction [3, 9, 11].

A widely used spatial domain method is the least significant bit (LSB) substitution in which lower order image bits (those that do not possess useful image information) are replaced with secret message bits [9,12]. The use of LSB substitution preserves image quality without entailing complex operations. In this method, the bits of secret data are hidden in the K-LSB plane in each pixel of a cover image. The most widely known LSB methods are LSB matching (LSBM), LSBM revised (LSBMR) and edge adaptive-based LSBMR steganography. However, most of these techniques are most of these techniques are probably easy to be broken. Therefore these methods have undergone improvements in various aspects [1, 2, 7, 8, 10, 13]. In particular, researchers have used chaos theory. Unlike traditional methods, chaotic methods are sensitive to primary conditions and nonperiodic, nonconvergence and controlling parameters. Hence, they have been utilised by many researchers as a vital solution in their work [9]. Although a 1D chaotic system is highly efficient, it has some inherent disadvantages, such as small key assignment and inadequate security that reduces its efficiency and performance.

Numerous systems encompassing one-, two- or higher-dimensional systems with chaotic maps have been introduced in recent years. 3D maps provide higher security and randomness than 1D and 2D maps [14-17]. Chaos-based steganography algorithms have attracted much attention in existing studies because of their efficiency and applicability to steganography for providing secure communication. Bandyopadhyay, Dasgupta, Mandal and Dutta [2] put forward a new approach secure data are built into digital images by using a 1D logistic map. This logistic map is used to encrypt secret messages before embedding. Rajendran and Doraipandian [5] put forward a novel method for hiding secret images using 1D logistic maps. These 1D logistic maps are utilised to generate pseudo random keys. These keys are used to randomly select the pixel positions of cover images for hiding secret images. Sharif, Mollaeefar and Nazari [6] also proposed a novel algorithm for image steganography based on chaos theory. The proposed algorithm involves a novel 3D chaotic map (LCA map) with a maximum Lyapunov exponent of 20.58, which is adopted to generate three chaotic sequences. Mishra, Routray and Kumar [9] proposed the embedding of secret information in a digital image in the spatial domain through LSB and Arnold's transform. Arnold's transform is applied two times in two different phases. Thenmozhi and Chandrasekaran [13] presented a novel technique for image steganography by using a DWT chaotic system. In this method, Henon mapping is applied to secret images, and 2D DWT is performed on cover images. Ghebleh and Kanso [18] developed a new robust chaotic method for digital image steganography, in which a 3D chaotic cat map is used to embed secret messages, lifted DWT is adopted to provide robustness, and Sweldens' lifting is used to ensure integer-to-integer transformation. Bilal, Imtiaz, Abdul, Ghouzali and Asif [19] introduced a zero-steganography algorithm based on chaos theory which embeds data according to the relationship between cover images, chaotic sequences and payloads rather than directly embedding data in cover images. Alam, Kumar, Siddiqui and Ahmad [20] improved a method for image steganography by utilising edge detection and logistic maps as a random generator of secret keys for random LSB substitution. Sabery and

Yaghoobi [21] proposed the use of a simple logistic map to hide secret images in host images. Embedding was performed in the logistic map to determine the blocks of pixels. Roy, Sarkar and Changder [22] presented chaos-based adaptive image steganography, in which the efficiencies of matrix encoding and LSBM are combined for embedding data and chaos is utilised to provide enhanced security. Kanso and Own [23] introduced a digital image steganography method based on Arnold's cat map. Anees, Siddiqui, Ahmed and Hussain [24] proposed a steganographic method in the spatial domain in which chaotic maps are used to resolve pixel positions. Raghava, Kumar, Deep and Chahal [25] proposed the new use of Henon chaotic maps to boost the conventional LSB technique for image steganography.

The current study mainly focuses on made image steganography using LSB techniques is complex and the hidden information is controlled by the secret keys and cannot be retrieved without the same secret keys. A new approach to LSB-based image steganography that uses secret maps is introduced. A secret map is controlled by using secret keys to secure hidden information. Hidden information may be inserted sequentially or randomly. In this study, the hidden information is randomly distributed before hiding it in a cover image to provide better security than sequential methods. The hidden information is permuted by using a 3D Chebyshev map. Insertion is performed through the chaotic sequence generated by the chaotic map. The cover image pixels are randomly selected on the basis of the secret map. The secret map is created by using secret keys that are generated through a 3D logistic map. The hidden information is controlled by the secret keys and cannot be retrieved without the same secret keys. Thus, using secret keys enhances the security of hidden information in LSB-based image steganography.

The rest of the paper is organised as follows. Section 2 presents the chaotic map, its properties and the types used in this study. Section 3 describes the proposed algorithm. Section 4 provides the experimental results of the proposed algorithm. Section 5 discusses the analysis of the proposed algorithm based on several factors. Section 6 presents the conclusions and recommendations for future work.

2. CHAOTIC MAP

Chaos refers to a state of disorder. In the field of mathematics, chaotic behaviour is revealed by maps serving an evolution function. Discrete-time dynamical systems are also referred to as maps. Chaos theory is used to encrypt information, and DWT is used to hide information [5, 13]. This theory centres on system behaviour that is characterised by deterministic laws but shows randomness and unpredictability. That is, a dynamical system depends on its initial conditions with high sensitivity that any slight variation in the initial parameters results in a different chaotic sequence. Chaos is difficult to define comprehensively [2, 15]. The sensitivity of dynamical system is fractal in nature and thus benefits the search for solutions to nonlinear equations. Chaos theory boosts the confidentiality, nonperiodicity, randomness and easy implementation are the main properties that lead to benefit of them in steganography techniques. Chaotic systems have been used in several fields, including nonlinear dynamics that is man-made and natural real systems. Numerous steganographic methods based on chaos theory have been proposed and discussed in the past few decades [2]. In these methods, secret keys are generated using 3D logistic and 3D Chebyshev maps.

2.1. 3D logistic map

A logistic map is a simple chaotic map which belongs to the family of first-order difference equations. It can be mathematically represented as follows:

$$X_{n+1} = RX_n(1 - Xn), \quad (1)$$

where the system parameter is μ ∈ [0,4] and the initial condition is $X_0 \in (0,1)$. A logistic map chaotically behaves with $R \in (3.5699456,4]$ [19, 20]. A 1D logistic map can be extended to the 3D, as defined in (2) to (4).

$$X_{n+1} = RX_n(1 - X_n) + \beta Y_n^2 X_n + \alpha Z_n^2, \quad (2)$$

$$Y_{n+1} = RY_n(1 - Y_n) + \beta Z_n^2 Y_n + \alpha X_n^2, \quad (3)$$

$$Z_{n+1} = RZ_n(1 - Z_n) + \beta X_n^2 Z_n + \alpha Y_n^2. \quad (4)$$

The parameters of a nonlinear system are valued in the range of $0.53 < R < 3.81$, $0 < \beta < 0.022$, $0 < \alpha < 0.015$, where X_0, Y_0 and Z_0 are defined in [1, 16, 17, 26].

2.2. 3D chebyshev map

Chebyshev polynomials are utilised to generate the secret keys required to hide information. Chebyshev polynomials are characterised as Fn (x) of the first type which is a polynomial of x with degree n. They comprise the prototype of a chaotic map and are defined as Fn(x) = cosnθ, where x = cosθ. By letting n = 0, 1, 2, 3, 4, we can obtain cos0θ = 1, cos1θ = cosθ, cos2θ = 2cos2θ − 1, cos3θ = 4cos3θ − 3cosθ and cos4θ = 8cos4θ − 8cos2θ + 1. With cosθ = x, we obtain F0(x) = 1, F1(x) = x, F2(x) = 2x2 − 1, F3(x) = 4x3 −3x and F4(x) = 8x4 −8x2 + 1. The transformations are expressed as

$$F2(x) = 2x^2 - 1, \tag{5}$$

$$F3(y) = 4y^3 - 3y, \tag{6}$$

$$F4(z) = 8z^4 - 8z^2 + 1. \tag{7}$$

The Chebyshev polynomial map is Fp: [−1, 1]→[−1, 1] of degree p, when p> 1 [16, 17]. The (5) to (7) are used to generate secret keys which are then used as a secret map of image pixels in the hiding process.

3. PROPOSED ALGORITHM

This section is composed of two phases (embedding and extracting phases) that are explained in the following subsections.

3.1. Embedding phase

The embedding phase includes several steps, including the following:

1. Select the secret message and host image.
2. Set the length of the secret message in the first two pixels of the host image.
3. Convert the secret message to ASCII values and then to binary numbers. For example, $S = 83$, 01010 011.
4. Initialise the secret parameters of the 3D Chebyshev map to generate secret keys X, Y and Z.

$$X = (X^* \ 10^4 \ mod \text{ the length of the binary secret message}), \tag{8}$$

$$Y = floor(Y^* \ 10^4 mod \ 3), \tag{9}$$

$$Z = (Z^* \ 10^4 mod 3). \tag{10}$$

5. Permute the secret message on the basis of the secret keys generated from (8) before hiding it in the host image. For example, let the secret message be 01010011 with a length of 8. Suppose that the secret keys of X are expressed as 1, 5, 6, 4, 0, 2, 3, 7. Then, the secret message is labelled as 10100011.
6. Decompose the binary numbers into three separate groups as follows: 10, 100, 011 (0, 1, 2).
7. Select the group that will be hidden first on the basis of the secret keys generated from (9). For example, let the generated secret keys be {1, 2, 0}. In this case, select 100 first, followed by 011 and 10.
8. Break down the red (R), green (G) and blue (B) components of the image. Store the components in three N×M arrays, where N and M are the number of array rows and columns, respectively.
9. Label the components as follows:
   RGB
   0 12
10. Select which component (R, G or B) will be hidden first on the basis of the secret keys generated from (10). For example, let the generated secret keys be {2, 0, 1}. In this case, the secret message 100 is hidden in the B component, followed by 011 in the R component and 10 in the G component.
11. Decompose each component (array) into nonoverlapping blocks by dividing N and M by 8. The result represents the number of blocks in each component. For example, the result is 128 blocks of 4×4 when$N$ and M are 512.
12. Initialise the secret parameters of the 3D logistic map.
13. Generate the secret keys for each block into R, G and B components.
14. Convert the secret keys into decimal numbers by using the following equations:

$$X = floor(X^* 10^4 mod 16), \tag{11}$$

$$Y = floor(Y{*}10^4 mod 16), \quad (12)$$

$$Z = floor(Z{*}10^4 mod 16), \quad (13)$$

Where X, Y and Z represent the secret keys for blocks R, G and B, respectively.

15. Store these secret keys in an 8×8 array with a range of 0–63. The values in the array should satisfy the condition without repeating the values in the rows and columns.
16. Map the values of the blocks with the values in Step 14 and hide their information. Hence, the host image pixels are randomly selected on the basis of the generated secret keys in each block in Step 14. For simplification, we take the following:

Secret keys (X)

5	6	3
2	4	0
1	7	8

Secret keys (Y)

2	5	6
3	4	1
0	8	7

Secret keys (Z)

3	1	0
6	5	8
4	7	2

Host image (R)

0	1	2
3	4	5
6	7	8

Host image (G)

0	1	2
3	4	5
6	7	8

Host image (B)

0	1	2
3	4	5
6	7	8

We choose the sixth (5) pixel in the block of host image (R) and convert it into binary form to embed 011 into 3LSB. Then, we choose the third (2) pixel in the block of host image (G) and convert it into binary form to embed 10 into 2LSB. Subsequently, we choose the fourth (3) pixel in the block of host image (B) and convert it into binary form to embed 100 into 3LSB.

17. Convert the binary values to decimal values.
18. Repeat Steps 13 to 16 to embed all bytes of the secret message in all components of the host image.
19. Obtain the stego image. Figure 1 presents the diagram of the embedding phase.

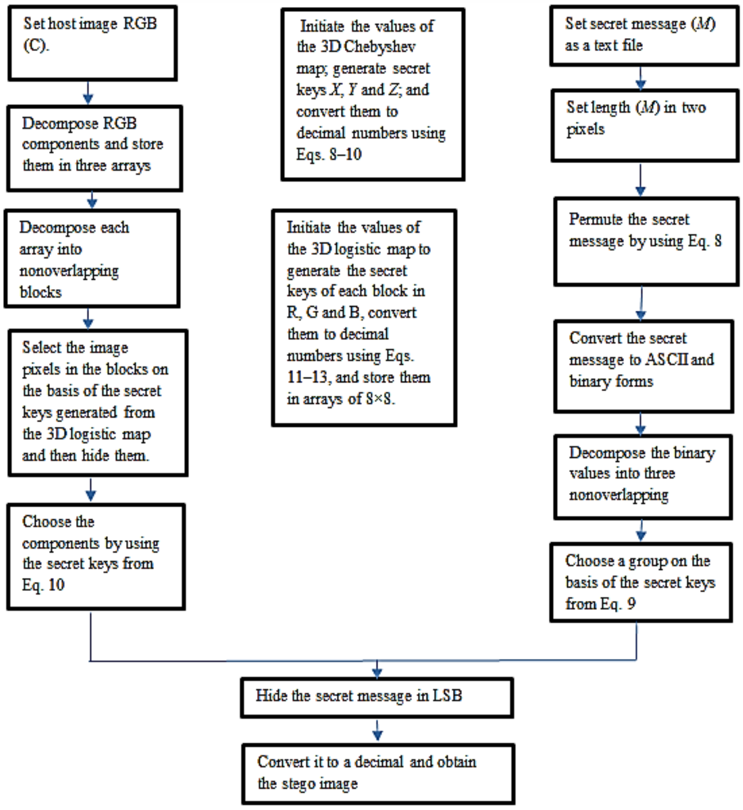

Set host image RGB (C).
Decompose RGB components and store them in three arrays
Decompose each array into nonoverlapping blocks
Select the image pixels in the blocks on the basis of the secret keys generated from the 3D logistic map and then hide them.
Choose the components by using the secret keys from Eq. 10
Initiate the values of the 3D Chebyshev map; generate secret keys X, Y and Z; and convert them to decimal numbers using Eqs. 8–10
Initiate the values of the 3D logistic map to generate the secret keys of each block in R, G and B, convert them to decimal numbers using Eqs. 11–13, and store them in arrays of 8×8.
Set secret message (M) as a text file
Set length (M) in two pixels
Permute the secret message by using Eq. 8
Convert the secret message to ASCII and binary forms
Decompose the binary values into three nonoverlapping
Choose a group on the basis of the secret keys from Eq. 9
Hide the secret message in LSB
Convert it to a decimal and obtain the stego image

Figure 1. Diagram of embedding phase

3.2. Extraction phase

In this phase, the secret message is retrieved from the stego image. This procedure is the opposite of the embedding process. In the extraction phase, the receiving party must be aware of the initial values of the 3D, Chebyshev and 3D logistic maps to produce secret keys *X*, *Y* and *Z*. The stego image is used as input in this phase. Subsequently, the stego image is blocked into nonoverlapping 4×4 blocks, and the image pixels are selected in the blocks on the basis of the secret keys for each block of the 3D logistic map, which are *X* for R, *Y* for G and *Z* for B. The procedure implemented in the embedding phase is then run. *Z* of the 3D Chebyshev map is obtained by using chaotic sequences. The order of the components is selected in the embedding process, whereas the chaotic sequences of *Y* determine the order of the groups of bits that are hidden. The original order of characters in the secret message is known through the *X* values.

4. EXPERIMENTAL RESULTS

The embedding and extraction phases of more than 30 images were run on MATLAB R2018a on a computer with Windows 10 64 bit, Intel Core i7-7500U processor, 8 GB CPU and 2400 MHz RAM. In this section, four standard well-known images, namely, Lena, Pepper, Baboon and Barbara, are presented. Figure 2(a–d) illustrates the host and stego images. As shown in the figure, the host and stego images do not present significant differences. Hence, the proposed algorithm can successfully hide secret messages in host images without any distortion. The correct secret messages can be easily and correctly extracted from stego images with valid stego keys when stego images are transmitted to authorised receivers, as explained in the next section. The following initial values were used in the 3D logistic and 3D Chebyshev maps in all experiments:

- For the 3D logistic map, $x_0 = 0.976$, $y_0 = 0.677$, $z_0 = 0.973$, $R = 3.79$, $\beta = 0.020$, $\alpha = 0.014$, where x denotes R, y denotes G and z denotes B.
- For the 3D Chebyshev map, $x_0 = 0.234$, $y_0 = -0.398$, $z_0 = -0.88$.

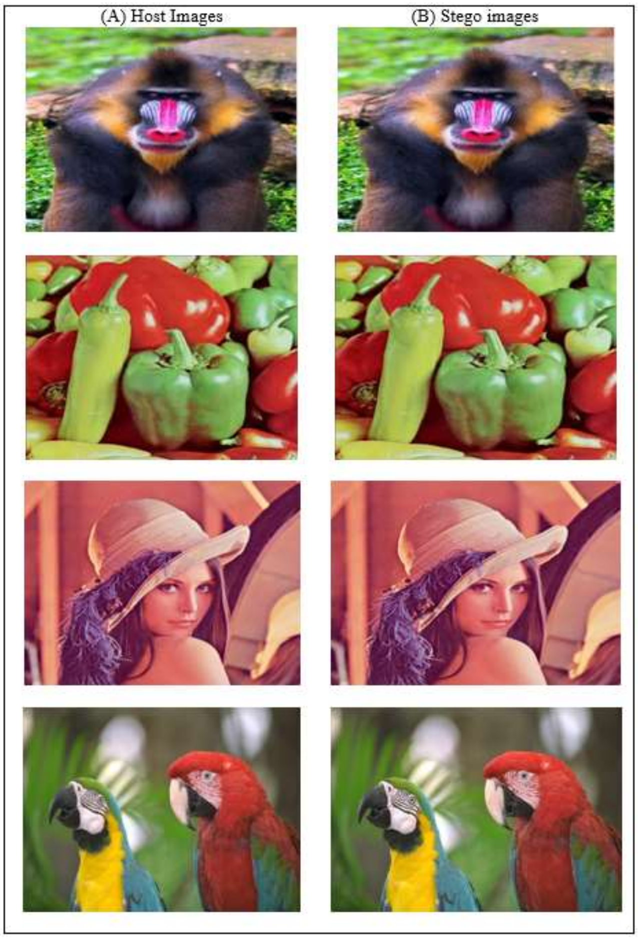


Figure 2. (A) host images, (B) stego images

5. SECURITY ANALYSIS

In this section, several statistical analyses are presented to verify the effectiveness and efficiency of the proposed algorithm against statistical attacks.

5.1. Correlation coefficient

Correlation coefficient r is used to measure the extent and direction of the linear correlation of two random variables. A correlation coefficient close to 1 indicates that two random variables are closely related; the opposite is true when the correlation coefficient is close to 0. Coefficient r can be calculated as follows [18]:

$$r=\frac{\sum_i(X_i-X_m)(Y_i-Y_m)}{\sum_i\sqrt{\sum_i(X_i-X_m)^2}\sqrt{\sum_i(Y_i-Y_m)^2}}, \quad (14)$$

Where X_i is the pixel intensity of the original image, X_m is the mean value of the original image intensity, Y_i is the pixel intensity of the stego image and Y_m is the mean value of the stego image intensity. The results of this test are shown in Table 1. All values in Table 1 are close to 1, indicating that the host and stego images are closely related.

Table 1 Correlation coefficient results

Image	Correlation coefficients		
	R	G	B
Baboon	0.9998	0.9998	0.9997
Lena	0.9983	0.9986	0.9965
Peppers	0.9995	0.9998	0.9994
Barbara	0.9998	0.9998	0.9998

5.2. Information entropy

The security of a steganographic system is measured in terms of entropy. Lete_1, e_2,..., e_m be m possible elements with probabilities P(e_1), P(e_2), ..., P(e_m). The entropy is given as

$$H(e) = -\sum_{i=0}^{m-1} P(e_i)log_2P(e_i). \quad (15)$$

This equation yields an estimate of the average minimum number of bits that is needed to encode a string of bits on the basis of the frequency of the symbol [27].

5.3. Homogeneity

The value returned in homogeneity analysis is used to determine how close the element distribution in the grey-level co-occurrence matrix(GLCM) is to the GLCM diagonal. Image homogeneity is calculated as

$$\text{Hom}=\sum_{i,j}\frac{p(i,j)}{1+|i-j|}, \quad (16)$$

where $p(i, j)$ denote the pixel values at the i^{th} row and j^{th} column and (i, j) represent the indices of row and column numbers, respectively [6].

5.4. Contrast

Contrast analysis produces a measure of the intensity contrast between a pixel and its neighbour in an entire image. For viewers, contrast analysis helps them recognise objects in the texture of an image. Contrast analysis is written as [6]

$$\text{C}=\sum_{i,j}|i-j|^2\, p(i,j). \quad (17)$$

Table 2 presents the results of the tests on the four standard images.

Table 2. Statistical analysis of four images

Statistical Analysis	Images	Host image			Stego image		
		R	G	B	R	G	B
Entropy	Baboon	7.8457	7.7842	7.5144	7.8457	7.7852	7.5146
	Lena	7.2477	7.5883	6.9232	7.2477	7.5884	6.9232
	Peppers	7.3857	7.6658	7.1687	7.3859	7.6658	7.1687
	Barbara	7.4892	7.4859	7.2022	7.4892	7.4859	7.2022
Homogeneity	Baboon	2.3501e+03	2.3054e+03	1.9043e+03	2.3491e+03	2.3031e+03	1.9025e+03
	Lena	1.4164e+03	752.1079	796.8155	1.4137e+03	749.4450	794.8281
	Peppers	2.6881e+03	2.2229e+03	1.2909e+03	2.6864e+03	2.2209e+03	1.2899e+03
	Barbara	2.7576e+03	3.0292e+03	1.9838e+03	2.7561e+03	3.0281e+03	1.9827e+03
Contrast	Baboon	2.7475e+09	2.5673e+09	2.1547e+09	2.7475e+09	2.5672e+09	2.1546e+09
	Lena	1.1391e+08	6.2484e+07	6.4222e+07	1.1367e+08	6.2307e+07	6.3999e+07
	Peppers	1.6035e+09	1.6052e+09	9.1440e+08	1.6034e+09	1.6052e+09	9.1437e+08
	Barbara	8.4412e+09	7.2760e+09	6.5267e+09	8.4399e+09	7.2773e+09	6.5267e+09

5.5. Image histogram

A histogram shows the exact occurrence of each pixel in the image. The high similarity between the host and stego image histograms indicates the occurrence of minimal distortion after embedding the secret image into the host image [5,10]. This test is performed on many images. The histogram of the Lena image is presented. Figure 3 shows the histogram of the host and stego images of three components. From Figure 3 can be shown that the histogram of the proposed algorithm highlights slight changes between the host and stego images.

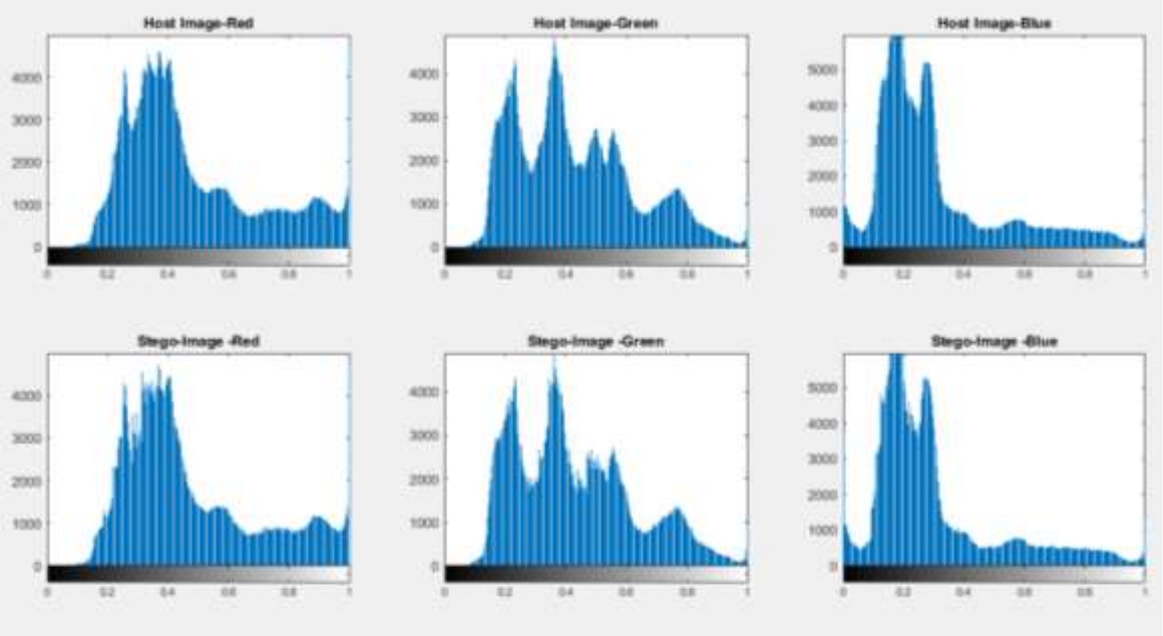
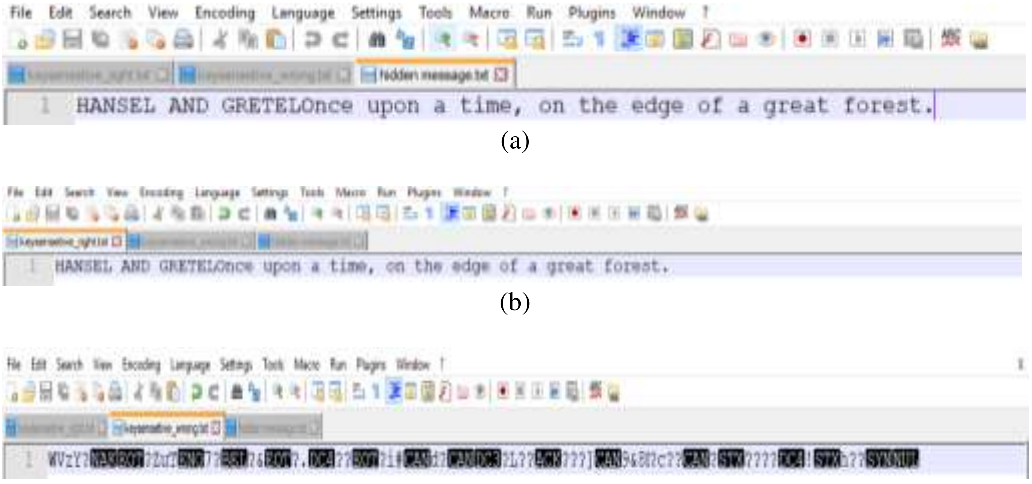


Figure 3. Histogram of host and stego images of three components

5.6. Key sensitivity

Chaotic maps are extremely sensitive to initial conditions and system control parameters. The slightest change can cause difficulties in the extraction of hidden messages from stego images [6, 18]. The key sensitivity test conducted in this work is aimed at establishing the sensitivity of the proposed algorithm to slight modifications in secret keys. 3D logistic and Chebyshev maps are used in the proposed algorithm and are rigorously evaluated. The sensitivity of the proposed algorithm towards initial state conditions is shown accordingly. The Pepper image is used as the host image in this test. The first change is applied to the initial values of the 3D logistic map. The subsequent change is applied to the initial values of the 3D Chebyshev map. Suppose that the selected keys for the 3D logistic map are $\alpha = 0.014$, $\beta = 0.020$ and $R = 3.79$ while the slightly different keys are $\alpha = 0.01400001$, $\beta = 0.020$ and $R = 3.79$; $\alpha = 0.014$, $\beta = 0.02000001$ and $R = 3.79$; and $\alpha = 0.014$, $\beta = 0.020$ and $R = 3.7900001$. Figure 4 shows that the hidden message cannot be extracted from the stego image. Suppose that the selected keys for the 3D Chebyshev map are x0 = 0.234, y0 = −0.398 and z0 = −0.88 while the slightly different keys are x0 = 0.23400001, y0 = −0.398 and z0 = −0.88; x0 = 0.234, y0 = −0.39800001 and z0 = −0.88; and x0 = 0.234, y0 = −0.398 and z0 = −0.8800001. Briefly, we present only the case of α = 0.014, β = 0.020 and R = 3.79 and the slight changes of α = 0.01400001, β = 0.020 and R = 3.79.


File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
keygenerative_right.txt
keygenerative_wrong.txt
hidden message.txt

```
HANSEL AND GRETELOnce upon a time, on the edge of a great forest.
```

(a)

```
HANSEL AND GRETELOnce upon a time, on the edge of a great forest.
```

(b)

```
WVzY?NAKEOT?ZuTENQ??BEL?&EOT?.DC4??EOT?i#CANd?CANDC3?L??ACK???]CAN9&8?c??CAN?STX????DC4!STXh??SYNNUL
```

(c)

Figure 4. (a) Hidden message, (b) text extraction using the right key, (c) text extraction using the wrong key

5.7. Hiding capacity

Hiding capacity refers to the maximum number of bits that can be hidden in a host image while ensuring the acceptable quality of the resultant stego image. A large hiding capacity boosts the performance of steganographic schemes [7]. In the proposed algorithm, one byte is embedded in each pixel of the true image. Each pixel contains three components, namely, R, G and B. Each component contains one byte. Therefore, the capacity of the proposed algorithm is equal to 8/24.

5.8. Quality index

We measure the quality of the stego image by using a quality index as shown in Table 3 [20] calculated as

$$Q = \frac{4\,\sigma_{HT} H' T'}{(\sigma_{H+}^2 \sigma_T^2)[H'^2 + T'^2]}, \tag{18}$$

$$\sigma^2_{H=\frac{1}{N-1}} \sum_{i=1}^{N} (H_i - H')2,$$

$$H' = \frac{1}{N} \sum_{i=1}^{N} H_{i,=}\ T' = \frac{1}{N} \sum_{i=1}^{N} T_i,$$

$$\sigma_H^2 = \frac{1}{N-1} \sum_{i=1}^{N} (T_i - T')^2\ ,$$

where n is the number of pixels in the image, H is the host image and T is the stego image. Q falls in the range of 1 and −1. The host and stego images are dissimilar when the calculated value is −1, whereas the two images are identical when the calculated value is 1 [6]. This test is applied to the Barbara image for the three components R, G and B after hiding the secret message with 50,000 letters. The results are presented in Table 3. Some of the results are 1, and others are close to 1. Thus, the proposed algorithm has good image quality, and the stego image has high similarity with the host image. Therefore, statistical analysis cannot be used to extract secret messages and overcome the steganography algorithm.

Table 3. Results of quality index

Images	Quality Index		
	R	G	B
Baboon	0.9992	1.0000	1.0000
Lena	1.0000	1.0000	1.0000
Peppers	1.0000	0.9960	0.9910
Barbara	1.0000	1.0000	0.9997

5.9. Mean square error

The mean square error (MSE) is calculated by comparing the bytes of two images. A pixel comprises 8 bits, and thus, 256 levels are available to represent various grey levels. MSEs are valuable when the bytes of an image are compared with the corresponding bytes of another image. Let h and s be the host and stego images, respectively. MSE can be computed as

$$\text{MSE}=\frac{\mathbf{1}}{\boldsymbol{HW}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{H}}\sum_{\boldsymbol{j=1}}^{\boldsymbol{W}}(\boldsymbol{P}(\boldsymbol{i},\boldsymbol{j})-\boldsymbol{S}(\boldsymbol{i},\boldsymbol{j}))^{\mathbf{2}}, \tag{19}$$

where H and W respectively denote the numbers of rows and columns of the host image, P (i, j) represents the pixel of the host image at the (i, j) position and S(i, j) represents the pixel of the stego image at the (i, j) position. The best value of MSE is the value that minimises it [7, 11]. This test is applied to four images for three components R, G and B after hiding the secret message with 50,000 letters. The results are shown in Table 4. The small values in Table 4 indicate that the proposed algorithm passes the test.

Table 4. Results of MSE

Images	MSE		
	R	*G*	*B*
Baboon	1.5877	1.8215	1.6841
Lena	2.9967	2.0188	1.8811
Peppers	2.0621	2.2507	2.0776
Barbara	1.1518	1.0852	1.1426

5.10. Peaksignal-to-noise ratio

The peak signal-to-noise ratio (PSNR) is a parameter used to measure the amount of imperceptibility in decibels. It measures the quality between the host and stego images. A large PSNR value indicates that a small difference exists between the host and stego images. By contrast, a small PSNR value indicates a huge distortion between the host and stego images. The steganographic algorithm aims to provide a large PSNR value. PSNR is defined on the basis of the MSE as follows:

$$\text{PSNR}=0.\,log_{10}\left(\frac{L^2}{MSE}\right)=20.log_{10}\left(\frac{L^2}{\sqrt{MSE}}\right), \tag{20}$$

where *L* denotes a greyscale image's peak signal level and is equal to 255 [5, 7, 28]. The PSNR test is applied to four images for three components R, G and B after hiding the secret message with 50,000 letters. The results are shown in Table 5. The proposed algorithm obviously generates large PSNR values, which indicate strong resistance against statistical attacks.

Table 5. Results of the peak signal-to-noise ratio (PSNR)

Images	PSNR		
	R	*G*	*B*
Baboon	45.9012	45.5605	46.1571
Lena	39.1357	39.1237	39.1989
Peppers	45.0216	44.6416	44.9892
Barbara	47.5509	47.8098	47.5859

5.11. Image fidelity

Image fidelity is another metric used to show the robustness of the proposed scheme. Image fidelity is calculated as follows [6]:

$$IF = 1\text{-}\frac{\sum_{I,J}(P(i,j)-S(i,j))^2}{\sum_{I,J}P(i,j)\times S(i,j)}. \tag{21}$$

This test is applied to four images for three components R, G and B after hiding the secret message with 50,000 letters. The results are shown in Table 6.

Table 6. Results of image fidelity

Image	Image Fidelity		
	R	*G*	*B*
Baboon	0.9999	0.9999	0.9998
Lena	0.9994	0.9994	0.9994
Peppers	0.9999	0.9999	0.9997
Barbara	0.9999	0.9999	0.9999

5.12. Comparison of the proposed scheme with other methods

The results of the proposed algorithm for the Baboon image with a size of 256×256 in three tests are compared with those of other methods as shown in Table 7. The proposed algorithm is clearly superior to all methods in terms of PSNR, quality index and image fidelity

Table 7. Results of comparison between the proposed scheme and other methods

Methods	PSNR	Quality Index	Image Fidelity
Proposed method	45.8661	0.9998	0.9999
Sharif, Mollaeefar and Nazari [6]	38.7540	0.99967	0.99940
Ghebleh and Kanso [14]	36.5437	0.99905	0.99900
Bandyopadhyay, Dasgupta, Mandal and Dutta [2]	33.5467	0.99865	0.99100

5.13. Knownhost attack

The adversary in a known host attack holds information about the host image. This adversary then compares the host image with the stego image through statistical analysis to identify any pattern differences. This type of attack can be avoided by determining host pixel positions using high-level chaotic maps. Such process depends greatly on (11-13), the results of which prevent the adversary holding the host and stego images (without the secret message) from determining critical information through statistical analysis. As indicated by these facts and the experimental analysis (Sections 5.1 to 5.5), the proposed algorithm hides secret data such that only the most crucial change between the host and the stego image is exist. Therefore, the adversary cannot gain anything except similar patterns. The proposed algorithm can has good statistical analysis, which in turn enhances its robustness against known host attacks.

5.14. Known message attack

The adversary in a known message attack is aware of the original message. Any adversary holding the original message and host image cannot obtain hidden information because the proposed algorithm depends heavily on secret keys and host images. Even the slightest change in secret keys or host images can change the position of the embedded secret data. Thus, the proposed algorithm is robust against known message attacks.

6. CONCLUSION AND FUTURE WORK

Image steganography using LSB and secret map techniques is proposed in this study. The secret maps are primarily based on 3D chaotic maps, which are 3D Chebyshev and 3D logistic maps. The random concept focuses on the insertion and selection of host image pixels. This process provides high level of security and resistance to different types of attacks. Empirical results, such as correlation coefficient, information entropy, homogeneity, contrast, image histogram, hiding capacity, quality index, MSE, PSNR and image fidelity, prove the satisfactory performance of the proposed algorithm and its superiority to other algorithms. The proposed algorithm also has high sensitivity to its secret keys. Future work should investigate image steganography with secret map bioinformatics, such as immune system, swarm and ant colony algorithms.

REFERENCES

[1] G. Swain and S. K. Lenka, “A novel steganography technique by mapping words with LSB array,” *International Journal of Signal and Imaging Systems Engineering,* vol. 8, pp. 115-122, 2015.

[2] D. Bandyopadhyay, K. Dasgupta, J. Mandal, and P. Dutta, “A novel secure image steganography method based on chaos theory in spatial domain,” *International Journal of Security, Privacy and Trust Management* (IJSPTM), vol. 3, pp. 11-22, 2014.

[3] M. A. Al-Husainy, “Image steganography by mapping pixels to letters,” *Journal of Computer Science,* vol. 5, pp. 33, 2009.

[4] H.-C. Wu*, et al.*, “Image steganographic scheme based on pixel-value differencing and LSB replacement methods,” *IEEE Proceedings Vision, Image and Signal Processing,* vol. 152, pp. 611-615, 2005.

[5] S. Rajendran and M. Doraipandian, “Chaotic map based random image steganography using LSB technique,” *IJ Network Security,* vol. 19, pp. 593-598, 2017.

[6] A. Sharif, M. Mollaeefar, and M. Nazari, “A novel method for digital image steganography based on a new three-dimensional chaotic map,” *Multimedia Tools and Applications*, vol. 76, pp. 7849-7867, 2017.

[7] T. Bedwal and M. Kumar, “An enhanced and secure image steganographic technique using RGB-box mapping,” *IET Conference Publications*, 2013.

[8] A. T. Al-Taani and A. M. Al-Issa, “A novel steganographic method for gray-level images,” *International Journal of Computer, Information, and Systems Science, and Engineering,* vol. 3, 2009.

[9] M. Mishra, A. R. Routray, and S. Kumar, “High security image steganography with modified Arnold cat map,” *arXiv preprint arXiv: 1408.3838*, 2014.

[10] M. R. Islam, A. Siddiqa, M. P. Uddin, A. K. Mandal, and M. D. Hossain, “An efficient filtering based approach improving LSB image steganography using status bit along with AES cryptography,” in *2014 International Conference on Informatics, Electronics & Vision (ICIEV)*, pp. 1-6, 2014.

[11] A. Singh and H. Singh, “An improved LSB based image steganography technique for RGB images,” in *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1-4. 2015.

[12] S. D. Ahmadi and H. Sajedi, “Image steganography with artificial immune system,” in *2017 Artificial Intelligence and Robotics (IRANOPEN)*, pp. 45-50, 2017.

[13] S. Thenmozhi and M. Chandrasekaran, “A novel technique for image steganography using nonlinear chaotic map,” in *2013 7th International Conference on Intelligent Systems and Control (ISCO)*, pp. 307-311, 2013.

[14] I. Bremnavas, B. Poorna, and I. R. Mohamed, “Secured medical image transmission using chaotic map,” *Elixir Comp. Sci. Eng*, vol. 54, pp. 2472-2478, 2013.

[15] C. Thampi and D. Jose, “More secure color image encryption scheme based on 3D chaotic maps,” *International Journal for Advance Research in Engineering and Technology,* vol. 1, 2015.

[16] A. M. Alabaichi, “Color image encryption using 3D chaotic map with AES key dependent S-box,” *International Journal of Computer Science and Network Security (IJCSNS),* vol. 16, pp. 105-115, 2016.

[17] A. Alabaichi, “True color image encryption based on DNA sequence, 3D chaotic map, and key-dependent DNA S-box of AES,” *Journal of Theoretical & Applied Information Technology,* vol. 96, 2018.

[18] M. Ghebleh and A. Kanso, “A robust chaotic algorithm for digital image steganography,” *Communications in Nonlinear Science and Numerical Simulation,* vol. 19, pp. 1898-1907, 2014.

[19] M. Bilal, S. Imtiaz, W. Abdul, S. Ghouzali, and S. Asif, “Chaos based zero-steganography algorithm,” *Multimedia Tools and Applications,* vol. 72, pp. 1073-1092, 2014.

[20] S. Alam, V. Kumar, W. A. Siddiqui, and M. Ahmad, “Key dependent image steganography using edge detection,” in *2014 Fourth International Conference on Advanced Computing & Communication Technologies*, pp. 85-88, 2014.

[21] M. Sabery K. and M. Yaghoobi, “A simple and robust approach for image hiding using chaotic logistic map,” in *2008 International Conference on Advanced Computer Theory and Engineering*, pp. 623-627, 2008.

[22] R. Roy, A. Sarkar, and S. Changder, “Chaos based edge adaptive image steganography,” *Procedia Technology,* vol. 10, pp. 138-146, 2013.

[23] A. Kanso and H. S. Own, “Steganographic algorithm based on a chaotic map,”*Communications in Nonlinear Science and Numerical Simulation*, vol. 17, pp. 3287-3302, 2012.

[24] A. Anees, A. M. Siddiqui, J. Ahmed, and I. Hussain, “A technique for digital steganography using chaotic maps,” *Nonlinear Dynamics,* vol. 75, pp. 807-816, 2014.

[25] N. S. Raghava, A. Kumar, A. Deep and A.Chahal, “Improved LSB method for Image Steganography using H´enon Chaotic Map,” *open journal of information security and applications*, vol. 1, no. 1, pp. 34-42, jun 2014.

[26] A. I. Salih, A. Alabaichi, and A. S. Abbas, “A novel approach for enhancing security of advance encryption standard using private XOR table and 3D chaotic regarding to software quality factor,” *ICIC Express Letters Part B: Applications, An International Journal of Research and Surveys*, vol. 10, no. 9, pp. 823–832, 2019.

[27] K. Raja, C. Chowdary, K. Venugopal, and L. Patnaik, “A secure image steganography using LSB, DCT and compression techniques on raw images,” in *2005 Third International Conference on Intelligent Sensing and Information Processing*, pp. 170-176, 2005.

[28] S. M. Karim, M. S. Rahman, and M. I. Hossain, “A new approach for LSB based image steganography using secret key,” in *14th International Conference on Computer and Information Technology* (*ICCIT*), pp. 286-291, 2011.