Adaptive LSB Steganography Based on Chaos Theory and Random Distortion

Kemal TUTUNCU[1], Baris DEMIRCI[2]
[1]*Department of Electrical & Electronics Engineering, Faculty of Technology, Selcuk University, 42075, Selcuklu-Konya, Turkey*
[2]*Bahçelievler Türk Telekom Mesleki ve Teknik Anadolu Lisesi, 34197, Bahcelievler-Istanbul, Turkey*
ktutuncu@selcuk.edu.tr

***Abstract*—Image steganography is a technique to hide secret information in an image without leaving any apparent evidence of image alteration. Hiding capacity, perceptual transparency, robustness, and resistance against attack must be considered as characteristics of the image steganography algorithms. In this study, Improved Chaos Based Bit Embedding has been proposed as a new steganography algorithm. It is based on two basic principles. One of them is determining the bits in which the secret data will be embedded by logistic map and the other one is embedding the secret data into only one of the three color channels that is chosen randomly. It distorts the other remaining channels so that it is harder to obtain the text within the image by an unwanted person. The proposed algorithm has been tested on 10 sample images along with the four basic steganography algorithms: Least Significant Bit Embedding, Pseudo Random Least Significant Bit Embedding, EzStego, and F5. It has been seen that generating unpredictable indexes by the chaotic random number generators, and embedding the secret data into only one of the three channels (distorting remaining channels) increased resistance against attacks. Perceptual transparencies and capacity ratio of the proposed algorithm are compatible with the other four algorithms.**

***Index Terms*—ciphers, chaos, data encapsulation, data security, digital images.**

I. INTRODUCTION

The growing computer and network technologies have substantially facilitated dataflow and exchange in communication. This facilitating was accompanied by security issues involving the processing and protection of large quantities of data. The internet being the source of communication for everyone from the government's secret services to the fan clubs of famous people increased the need for security. In addition, the usage of devices such as mobile phones and tablets has become more popular among people of all views, causing serious debates regarding data security. Cyber-attacks on fields that require privacy from banking and national security to medicine and private life have exacerbated these debates and have increased the number of studies in the discipline of data security. Thus, ensuring privacy or security is one of the key issues in today's communication systems. In applications where the importance of privacy is essential the primary goal is to send the information that is requested to be private to the intended recipient in such a way as to ensure that it does not fall into the hands of an unrelated third party and that said party cannot understand it. However, due to whatever reason, hidden information becoming known by falling into the hands of a third party is an undesirable situation.

All of these reasons led to a number of longstanding security applications being transferred to the computer sciences. Data security has been divided into many different areas and disciplines such as cryptography, secret sharing, watermarking, and steganography.

While cryptography and secret sharing extend far back to ancient history, watermarking and steganography are relatively new fields that have only been able to gain momentum with the expansion and widespread usage of digital computers. The common characteristics of these algorithms are concerned with data security, in other words, their aim is to protect data being accessed by third party. Cryptography is based on the modification of the requested data that can only be regenerated by the receiver [1]. In this sense, secret sharing is close to cryptography. According to this discipline, the data is divided between a certain number of people, and only when a determined amount of shares come together make the actual data be obtained [2].

In this regard, steganography is navigating on a whole different course compared to the other algorithms. The main goal of steganography is not to modify secret data but to hide it within a different unrelated data. This unrelated type of data should not attract the attention of unwanted persons in case of having fallen into their hands. These data can sometimes be a video file, an image, an html file, and sometimes even a musician's number one hit. It comes under the assumption that if the feature is visible, the point of attack is evident, thus the goal here is always to conceal the very existence of the embedded data [3].

Image steganography is the driving force among the steganography algorithms that make use of different multimedia carriers since the image is the most common form that is shared and transferred in both social media and e-mail environments. Video and sound files' sharing are limited or prohibited in most of the institutions. In addition, these files get the attention of third parties. Moreover, some other software or coder/decoder may be required to open these files, whereas an image file can be stored and opened easily without any additional software.

Image steganography algorithms are divided into two categories: spatial domain and transform domain based algorithms. Spatial domain based algorithms are the most important and widely used ones out of these two categories. The subtitles under this category are: least significant bit, pixel value difference, edge-based data embedding, random pixel embedding, addressing pixel to a hidden data, labeling, pixel intensity-based, tissue-based and histogram shifting algorithms. Spatial domain based algorithms generally embed the sensitive information in the Least Significant Bits

Digital Object Identifier 10.4316/AECE.2018.03003

(LSBs) of the cover image pixels [4–13]. The original image suffers less deterioration and more extensive data can be hidden within an image with these algorithms. These algorithms are endurable since a small distortion on the image the original hidden data will be lost, and also many steganalysis algorithms can solve these data. On the other hand, transform domain based algorithms embed sensitive information in the cover image by modulating coefficients in a transform domain, such as the Discrete Fourier Transform, Discrete Cosine Transform, or Wavelet Transform [14–16]. As for conversion field algorithms, they are more difficult and a more complex algorithm structure is concerned. Different conversion algorithms can be applied to embed the data. Discrete Fourier Conversion, Discrete Cosine Conversion, Discrete Wavelet Conversion are examples of conversions that can be used for this purpose. These algorithms are more robust than spatial domain algorithms with respect to cropping, compression, and image processing. In addition, they provide better resistance against statistical attacks than spatial domain algorithms. Despite the fact that the transform domain based algorithms are more resistant and robust, the spatial domain based algorithms have somewhat wider usage due to being simpler and faster.

Before going into the literature review regarding steganography on images, it will be appropriate to indicate the objectives of a steganography algorithm. Expectations regarding steganography on images have been determined as in Table I by Hussain and Hussain [17].

TABLE I. THE CRITERIA DETERMINING THE QUALITY OF THE STEGANOGRAPHY ALGORITHM AND THE ADVANTAGES AND DISADVANTAGES OF THESE CRITERIA

Criterion	Description	Wanted	Unwanted
High Capacity	The maximum amount of data that can be embedded into a cover image	High	Low
Perceptual Transparency	The difference being detected by a person and then being suspicious of data retention after the embedding process	High	Low
Substantiality	The text preservation or non-preservation (in other words its original condition) without loss after processes such as enlargement and rotation occur after embedding.	High	Low
Annealing Resistance	The possibility of changing/updating the text after embedding it into a cover image	High	Low
Computational Complexity	The calculation cost of restoring embedded data and embedding it into a cover image, in other words difficulty	Low	High

II. RELATED WORKS

Some of the most commonly used image steganography algorithms are LSB embedding, Pseudo Random Least Significant Bit (PRLSB) embedding, pallet based embedding and quantification based embedding [18-21]. Going through the available literature on these and other algorithms, it has been seen that most of the algorithms are formed on the basis of the LSB embedding algorithm, and any improvement on LSB embedding algorithm can cause significant results. The weakest chain of the LSB embedding algorithm is embedding the data into an LSB of pixel of cover image in an order. This can cause detection, as well as extraction of the secret message easily. PRLSB embedding algorithm is just a step ahead of the improvement of LSB embedding algorithm. However, random number generators of PRLSB produce indexes that can be easily predicted. Thus, more powerful number generator must be produced in order to overcome weak resistance against any attack regarding to LSB based algorithms. At this point, the chaos theory plays an important role due to being unpredictable [22].

In the literature, the chaos theory is used by pioneers [23-26] in the field of image cryptography and watermarking [27-29]. It has also been studied merely in the field of image steganography [30-35]. This has been a factor that has clinched the ideas regarding using chaos based random number generators in the course of this study. In the study [30], the authors proposed steganography algorithm that only works for JPEG images. Their algorithm first finds the best parameters for the Logistic Map by using GA, and then rearranges the secret message and embeds it into the cover image using LSB. They used 4 cover images for test regarding to Peak Signal-to-Noise Ratio (PSNR) and embedding capacity values. They claimed that their algorithm achieved a high embedding capacity, at the same time preserving good image quality and high security. In the study [31] the authors proposed a variable-sized steganographic algorithm based on a single 2-D chaotic map. They applied their algorithm on 4 cover images and claimed that resistance of their algorithm against some existing steganalytic attacks is high. In order to prove by its usage, they haven't made any comparisons of their algorithms with previous well known algorithms regarding capacity, perceptibility, and security. In the study [32], they combined matrix encoding and LSB for embedding in the edge regions of a cover image. Their algorithm makes use of cat chaotic mapping before embedding the text into cover image. They used 4 cover images to compare their algorithm with an LSB embedding algorithm and Pixel Value Differencing (PVD) algorithms and claimed that their algorithm performs better in terms of stego image fidelity. In the study [33], the authors used discrete cross-coupled chaotic maps for specifying the location of the different parts of the secret data in the image. They claimed that their algorithm had high robustness and resistance against hackers. In the studies of [34-35], the authors used a chaos theory for transform domain based image steganography that is completely different from the spatial domain steganography. As mentioned previously, this study makes use of an LSB based embedding algorithm of spatial domain steganography. Thus the algorithms followed in [34-35] are completely different from those applied in this study. In the study [34], a digital image steganographic technique, based on a 3-dimensional chaotic cat map and lifted discrete wavelet transforms, has been proposed. The authors claimed that the algorithm proposed is shown to have superior

performance regarding to fidelity by using one cover image. In the study [35], the authors proposed a transform domain based steganography that is based on a combination of integer wavelet transform (IWT) and a chaotic map for digital images. According to the authors of this paper, their algorithm has good imperceptibility and a higher PSNR value when it was compared to four other transform domain based algorithms. At this point, it should be noted that the previous studies [30-35] used different images and test algorithms for the sake of comparison. Thus there is no way of comparing these studies to each other or to the proposed algorithms since different cover images, text messages and test algorithms were used.

In this regard, a new steganography algorithm that uses chaotic algorithm by a logistic map to hide secret data in cover images is proposed. The proposed algorithm also includes embedding the secret data into only one of the three color channels that has been chosen randomly, and distorts the other remaining channels. The proposed algorithm works on any colored cover images, and has higher safety or resistance against attacks. The proposed algorithm can be regarded as a fixed-size embedding algorithm. The proposed algorithm is compared to four other well-known image steganography algorithms in order to establish the advantages of the proposed one.

III. Other Algorithms Compared In This Study

In this article, in order to test the performance of the proposed algorithm, a comparison was made with four fundamental steganography algorithms from various fields. These are; LSB embedding, PRLSB embedding, EzStego, and F5 algorithms.

Almost all of these algorithms are based on LSB embedding algorithm. In LSB algorithm, the data that will be embedded in the cover image is expressed in a binary form. Each bit of the data is embedded into the last pixel of the cover image respectively. Thus, the change in the last bit of a pixel having a value range of 0-255 leads to a maximum difference in the rate of 1/255. This amount means a very small color change that cannot be detected by the human eye. Therefore, when the stego image with embedded data reaches the desired person, the data can be obtained by that person quite easily.

PRLSB is a different algorithm that is based on the same principles as LSB algorithm. The starting point of this algorithm is to place the text bits randomly rather than in a certain order. Thus, the aim is that even if the cover image and the stego image fall into unwanted hands the difference cannot be told. However, it is known that pseudo random number generators are more susceptible to statistical tests.

Ezstego algorithm is an algorithm that can embed data into palette-based images. The principle of this algorithm is embedding the data into the palette index rather than the last bit. The images on the palette are sorted according to proximity. The Euclidean Distance is often used as a proximity measure. After that, the pixels of the image are scanned and a matrix that shows where each pixel will be located in the new palette is obtained. The data embedding process is performed on this matrix by the form of LSB embedding. The extraction of data can easily be performed.

As for F5 algorithm, it is a algorithm that can be applied to images with the JPEG standard. It is based on discrete cosine transform and quantization.

IV. Proposed Algorithm

It is useful to talk about the chaos concept before explaining the proposed algorithm.

A. The Chaos Concept and the Logistic Map

The Chaos Theory is a field that studies the behavior of dynamic systems. It is greatly sensitive to initial conditions. The changes depend on the time of chaotic systems cannot be predicted in the long run. However, this change is deterministic, and it can be determined by the initial conditions. This determinable and natural structure of the chaotic systems makes these systems unpredictable. A discrete chaotic system is defined by an iterative "f" function (chaotic map) as can be seen in Equation 1 and an "I" state space [36].

$$x_{n+1}=f(x_n)x_{n+1}=f(x_n) \quad (1)$$

In this study, one-dimensional chaotic map difference equation which is expressed by the Equation 2 was used in order to generate a random number.

$$x_{n+1}=f(x_n,r),n=0,1,2,3... \quad (2)$$

In this formula, r is the system control parameter while x represents the state variable. The value x_{n+1} of the system depends on the value of x_n and r. The logistic map as expressed in Equation 3 is an one-dimensional polynomial map which varies in certain circumstances.

$$x_{n+1}=rx_n(1-x_n) \quad (3)$$

There are disadvantages alongside the advantages of a chaotic number generation. The disadvantage is related with testing chaos based image steganography since mostly developed testing tools are based on statistical generators. There isn't a standard algorithm that can test chaotic generators entirely. The advantage is the nature of the chaotic number generation where produced number are unpredictable.

B. Proposed Algorithm

Although the Classic-LSB embedding algorithm results in a difference at undetectable level by the human eye degree, it is considered insecure since it places the text bits sequentially. In spite of PRLSB data embedding algorithm is more reliable in this respect, these algorithms are weak against brute-force algorithms and has easily soluble nature because these generators are known to generate pseudo random number in fact. These numbers are predictable since the algorithms used in the generators are known.

For the reasons listed, in this study, a new algorithm based on the LBS embedding approach whose indices are determined by the chaotic random number generators is proposed. Chaotic random number generators produce random and unpredictable numbers.

The developed algorithm is based on two basic principles. One of them is determining the bits that will be embedded in data as chaotic, and the other is inability to determine that data is embedded in which color channel by distortion of other remaining channels.

The proposed algorithm performs following steps to embed secret data (text) in a colorful cover image:

Step 1: Receive the text will be embedded and cover image source. Ask the user to enter the parameters r and x.

Step 2: Allocate the text into bits (translate into binary data).

Step 3: Produce chaotic random numbers as same amount as the number of bits in the text. Import these numbers into array (array).

Step 4: Transfer the sequential numbers from one up to the number of pixels in cover image into the dynamic arrays (sequence).

Step 5: Apply the steps 6-8 until all bits are embedded.

Step 6: Receive the next element of the chaotic array (k). Admit this element of the indices in sequence array as indices (indices = sequence (k)), and remove it from the sequence array. Define the data to the indices.

Step 7: If the indices are divided by 3, embed the data into the blue channel. If it is divided by 2, embed the data into the green channel. Otherwise, embed the data into the red channel.

Step 8: Disrupt the last pixels of the channels except the data embedded channel (do 1 if it is 0 or vice versa).

Step 9: End the process.

Figure 1. (a) Cover image (b) The stego image (c) The different regions between the two images.

In Figure 1, the difference resulting data embedding on a sample cover image is shown. The histograms of these images and the difference between histogram values can be seen in Figure 2. It should be noted here that both cover and stego images were converted to gray scale images to compare the histogram values. As can be seen from Figure 2, some colors in both gray scale images haven't been changed. Thus, the difference between the histogram of cover image and stego image can be "0" for related colors.

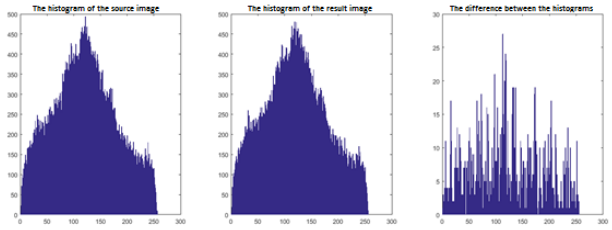

Figure 2. The histograms of the cover image and stego image, and the difference between these histograms

The required algorithm for uncovering the embedded data (text) is listed below.

Step 1: Receive the stego image. Ask the user to enter the parameters r and x. Receive the length of text from the user.

Step 2: Produce chaotic random numbers as the same amount as number of bits in the text. Import these numbers into array (array).

Step 3: Transfer the sequential numbers from one up to the number of pixels in the cover image into the dynamic arrays (sequence).

Step 4: Apply the steps 5-6 until all bits are obtained.

Step 5: Receive the next element of the chaotic array (k). Admit this element of the indexes in sequence array as indexes (indexes = sequence (k)), and remove it from the sequence array.

Step 6: If the indexes are divided by 3, receive the bit from the blue channel, if it is divided by 2, receive the bit from the green channel; otherwise, receive the bit from the red channel, and insert it into the bit array.

Step 7: Convert the bit array into text with respect to the selected coding algorithm.

Step 8: End the process.

V. Testing Algorithms

In this study, PSNR test, Chi-square test, Structural Similarity Measure (SSIM) test and Capacity test were applied as test algorithms. They are commonly referenced in the literature. Moreover; In case of falling both the cover image and the stego image into the hands of undesirable persons, the required algorithm complexity was noted in order to regain the text.

A. Chi-Square Test

Chi-square test is known as X^2 test in literature. [37,38]. It is one of the commonly used steganalysis algorithm which was proposed by Westfeld and Pfitzmann in 2000 [38]. It is based on a powerful statistical attack based on histogram analysis of Pairs of Values (PoVs) that are swapped during the message embedding process. Since swapping one value into another does not change the sum of occurrences of both values in each pair, the arithmetic mean of the two frequencies for each pair is the same in both the original and stego image. This fact allows us to obtain the expected frequency from the stego image. Chi-square test can be used to determine the degree of similarity between cover and stego images. The lower Chi-square test value the better similarity between images. The steps of Chi-square test are as follows:

Step 1: Suppose that there are k number of categories and that we have a random sample of observation. Each observation must fall within only one category. For example, for a palette image there are at most 256 colors in the palette, which means at most 128 PoVs and k = 128.

Step 2: The theoretically expected frequency in ith category, i = 1, 2, . . k after embedding an equally distributed message bits is defined as

$$n_i^* = \frac{\left|\left\{\text{colour}\,\middle|\,\text{sortedIndexOf(colour)} \in \{2i,2i+1\}\right\}\right|}{2} \quad (4)$$

Step 3: the actual frequency of occurrence in the sample is

$$n_i = \left|\left\{\text{colour}\,\middle|\,\text{sortedIndexOf(colour)}=2i\right\}\right| \quad (5)$$

Step 4: Chi-square statistic is calculated as

$$X^2 = \sum_{i=1}^{k} \frac{\left(n_i - n_i^*\right)^2}{n_i^*} \quad (6)$$

with k-1 degrees of freedom.

B. Capacity Test

Capacity test is a test for determining the maximum number of bits of data can be embedded into a cover image. It is not directly a reliability test. Most of the time it is used

for measuring efficiency. Capacity test supplies variability for each algorithm but it does not have a certain formula.

C. SSIM Test

SSIM test is one of the basic tests which measure the similarity between two images. This approach which is developed with different statistical algorithms is revealed by Wang and his colleagues in 2004 with the name Mean Structural Similarity Index: M–SSIM. A MSSIM value between the values [0…1] is produced in this approach [39].

SSIM is designed by modeling any image distortion as a combination of three factors that are loss of correlation, luminance distortion and contrast distortion.

Equations 7-10 show how to calculate SSIM given two images.

The first equation (7) is the luminance comparison function which measures the closeness of the two images mean luminance (μ_x and μ_y). This factor is maximal and equal to 1 only if $\mu_x=\mu_y$.

The second equation (8) is the contrast comparison function which measures the closeness of the contrast of the two images. Here the contrast is measured by the standard deviation σ_x and σ_y. This term is maximal and equal to 1 only if $\sigma_x=\sigma_y$.

The third equation (9) is the structure comparison function which measures the correlation coefficient between the two images x and y. Note that σ_{xy} is the covariance between x and y. The positive values of SSIM index are in [0,1]. A value of 0 means no correlation between images, and 1 means that x=y. The positive constants c1, c2 and c3 are used to avoid a null denominator. [40]

$$l(x,y)=\frac{2\mu_x\mu_y+c_1}{\mu_x^2+\mu_y^2+c_1} \tag{7}$$

$$c(x,y)=\frac{2\sigma_x\sigma_y+c_2}{\sigma_x^2+\sigma_y^2+c_2} \tag{8}$$

$$s(x,y)=\frac{\sigma_{xy}+c_3}{\sigma_x\sigma_y+c_3} \tag{9}$$

and SSIM is calculated with following formula:

$$SSIM(x,y)=\left[l(x,y)\right]^{\alpha}+\left[c(x,y)\right]^{\beta}+\left[s(x,y)\right]^{\gamma} \tag{10}$$

D. PSNR Test

This test is a test which is commonly used in order to measure the difference between two series of numbers, and it is based on the Mean Squared Error (MSE). MSE is computed by performing byte-by-byte comparisons of the cover and stego-image. The computation can be expressed as follows [41]:

$$MSE=\frac{1}{NxM}\sum_{i=1}^{M}\sum_{j=1}^{N}(X_{ij}-Y_{ij})^2 \tag{11}$$

Where M, N are the number of rows and columns in the cover image, X_{ij} is the pixel value from cover image, and Y_{ij} is the pixel value from the stego-image. The lower value of MSE indicates similarity between compared images.

According to Equation (11), the cover image and stego image are scanned pixel by pixel and the squares of the difference between the two pixels at the same location are summed. The reason for the summing of the squares is to prevent the positive and negative error summation from zero. The MSE represents the cumulative squared error between the cover and the stego image, whereas PSNR represents a measure of the peak error. PSNR measures in decibels the quality of the stego image compared with the cover image. A greater value of PSNR means better image quality. When there is uniformity of two images, PSNR has a value tending towards infinity. The main purpose of all steganography algorithms is to minimize the value of the MSE, and accordingly maximizing the value of PSNR. Thus PSNR is based on MSE and calculated as in the following equation [42]:

$$PSNR=10\log\left(\frac{R^2}{MSE}\right)(dB) \tag{12}$$

In Equation (12), R is the maximum fluctuation in the cover image data type. For example, if the cover image has a double-precision floating point data type, then R is 1. If it has an 8-bit unsigned integer data type, R is 255, etc.

VI. Test Results

As presented in the literature survey, not only did the previous studies compared their studies with the basic steganography algorithms but no common images were used as cover images as well [30-35]. Thus, there is no way of comparing this study with the results of their studies. Even they didn't make use of the same 5 tests (Chi-square, capacity, SSIM, PSNR, and computational complexity) as they were used in this study to measure the success of their studies. The final word is for the secret text message that will be embedded into the cover image. None of the authors of the previous studies [30-34] presented the secret text they used. These reasons let the authors of this paper to compare the proposed algorithm with 4 basic steganography algorithms by presenting the secret text, and using 4 different test algorithms.

10 cover images were used to measure the difference of the proposed algorithm against the other 4 algorithms. Each image was converted to a 256 color valued gif format for the palette-based test algorithm, and from gif to jpg format for F5 algorithm. As the other algorithms are able to operate on colored cover images, they were tested on the jpg formatted cover images.

A standard algorithm that can entirely test chaotic generators does not exist. It is known that the proposed algorithm is more suitable for testing on statistical generators. Nevertheless, the proposed algorithm will be tested by 5 testing algorithms mention in Section V along with the other algorithms. In figure 3, there are 10 cover images used for test purposes.

(a) (b) (c)

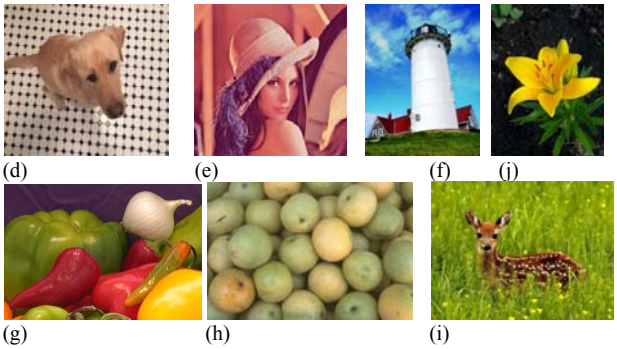

Figure 3. Cover images used for testing

The following poem belongs to Karacaoglan who is very famous Turkish poet was chosen as secret text to be embedded into the cover images. The text has 470 characters including space ones.

Ala gözlerini sevdiğim dilber Yurtlarınız çayır çimen pınar mı Mevlam güzelliği hep sana vermis Seni gören başkasını dener mi Sallanı sallanı gelmiş pınara Kadir Mevlam işimizi onara Gün doğmadan şavkın düşmüş Gün üstüne bir gün daha doğar Kırmızı gülden rengin almışsın Güzellikte kemalini bulmuşsun Sallanı sallanı suya gelmişsin Güzel senin ziyaretin pınar mı Karac'oğlan der ki ermediler mi Tomurcuk güllerin dermediler mi Seni sevdiğine vermediler mi Aşıkın ağlatan güzel güler mi

In the subsequent section of this chapter will show the results of the tests held by entering the parameters in Table II.

TABLE II. THE PARAMETERS OF THE ALGORITHMS USED

Algorithm Name	Parameters
LSB Embedding	None
PRLSB Embedding	Random Start : 3
EzStego	None
F5	Quality : 80
Proposed Algorithm	r: 3,573489 x : 0,970120

TABLE III. PSNR TEST RESULTS

Image No.	Image Size	LSB	PRLSB	EzStego	F5	Proposed Algorithm
a	518 x 391	74,52	72,36	63,96	36	68,97
b	640 x 370	75,25	73,81	61,68	31,77	69,88
c	650 x 644	77,78	76,33	59,04	37,28	72,33
d	1632 x 1224	84,15	82,7	71,74	38,29	78,83
e	512 x 512	75,8	74,02	56,59	36,44	70,14
f	480 x 640	76,17	74,66	64,13	34,99	70,92
g	198 x 135	65,77	64,43	61,89	33,33	60,15
h	732 x 486	76,97	75,38	59,13	36,44	71,34
i	600 x 450	75,75	74,63	67,48	32,01	70,21
j	500 x 667	76,38	75,57	65,24	36,17	70,97

As mentioned previously, PSNR is directly proportional to the sum of the difference of all the pixels of both cover and stego images. It produces high values for images that are close to each other and low values for images far from each other. From this point of view it can be claimed that algorithms that generate the highest value are more successful than their competitors. In other words, it tries to detect differences as closely as possible to human mentality. The most successful algorithms for PSNR test are LSB, PRLSB and the proposed one as can be seen from Table III. The least successful algorithms for PSNR test are F5 and EzStego algorithms.

TABLE IV. CHI-SQUARE TEST EXPERIMENT RESULTS

Image No.	LSB	PRLSB	EzStego	F5	Proposed Algorithm
a	1,49	5,28	182,77	-	7,64
b	1,49	3,27	87,9	-	7,94
c	0,74	2,3	33,53	-	3,98
d	0,17	0,41	5,54	-	0,84
e	1,32	2,3	43,59	-	3,95
f	2,34	4,12	92,33	-	8,18
g	8,55	28,26	341,26	-	58,13
h	1,1	2,21	18,97	-	4,32
i	0,83	2,3	27,76	-	6,9
j	1	2,12	117,93	-	5,66

It can be brought forward that algorithms that produced lower results in Chi-square test have higher withstanding against statistical attacks. From this point of view, It can be seen from Table IV that Ezstego algorithm is inefficient than the others in this field. LSB embedding, PRLSB embedding and the proposed algorithms are more successful than Ezstego algorithm. If we consider that in terms of Chi-square test that is a statistical test; in literature it is indicated that F5 algorithm withstands statistical attacks [43]. Thus, the results of this algorithm were not tested with Chi-square test and were accepted as successful.

TABLE V. CAPACITY TEST RESULTS

Image No.	LSB	PRLSB	EzStego	F5	Proposed Algorithm
a	**75951**	**75951**	25317	9874	25317
b	**88800**	**88800**	29600	11544	29600
c	**156975**	**156975**	52325	20407	52325
d	**749088**	**749088**	249696	97381	249696
e	**98304**	**98304**	32768	12780	32768
f	**115200**	**115200**	38400	14976	38400
g	**10023**	**10023**	3341	1303	3341
h	**133407**	**133407**	44469	17343	44469
i	**101250**	**101250**	33750	13163	33750
j	**125062**	**125062**	41687	16258	41687

As the results of the capacity test, the results with a higher capacity are shown in bold numbers in Table V. When examining these results, it can be seen that LSB embedding and PRLSB embedding algorithms are more successful than the other algorithms. The proposed and Ezstego algorithms follow them. In this regard, F5 algorithm can be considered the most inefficient algorithm. It can be said that the algorithms with the higher valued test results are better than the others. Embedding less or more data does not have any significant meaning in terms of data security. It is only a performance measure for the majority of secret data that can be transferred. As mentioned before the weakest chain of LSB embedding algorithm is embedding the data into LSB of pixel of cover image in an order. This can cause detection and also extraction of secret message easily. From this perspective, low capacity ratio of the proposed algorithm can't be seen as a serious disadvantage since it is not directly related to security. Moreover, let's say that the cover image and stego image have passed into the wrong hands. The computational complexities needed to uncover the hidden text for five algorithms were presented in Table VI. It is clear that malicious persons being busy with more

algorithm complexity to obtain the hidden data is an advantage in terms of secret data embedding.

TABLE VI. ALGORITHMS AND COMPUTATIONAL COMPLEXITY

Algorithm Name	Complexity
LSB	O(n)
PRLSB	$O(n^2)$
EzStego	$O(n^2)$
F5	$O(n^2)$
Proposed Algorithm	$O(3*n^2)$

The computational complexity is the calculation cost of embedding the secret data into a cover image and restoring the embedded data from stego image. When the results in Table VI is examined it can be seen that LSB embedding algorithm has the lower complexity that means that embedding/restoring the secret data takes the lowest time among the other algorithms. It is the advantage when the time is primary concern for image steganography. But, LSB embedding algorithm suffers from being weak against the attacks since the secret data is embedded into the last bit of each pixel starting from the first pixel. At this point, it can be discussed about tradeoff between security and computational complexity. It can't be assumed that the lower computational complexity is always welcome. Of the remaining algorithms, including the proposed algorithm, they have similar computational complexities. As for the proposed algorithm, it can be seen that it is 3 times of the other algorithms. This ratio is acceptable regarding to any size of the secret data since the major value that determines the computational complexity is square of the size of secret data for all algorithms. Thus, the computational complexity of the proposed algorithm is still compatible with other algorithms. It should be noted here that one result of only embedding data into one channel, and meaninglessly distorting the remaining two channels causes this difference in computational complexity for the proposed algorithm. Distorting the last bit of the remaining channels increases the security of the proposed system without scarifying the perceptual transparency since it increases the efforts to find out which channel of related pixel is used to embed the data.

Finally, the algorithms were compared in terms of SSIM. The comparison results are shown in Table VII.

TABLE VII. SSIM TEST RESULTS

Image No.	LSB	PRLSB	EzStego	F5	Proposed Algorithm
a	0,999	0,99999	0,99976	0,98424	0,99999
b	0,999	0,99999	0,99825	0,88851	0,99996
c	0,999	0,99999	0,999	0,99853	0,999
d	0,999	0,99999	0,999	0,99896	0,999
e	0,999	0,99999	0,999	0,98832	0,99999
f	0,999	0,99999	0,999	0,98737	0,99999
g	0,997	0,99991	0,96209	0,90177	0,9997
h	0,999	0,99999	0,999	0,98567	0,99999
i	0,999	0,99999	0,999	0,99357	0,99999
j	0,999	0,99999	0,999	0,98891	0,99999

As the two images become similar, in other words when the differences decrease, SSIM test results approach 1. When the test results in Table VII are analyzed it can be seen that the LSB embedding, PRLSB embedding and proposed algorithms produce results close to 1. Ezstego algorithm, though not in the degree of these algorithms, can produce good results that approach 1. Since F5 algorithm produces the worst results, it's failed for this test.

VII. CONCLUSION

Steganography algorithms are often referenced in the field of data security. The LSB is the center of these algorithms. In this study, a new algorithm based on random indexes produced by chaotic algorithms has been proposed. It can be used on any of colored cover images. It hides the secret data in only one of the color channels, and chooses this channel randomly by using a logistic map based on the chaos theory. It distorts the LSB of other remaining channels so that it is harder to obtain the text within the stego image by unwanted persons.

LSB embedding, PRLSB embedding, and the proposed algorithms yield better results than F5 and EzStego algorithms regarding to PSNR test that is detection of differences between cover and stego images as closely as possible to human mentality. LSB embedding, PRLSB embedding and the proposed algorithms are more successful than Ezstego algorithm regarding to Chi-square test that shows the strength of the algorithm against statistical attacks. When it comes to capacity test, the proposed and Ezstego algorithms follow LSB and PRLSB embedding algorithms. In this regard, F5 algorithm can be considered the most inefficient algorithm. The LSB embedding, PRLSB embedding and proposed algorithms produce results close to 1 regarding to SSIM test that measures the similarity between cover and stego images. For the computational complexity the proposed algorithm is compatible with other algorithms. A slight difference in computational complexity is caused due to only embedding the secret data into one channel and meaninglessly distorting the remaining two channels. Distorting the remaining two channels increases the security of the stego image since any attack against the related stego image requires more efforts in searching and operation to find out which channel of the related pixel is used for embedding.

Applying steganalysis to the proposed and the other four algorithms mentioned in this study is subject to another study. Appropriate techniques of steganalysis must be chosen and then applied to each of the image steganography algorithms including the proposed algorithm. This is the different study area where spatial and transform domain algorithms are compared to each other or compared within the related domain.

The authors of this paper believe that different compression and cryptography algorithms must be studied in order to increase the embedding capacity and security of chaos based steganography algorithms without losing perceptual transparency. When the cryptography techniques are used to encrypt the secret data, the size of the secret data will increase. On the other hand, applications of compression algorithms on the encrypted data will reduce the size of encrypted data. Thus, there will be a tradeoff between capacity and security increases. Different techniques of cryptography and compression fields must be tried to find the optimal combination for image steganography.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and associate editor for their invaluable comments for improving the paper.

REFERENCES

[1] A. Beimel, "Secret-Sharing Schemes: A Survey," in Coding and Cryptology, 2011, pp. 11-46. doi:10.1007/978-3-642-20901-7_2

[2] A. Shamir, "How to Share a Secret," Commun. ACM, vol. 22, no. 11, pp. 612-613, Nov. 1979. doi:10.1145/359168.359176

[3] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: Survey and analysis of current methods," Signal Processing, vol. 90, no. 3, pp. 727-752, Mar. 2010. doi:10.1016/j.sigpro.2009.08.010

[4] C. Kurak and J. McHugh, "A cautionary note on image downgrading," in [1992] Proceedings Eighth Annual Computer Security Application Conference, 1992, pp. 153-159. doi:10.1109/CSAC.1992.228224

[5] I. S. Moskowitz, G. E. Longdon, and L. Chang, "A new paradigm hidden in steganography," in Proceedings of the 2000 workshop on New security paradigms, 2001, pp. 41-50. doi:10.21236/ADA462825

[6] T. Sharp, "An Implementation of Key-Based Digital Signal Steganography," in Information Hiding, 2001, pp. 13-26. doi:10.1007/3-540-45496-9_2

[7] E. Kawaguchi and R. Eason, "Principle and applications of BPCS-Steganography," in Principle and applications of BPCS-Steganography, Boston,USA, 1998, vol. 3528, pp. 464-473.

[8] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," IBM Systems Journal, vol. 35, no. 3.4, pp. 313-336, 1996. doi:10.1147/sj.353.0313

[9] I. S. Moskowitz and N. F. Johnson, "A detection study of an NRL steganographic method," NAVAL RESEARCH LAB WASHINGTON DC, 2002. doi:10.21236/ADA405340

[10] M. Noto, "MP3Stego: Hiding text in MP3 files," Sans Institute, p. 5, 2001.

[11] N. F. Johnson and S. Katzenbeisser, "A survey of steganographic techniques," in Information hiding, 2000, pp. 43-78.

[12] L. Zhi, S. A. Fen, and Y. Y. Xian, "A LSB steganography detection algorithm," in Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on, 2003, vol. 3, pp. 2780-2783. doi:10.1109/pimrc.2003.1259249

[13] Jessica Fridrich and Miroslav Goljan, "Digital image steganography using stochastic modulation," 2003, vol. 5020, pp. 5020-12. doi:10.1117/12.479739

[14] R. O. El Safy, H. H. Zayed, and A. El Dessouki, "An adaptive steganographic technique based on integer wavelet transform," in 2009 International Conference on Networking and Media Convergence, 2009, pp. 111-117. doi:10.1109/ICNM.2009.4907200

[15] C.-C. Chang, T.-S. Chen, and L.-Z. Chung, "A steganographic method based upon JPEG and quantization table modification," Information Sciences, vol. 141, no. 1, pp. 123-138, Mar. 2002. doi:10.1016/S0020-0255(01)00194-3

[16] R. Chu, X. You, X. Kong, and X. Ba, "A DCT-based image steganographic method resisting statistical attacks," in Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on, 2004, vol. 5, pp. V-953. doi:10.1109/icassp.2004.1327270

[17] M. Hussain and M. Hussain, "A survey of image steganography techniques," 2013.

[18] C.-K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," Pattern Recognition, vol. 37, no. 3, pp. 469-474, Mar. 2004. doi:10.1016/j.patcog.2003.08.007

[19] Y. K. Lee and L. H. Chen, "High capacity image steganographic model," IEE Proceedings - Vision, Image and Signal Processing, vol. 147, no. 3, pp. 288-294, Jun. 2000. doi:10.1049/ip-vis:20000341

[20] R.-Z. Wang, C.-F. Lin, and J.-C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm," Pattern Recognition, vol. 34, no. 3, pp. 671-683, Mar. 2001. doi:10.1016/S0031-3203(00)00015-7

[21] S.-H. Liu, T.-H. Chen, H.-X. Yao, and W. Gao, "A variable depth LSB data hiding technique in images," in Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on, 2004, vol. 7, pp. 3990-3994. doi:10.1109/icmlc.2004.1384536

[22] C. E. Shannon, "Communication theory of secrecy systems," The Bell System Technical Journal, vol. 28, no. 4, pp. 656-715, Oct. 1949. doi:10.1002/j.1538-7305.1949.tb00928.x

[23] W. Wong, L. Lee, and K. Wong, "A Modified Chaotic Cryptographic Method," in Communications and Multimedia Security Issues of the New Century: IFIP TC6 / TC11 Fifth Joint Working Conference on Communications and Multimedia Security (CMS'01) May 21-22, 2001, Darmstadt, Germany, R. Steinmetz, J. Dittman, and M. Steinebach, Eds. Boston, MA: Springer US, 2001, pp. 123-126. doi:10.1007/978-0-387-35413-2_11

[24] A. Kanso and N. Smaoui, "Irregularly decimated chaotic map (s) for binary digits generations," International Journal of Bifurcation and Chaos, vol. 19, no. 04, pp. 1169-1183, 2009. doi:10.1142/S0218127409023573

[25] A. Kanso, H. Yahyaoui, and M. Almulla, "Keyed hash function based on a chaotic map," Information Sciences, vol. 186, no. 1, pp. 249-264, 2012. doi:10.1016/j.ins.2011.09.008

[26] A. Kanso, "Self-shrinking chaotic stream ciphers," Communications in nonlinear science and numerical simulation, vol. 16, no. 2, pp. 822-836, 2011. doi:10.1016/j.cnsns.2010.04.039

[27] X. Wu and Z.-H. Guan, "A novel digital watermark algorithm based on chaotic maps," Physics Letters A, vol. 365, no. 5-6, pp. 403-406, 2007. doi:10.1016/j.physleta.2007.01.034

[28] S. Behnia, M. Teshnehlab, and P. Ayubi, "Multiple-watermarking scheme based on improved chaotic maps," Communications in Nonlinear Science and Numerical Simulation, vol. 15, no. 9, pp. 2469-2478, Sep. 2010. doi:10.1016/j.cnsns.2009.09.042

[29] M. Keyvanpour and F. Merrikh-Bayat, "An Effective chaos-based image watermarking scheme using fractal coding," Procedia Computer Science, vol. 3, pp. 89-95, Jan. 2011. doi:10.1016/j.procs.2010.12.016

[30] L. Yu, Y. Zhao, R. Ni, and T. Li, "Improved Adaptive LSB Steganography Based on Chaos and Genetic Algorithm," EURASIP Journal on Advances in Signal Processing, vol. 2010, no. 1, p. 876946, Jun. 2010. doi:10.1155/2010/876946.

[31] A. Kanso and H. S. Own, "Steganographic algorithm based on a chaotic map," Communications in Nonlinear Science and Numerical Simulation, vol. 17, no. 8, pp. 3287-3302, Aug. 2012. doi:10.1016/j.cnsns.2011.12.012

[32] R. Roy, A. Sarkar, and S. Changder, "Chaos based Edge Adaptive Image Steganography," Procedia Technology, vol. 10, pp. 138-146, Jan. 2013. doi:10.1016/j.protcy.2013.12.346

[33] S. Ahadpour and M. Majidpour, "Image Steganography Using Discrete Cross-Coupled Chaotic Maps," 2013.

[34] M. Ghebleh and A. Kanso, "A robust chaotic algorithm for digital image steganography," Communications in Nonlinear Science and Numerical Simulation, vol. 19, no. 6, pp. 1898-1907, Jun. 2014. doi:10.1016/j.cnsns.2013.10.014

[35] M. Y. Valandar, P. Ayubi, and M. J. Barani, "A new transform domain steganography based on modified logistic chaotic map for color images," Journal of Information Security and Applications, vol. 34, pp. 142-151, Jun. 2017. doi:10.1016/j.jisa.2017.04.004

[36] G. Sathishkumar, D. N. Sriraam, and others, "Image encryption based on diffusion and multiple chaotic maps," arXiv preprint arXiv:1103.3792, 2011. doi:10.5121/ijnsa.2011.3214

[37] B. Y. Ryabko, V. S. Stognienko, and Y. I. Shokin, "A new test for randomness and its application to some cryptographic problems," Journal of Statistical Planning and Inference, vol. 123, no. 2, pp. 365-376, Jul. 2004. doi:10.1016/S0378-3758(03)00149-6

[38] A. Westfeld and A. Pfitzmann, "Attacks on Steganographic Systems," in Information Hiding, 2000, pp. 61-76. doi:10.1007/10719724_5

[39] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004. doi:10.1109/TIP.2003.819861

[40] A. Horé and D. Ziou, "Image quality metrics: PSNR vs. SSIM" in 20th International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, 2010, pp 2366–2369 doi:10.1109/ICPR.2010.579

[41] H. Gupta, R. Kumar and S. Changlani, "Enhanced data hiding capacity using LSB-based image steganography algorithm." International Journal of Emerging Technology and Advanced Engineering, 3(6), pp. 212-214, 2013

[42] N. P. Kamdar, D. G. Kamdar and D. N. Khandhar, "Performance evaluation of lsb based steganography for optimization of psnr and mse." Journal of information, knowledge and research in electronics and communication engineering, 2(2), pp. 505-509, 2013

[43] A. Westfeld, "F5—a steganographic algorithm", In International workshop on information hiding, vol 2137, Springer, Berlin, Heidelberg. pp. 289-302, doi:10.1007/3-540-45496-9_21