

Multimedia Steganography Based on Least Significant Bit (LSB) and Duffing map

Jinan N. Shehab¹, HaraaRaheemHatem²

^{1,2}Communication Department, College of Engineering, University of Diyala

ABSTRACT

This paper presents hiding the text or image (secret information) inside other image (cover image) based on Least Significant Bits (LSB). The position of characters in original secret text and the position of pixels in original secret image have been changed by Duffing map (random number generator). The fundamental idea is to insert the secret message (text, gray image and color image) in the least significant bits of the cover image (gray or color image). This actually works because the Human Visual System (HVS) is not sensitive enough to pick out changes in color. The experiments and comparative studies show that the algorithms are characterized by many features of the ability of hiding huge data, and then the ability of extracting secret message without errors. Beside the return image, has efficacies (to human acquaintance) according to peak signal to noise ratio (PSNR) and mean square error (MSE), also retain both the explicitness and the characteristics of the both secret message and cover image.

Keyword: steganography, LSB algorithm, image steganography, text steganography, Duffing map.

اخفاء الوسائط المتعدده باستخدام البت الاقل وزنا في حسابات الارقام واستخدام مولد عشوائي للاعداد

جنان نصيف شهاب¹، حراء رحيم حاتم²

^{1,2}مدرس مساعد، كلية الهندسه جامعة ديالى

الخلاصه:

يقدم هذا العمل اخفاء نص او صورته داخل صورته بالاعتماد على البت الاقل وزنا بعد تغيير مواقع الحروف في النص الاصيلي ومواقع وحدات الصورة في صورته المراد اخفائها باستخدام المولد العشوائي للاعداد (Duffing map).

الفكره الاساسيه في هذا العمل هو ادخال النص او صورته (الملونه، الرماديه) في البتات الاقل وزنا في صورته الاصيليه (الملونه، الرماديه). يعتبر هذا العمل حقيقي لان العين البشريه لاتتحمس للتغيرات الطفيفه في الالوان. لقد بينت التجارب والدراسات ان الخوارزميات المستخدمه تتحدد صفاتها عن طريق قابليتها في اخفاء عدد كبير من البيانات وعن طريق قدرتها في استرجاع الرساله الاصيليه بدون اخطاء.

أمتلكت الصورة المسترجعه كفاءاتعالية (إلالتعارفالبشري) بالاعتماد على نسبةالضوضاء (PSNR) ومعدلمربعالخطأ (MSE). ان البرنامج المقترح وفر حمايه عاليه للنص والصوره السريه لحاجه المسترجع الى المفتاح الاصيلي وفي حالة حدوث تغير بسيط في المفتاح فستتعدم فرصه استرجاع المعلومات الاصيليه. لقد تم انجازالبحث باستخداملغة .MATLAB

The development of computer and expanding its use in different areas of life and work, the issue of information security has become increasingly important, security becomes increasingly important for many applications. One of the grounds discussed in information security is the exchange of information through the cover media, for that; different methods such as, steganography, coding, watermarking ... etc have been used to improve image security [1].

Steganography is the art of secret communication or the science of invisible communication. It is nothing more mechanism to conceal message (secret object) inside another innocuous message (cover object) in a way that nobody except the recipient (who must know the technique used) can detect there is a second (secret) message present [2]. In the face of, there are many different carrier file format (cover) can be used but digital image are the most popular because hold large amount of data and their frequency on the internet [3].

In this paper, will take one of the methods of steganography it is LSB, it's used to hide text in image and gray image in cover image. To add more security, the data to be hidden is permuted with a key created by Duffing map and then the new shuffling message is

embedding into cover image. To extract the hidden information, one should have the same key using in the transmitter to extract the message. This work includes two algorithms; hiding text in an image (gray-scale plus color image) and hiding gray-image in an image (gray-scale plus color image). In these algorithms, interest has been expressed to the quality of the extracted secret information (reconstructed message quality) beside the quality of the stego-image, compared with the original cover.

1- Least Significant Bit (LSB) substitution method

The LSB is a very popular way of embedding secret messages with simplicity. The fundamental idea here is to insert the secret message in the least significant bits of the cover images. This actually works because the human visual system is not sensitive enough to pick out changes in color (whether gray or color) and digital covers have a large number of redundant bits.

A basic algorithm for **LSB** substitution is to take the first N cover pixels where N is the total length of the secret message (for text and image where ($N=R \times C$ where R row and C column numbers in secret image)) that is to be embedded in bits. After that every pixel's last bit in cover image will be replaced by one of the message bits [4,5].

2- Duffing Map (also called as Holmes map)

A two-dimensional discrete-time nonlinear dynamical system was proposed by German electrical engineer Georg Duffing [6]. As a simplified model of the Poincare map for the Duffing map module is given by:-

$$\left. \begin{aligned} X_{n+1} &= Y_n \\ Y_{n+1} &= -bX_n + aY_n - Y_n^3 \end{aligned} \right\} \quad (1)$$

The map depends on the two [constants](#) or parameters a and b , this map is shown in Figure 1. The diagram is a strange attractor popularly known as the Duffing attractor.

4- The Steganography System Procedure

First in these systems, the cover image should be selected carefully like choosing the cover with low details (as shown in Figure 2, cover image with low details all have the same size 512×512) so when the information in the pixels is replaced with another information, the cover image will not have a noticeable degradation. In this work, the procedure of steganography divided in two sides:

4-1 Embedded Side

Figure 3 shows the stages involved in the sending process. Each stage will be briefly discussed below:

Step 1. Preparation of The Cover Image: Transform 2-D image ($R \times C$) into 1-D image (N).

Step 2. Preparation of The Secret Message: In this algorithm, a secret text is being reading and then transform each character into equivalent number according to the American Standard Code for Information Interchange (ASCII). From other side the secret image transform from 2-D into 1-D.

Step 3. Shuffling by Duffing Map: this contain many sub-steps;

Set the key (initial conditions $X(0) = 0.1$, $Y(0) = 0.003$ and parameters $a = 2.75$, $b = 0.15$) in acceptable intervals to generate random number. the real value result from Duffing map is modified to integer value between $(0, 255)$ [6].

$$\left. \begin{aligned} X_D(n) &= \text{mod}(\text{floor}(X_H(n) \times 10^{15}), 256) \\ Y_D(n) &= \text{mod}(\text{floor}(Y_H(n) \times 10^{15}), 256) \end{aligned} \right\} \quad (2)$$

1. Conduct the function “Sort” on X_D and Y_D for constructing scrambling index array I_1 and I_2 with dimension (same dimension of the secret text or image)arranged in ascending order.
2. Rearrangement of the decimal value on secret message according to the sort of the random key as shown in the Figure (4).

Step 4. Proposed Embedding Algorithms

A. Hiding Text in an Image:-In this algorithm, a secret text message is embedded in a cover image, as shown in Figure(3-A) . The algorithms step represented by:

- 1- After shuffling text, transform each decimal value into binary number (8-bits/numbers).
- 2- Convert the cover image into binary number (if gray image (8-bits/pixel)and (24-bits /pixels) if color image is used).
- 3- Replace the value of last bit in every pixel (in cover image) by the value of bit from secret text, then every pixel's last bit in cover image will be replaced by one of the message bits. As show in algorithm and Table 1.
- 4- Transform results back from binary to decimal to get stego-image

B. Hiding Image in an Image: In this algorithm, a secret image will be hidden in a cover image as shown in Figure(3-B).The steps for this algorithms are:

- 1- After shuffle secret image by using Duffing map. Each pixel in secret image represented by (8-bits/pixel (gray-image)).
- 2- Hidden each bit from secret image in the last bit from each pixel in cover image according to LSB algorithm.
- 3- Transform back from binary to decimal and then from 1-D to 2-D to get stego image.

4-2 Reconstructed Side

To extract the secret information, the receiver need stego-image and the secret key (initial conditions $(X(0), Y(0))$ and parameters (a,b)) of Duffing map . The extracting algorithm is the inverse of the embedding algorithms , as shown in Figure (5):

A- Extract Text from Image

As in extracting text from image as shown in Figure (5-A) ,the same steps will be followed:-

1. Convert the steg image from 2-D into 1-D and then convert each pixel to binary number (8-bit/pixel).
2. Take the last bit from each pixel to construct the secret text (binary)
3. Transform from binary to decimal value.
4. Return the value to their original position depending on initial condition $(X(0)$ and $Y(0))$ and parameters (a,b) from Duffing map.

5. After return every value to its position transforms each value into character according to ASCII.

B- Extracting Image from Image

Figure 5-B shows that this process will be done by following steps:

1. Convert the steg image from 2-D into 1-D and then convert each pixel to binary number (8-bit/pixel).
2. Take the last bit from each pixel to construct the secret image (binary)
3. Transform from binary to decimal value.
4. Return each pixel to its original position the value depending on initial condition (X(0) and Y(0)) and parameters (a,b) from Duffing map.
5. After return every value to its position transform from 1-D into 2-D to construct secret image.

5-Numerical Simulation Results

There are many tests that can be used to measure the quality and security of the image:-

5-1 Peak-Signal-to-Noise-Ratio (PSNR)

According to the Human Visual System (HVS), some amount of distortion between the original image and the modified one is allowed. The Peak Signal-to-Noise Ratio known as PSNR is used as the scale for image quality (which computes the peak signal-to-noise ratio) between the original image and stego image [7]. PSNR is usually measured in dB. To compute the peak signal to noise ratio, then:-

$$PSNR(dB) = 10 \log_{10} \frac{P^2}{MSE} \quad (3)$$

Where; P is the maximum pixel value. Also, the Mean Square Error (MSE) which measures the cumulative Mean Square Error between the original and the stego image. The MSE is defined

$$MSE = \frac{1}{R \times C} \sum_{i=0}^{R-1} \sum_{j=0}^{C-1} [X(i,j) - \hat{X}(i,j)]^2 \quad (4)$$

as:

Where: R : number of pixel in rows, C : number of pixel in columns, i and j : row and column numbers, $X(i,j)$: original image and $\hat{X}(i,j)$: stego image.

By using Matlab program, the simulation result for the proposed method are:-

1. Hiding a text into a gray image also hiding text in color image. The implementation results to hide three different text size and result of PSNR (between original image and Stego-image) for both in gray and color image are shown in Table(2).
2. Hiding a gray image into a gray image and a gray image into a color image and then color image into color image. The implementation results can be seen in Table (3).

From Tables 2,3 &4, the result of PSNR for using color image as cover are higher compare with gray image as cover image. The size of the color image is larger than the size of the gray image that is the problem, for that gray image has been used as cover image in this paper. Also we noted that PSNR is reduced when secret information (text or image) size increased because of more pixel in cover image is changed (more noise).

6- Histogram Analysis

The histogram of the cover image and the stego-image are found to show that the statistical properties of the cover image are not affected by changing one bit in some pixels [2]. Therefore, if the histogram of the cover is nearly equal to the histogram of the steg- image, this means that the proposed system is good enough to avoid the attackers. The three types of the images used in our algorithm. Figure 6 represented one example (Baby.jpg) of the cover and stego-images histograms, we noted that histogram of image before hiding information is the same that after hiding information because of the small change in some pixels don't effect on the histogram of the cove image as shown in Figure 6.

7- Key Space Analysis

Key space size is the total number of different keys that are used in the encoding .Here , the possible key size is 10^{30} keys for system . Exhaustive key search will take 2^d operations to succeed, where d is the key size in bits [2]. Any attacker simply tries all keys, one by one, and checks whether the given secret image. Therefore, the combinations of the parameters and initial conditions are large enough to prevent such exhaustive search.

7-1 Key Sensitivity Test

The key sensitivity is the degree of the changes in the encoding image caused by a tiny change in secret key [7,8], as shown in Figure (7).From this test the proposed algorithm is very sensitivity to tiny change in key= 10^{-15} , then only by using the exact key can return the original secret message.

8-Conclusion

The simulation results show that, the proposed algorithm has high HVS for extracted secret message, also the stego-image is obtained with very close properties to the original cover image according to PSNR, MSE, HVS, and histogram tests, so it is so difficult to distinguish between

them. Using Duffing map to encoding secret message gives large enough key space $=10^{30}$ and very sensitive to the secret keys.

REFERENCES

- [1] ShashikalaChannalli And Ajay Jadhav, "Steganography An Art Of Hiding Data", Sinhgad College Of Engineering, Pune,ShashikalaChannalli Et Al /International Journal On Computer Science And Engineering Vol.1(3), 137-141,2009.
- [2] ZaynabNajeebAbdulhameed,"High Capacity Steganography Based On Chaos And Contourlet Transform For Hiding Multimedia Data",M.Sc. Thesis, Department of Electronics & CommunicationsEngineering ,University of AL-Mustansiriya2014.
- [3] Morkel , Eloff , Olivier" An Overview Of Image Steganography" Information and Computer Security Architecture (ICSA) Research Group Department of Computer Science, University of Pretoria, Pretoria, South Africa,2005.
- [4] FahimIrfanAlam "An Investigation into Encrypted Message Hiding Through Images Using LSB", International Journal of Engineering Science and Technology (IJEST), 2011.
- [5] Bhavana.S, and K.L.Sudha" Text Steganography Using LsbInsertion Method Along With Chaos Theory", International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.2, No.2, April 2012.
- [6] Jing Xia, SuwenZheng, BaohongLv, CaihongShan"Harmonic Solutions of Duffing Equation with Singularity via Time Map", Applied Mathematics, 2014, 5,1528-1534.
- [7] Shreenandan Kumar, SumanKumari, SuchetaPatro, TusharShandilya and Anuja Kumar Acharya"Image Steganography using Index based Chaotic Mapping", International Conference on Distributed Computing and Internet Technology (ICDCIT), 2015.
- [8] Rosanne English," Comparison of High Capacity Steganography Techniques", IEEE , International Conference of Soft Computing and Pattern Recognition, 978-1-4244-7896-2,2010,(IVSL).

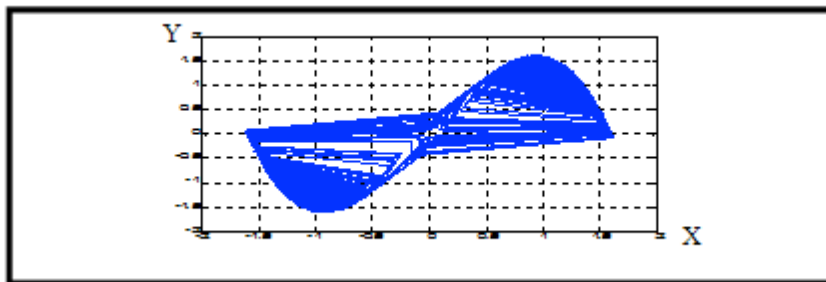


Figure (1) Duffing map attractor.



Figure (2) cover images.

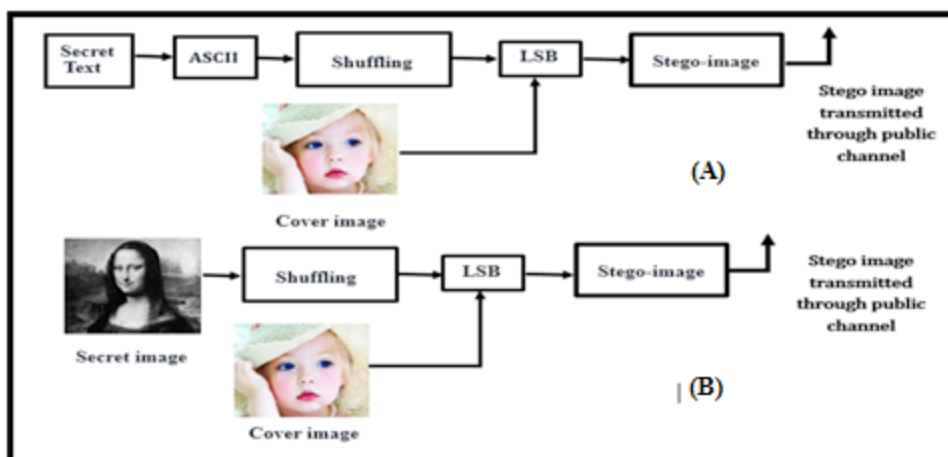


Figure (3) proposed of embedded system.

A: embedded system for Text.

B: embedded system for gray level image.





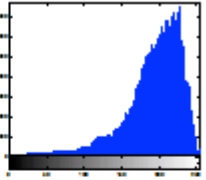

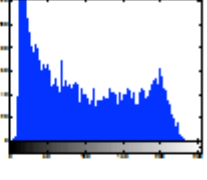

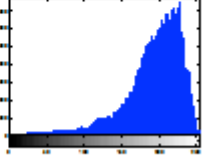

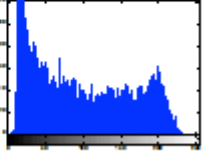
			
University of Diyala/ College of Engineering/ Communications Department/ paper title(Multimedia Steganography Based on Least Significant Bit)	University of Diyala/ College of Engineering/ Communications Department/ paper title(Multimedia Steganography Based on Least Significant Bit)	tynttderiaelSlen rtDanSosm nttrgmeecdoatinLeinaeegafig siM nil)Cioa/no aCeaoipntapocvryEheDtseuem / irigm Bi/i pf ieUty paitaBnlngls	
Image	Histogram	Secret image	Histogram
			
Cover image	Befor hiding information	Before stego	Before embedding
			
Stego-image	After hiding information	After reconstruction	After reconstruction

Figure (6) Cover and Stego -Images Histograms.











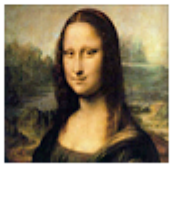
Cover image	Secret image	Stego-image	Extraction Secret image	PSNR
				56.3197
				61.1079
				56.3367

Figure (7) Sensitivity Tests of Keys A) Original Image B) Decoding Image by Using Original Key and C) Decoding Image by Using Neighbored Key.

Table (1) Embedding Steps (Replaced Last Bit in Every Pixel by Bit from Secret Message).




Data	Decimal value(ASCII)	Binary transform		
Secret message (as letter (A))	65	0 1 0 0 0 0 0 1		
Cover image (gray-scale)	(7-pixel in 1-row)	Binary transform	Embedding steps	Number of changing bits
	1. P1=241	1. 1111000 <u>1</u>	1. 1111000 <u>1</u>	3-bits in 8-pixels to hide one character (8-bits)
	2. P2=241	2. 1111000 <u>1</u>	2. 1111000 <u>0</u>	
	3. P3= 239	3. 1110111 <u>1</u>	3. 1110111 <u>0</u>	
	4. P4=237	4. 1110110 <u>1</u>	4. 1110110 <u>0</u>	
	5. P5=234	5. 1110101 <u>0</u>	5. 1110101 <u>0</u>	
	6. P6=230	6. 1110011 <u>0</u>	6. 1110011 <u>0</u>	
	7. P7=227	7. 1110001 <u>1</u>	7. 1110001 <u>1</u>	
	8. P8= 224	8. 1110000 <u>0</u>	8. 1110000 <u>0</u>	

Table (2) Hiding Text in Image and PSNR to each State.

Text	PSNR(<i>girl</i>)			
	Gray image	Color image	Gray Stego- image	Color Stego-image
Red	92.8078	97.121	 PSNR=75.8848	 PSNR=80.1404
Communication Department	83.9617	87.9367		
University of Diyala/ College of Engineering/ Communications Department/ paper title(Multimedia Steganography Based on Least Significant Bit)	75.8848	80.1404		

Table(3) Results of Hiding Image in Image also PSNR for Each State.

Table(4) PSNR for Different Size of Secret Image.

Size of Secret image	Image name (image size)					
	Little girl.jpg(512×512)		Chicken.jpg(512×512)		Baby.jpg(512×512)	
	PSNR(gray in gray image≈ color in color image)	PSNR (gray in color)	PSNR(gray in gray image≈ color in color image)	PSNR (color)	PSNR(gray in gray image≈ color in color image)	PSNR (color)
50×50	62.3585	67.1441	62.3680	67.1353	62.3109	67.1379
100×100	56.3464	61.1036	56.3276	61.0907	56.3197	61.1079
150×150	52.8151	57.5705	52.8103	57.5844	52.8085	57.6048
200×200	50.4016	55.0618	50.4861	55.0774	50.3459	55.0841
250×250	48.2149	53.1297	48.2107	53.1449	48.2187	53.1468