Building a Reliable Steganography System Using Random Key in a spatial Domain of Image

Amer J. Sadiq* ***Zainab S. Qasim****

Received 20, December, 2012
Accepted 11, March, 2014

Abstract:

With time progress importance of hiding information become more and more and all steganography applications is like computer games between hiding and extracting data, or like thieves and police men always thieve hides from police men in different ways to keep him out of prison. The sender always hides information in new way in order not to be understood by the attackers and only the authorized receiver can open the hiding message.

This paper explores our proposed random method in detail, how chooses locations of pixel in randomly , how to choose a random bit to hide information in the chosen pixel, how it different from other approaches, how applying information hiding criteria on the proposed project, and attempts to test out in code, and in practice, through example.

Key words: information hiding, Image processing.

1. Introduction

There are important distinctions to be made between cryptography and steganography. Cryptography changes the message such that even if it is overheard by a third party, it would be unintelligible. Steganography, on the other hand, involves hiding message in such a way that the casual observer should not be able to detect the hidden information. Steganography is a good choice in situations where it is a priority to disguise the occurrence of communication. Seemingly meaningless data can contain complex details, maps, or text. Steganography is often combined with cryptography to provide an additional layer of security.

Both steganographic and watermarking methods are lie under the information hiding domain. The digital information hiding can be characterized by utilizing the theories of digital communication system [1].

The following is the block diagram for traditional information hiding system.

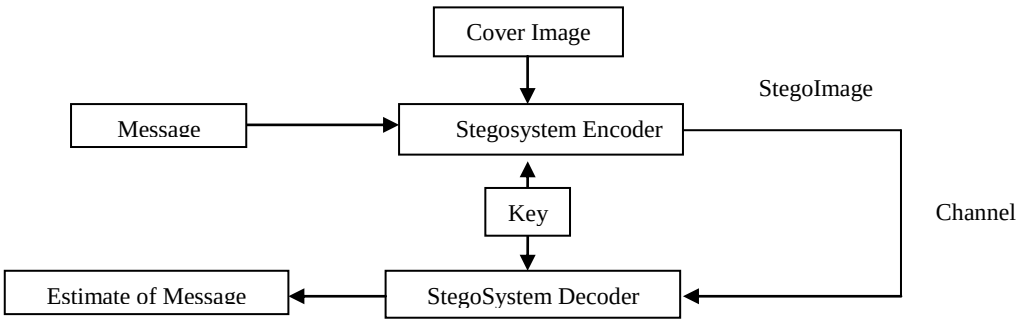

Fig .1 the block diagram for traditional information hiding system

*Baghdad university-College of Science for Women-Computer Department

2-Characterizing Data Hiding Techniques

Steganographic techniques embed a message inside a cover; various features characterize the strengths and weaknesses of the methods. The relative importance of each feature depends on the application [2].

Hiding Capacity: Hiding capacity is the size of information that can be hidden relative to the size of the cover. A larger hiding capacity allows the use of a smaller cover for a message of fixed size, and thus decreases the bandwidth required to transmit the stego-image.

Perceptual Transparency: The act of hiding the message in the cover necessitates some noise modulation or distortion of the cover image. It is important that the embedding occur without significant degradation or loss of perceptual quality of the cover. In a secret communications application, if an attacker notices some distortion that arouses suspicion of the presence of hidden data in a stegoimage,the steganographic encoding has failed even if the attacker is unable to extract the message. Preserving perceptual transparency in an embedded watermark for copyright protection is also of paramount importance because the integrity of the original work must be maintained [3].

For applications where the perceptual transparency of embedded data is not critical, allowing more distortion in the stego-image can increase hiding capacity, robustness, or both.

Robustness: Robustness refers to the ability of embedded data to remain intact if the stego-image undergoes transformations, such as linear and non-linear filtering, addition of random noise, sharpening or blurring, scaling and rotations, cropping or decimation, lossy compression, and conversion from digital to analog form and then reconversion back to digital form (such as in the case when a hard copy of a stego-image is printed and then a digital image is formed by subsequently scanning the hardcopy.) Robustness is critical in copyright protection watermarks because pirates will attempt to filter and destroy any watermarks embedded in images. Anti-watermarking software is already available on the Internet and have been shown effective in removing some watermarks. These techniques can also be used to destroy the message in a stego-image [4].

Tamper Resistance: Beyond robustness to destruction, tamper-resistance refers to the difficulty for an attacker to alter or forge a message once it has been embedded in a stego-image, such as a pirate replacing a copyright mark with one claiming legal ownership. Applications that demand high robustness usually also demand a strong degree of tamper resistance. In a copyright protection application, achieving good tamper resistance can be difficult because a copyright is effective for many years and a watermark must remain resistant to tampering even when a pirate attempts to modify it using computing technology decades in the future [4].

Other Characteristics: Computational complexity of encoding and decoding is another consideration and individual applications may have additional requirements. For example, for a copyright protection application, a watermark should be resistant to collusion attacks where many pirates work together to identify and destroy the mark [4].

PSNR: Developers and implementers of lossy image compression methods need a standard metric to measure the quality of reconstructed images compared with the original ones. The better a reconstructed image resembles the original one, the bigger should be

the value produced by this metric. Such a metric should also produce a dimensionless number, and that number should not be very sensitive to small variations in the reconstructed image.

A common measure used for this purpose is the *peak signal to noise ratio* (PSNR). It is familiar to workers in the field, it is also simple to calculate, but it has only a limited, approximate relationship with the perceived errors noticed by the human visual system [5].

This is why higher PSNR values imply closer resemblance between the reconstructed

and the original images, but they do not provide a guarantee that viewers will like the

reconstructed image.

Denoting the pixels of the original image by *Pi* and the pixels of the reconstructed

image by *Qi* (where $1 \leq i \leq n$), we first define the *mean square error* (MSE) between

the two images as

$$\mathbf{MSE} = 1 / n \sum_{k=1}^{n} (\mathbf{Pi} - \mathbf{Qi})^2 \quad \ldots(1)$$

It is the average of the square of the errors (pixel differences) of the two images. The *root mean square error* (RMSE) is defined as the square root of the MSE, and the PSNR is defined as

$$\mathbf{PSNR} = 20 \log 10 \frac{\max i|Pi|}{\mathbf{RMSE}} \quad \ldots.(2)$$

The absolute value is normally not needed, since pixel values are rarely negative. For a bi-level image, the numerator is 1. For a grayscale image with eight bits per pixel, the numerator is 255. For color images, only the luminance component is used.

Greater resemblance between the images implies smaller RMSE and, as a larger PSNR. The PSNR is dimensionless, since the units of both numerator and denominator are pixel values. However, because of the use of the logarithm, we say that the PSNR is expressed in *decibels* (dB). The use of the logarithm also implies less sensitivity to changes in the RMSE. For example, dividing the RMSE by 10 multiplies the PSNR by 2. Notice that the PSNR has no absolute meaning. It is meaningless to say that a PSNR of, say, 25 is good. PSNR values are used only to compare the performance of different lossy compression methods or the effects of different parametric values on the performance of an algorithm. The MPEG committee, for example, uses an informal threshold of PSNR = 0.5 dB to decide whether to incorporate a coding optimization, since they believe that an improvement of that magnitude would be visible to the eye.

Typical PSNR values range between 20 and 40. Assuming pixel values in the range [0, 255], an RMSE of 25.5 results in a PSNR of 20, and an RMSE of 2.55 results in a PSNR of 40. An RMSE of zero (i.e., identical images) results in an infinite (or, more precisely, undefined) PSNR. An RMSE of 255 results in a PSNR of zero, and RMSE values greater than 255 yield negative PSNRs [5].

3-The proposed system method

In this section, we explore how proposed method hide information into target content and retrieve information from the target content without damaging it. With our method a key is used as password and multiply it by series of numbers in order to get the status "ab" not equal to "ba" and then extract sub key from that key; this sub key used as seed for random function to determine which bit will be chosen for hide information.

So the receiver use the same key and sub key that used by the sender to determine the bit used for hide, For the applications for digital watermarking and steganography, this extraction keys must be shared among sender and receiver in order to extract a proper hidden bit codes from the target content. Considering the difficulties for secret key transportation, this method should be applied in situations where the sender and the receiver are same person or use certification authorities to assure the integrity of the key.

In this project we hide the text in image, by random choice used for the pixels within the image, so we will use a function for this purpose to generate a random numbers.

This function working by entering as input parameters the start and end points where the start point is the first pixel in the image and the end point is the size of image minus 3(the size of image computed by image width * image height, we use size minus 3 because the last 3 pixels of the image size booked up to hide the length of the text which we hide in the image).

4- The hiding process of the proposed system

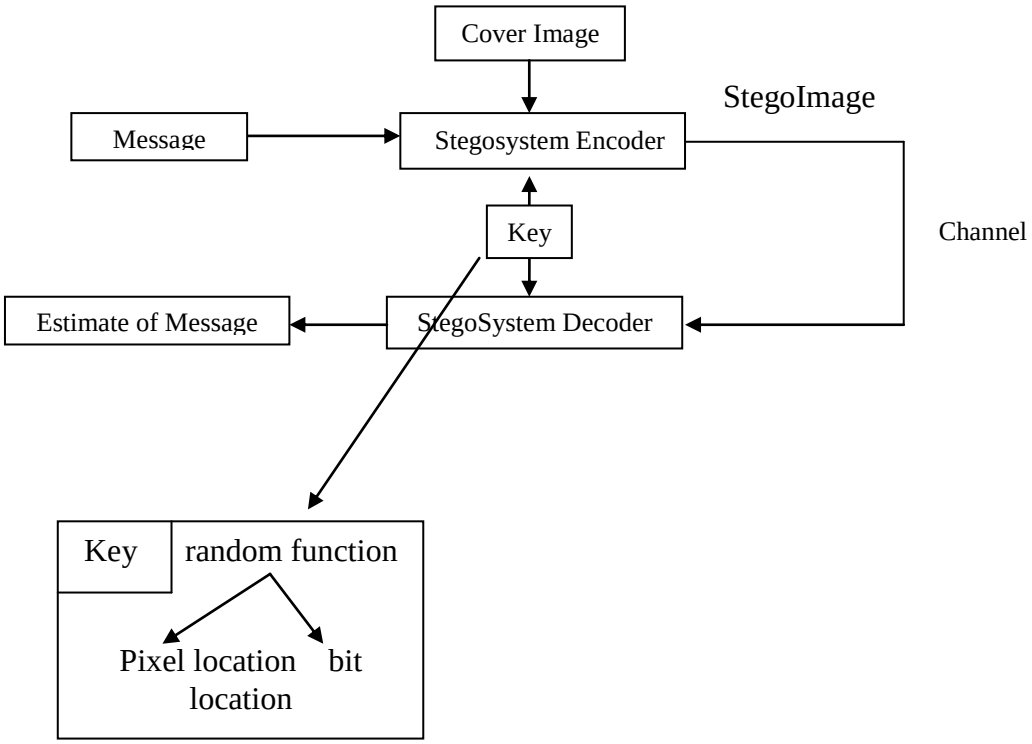

Fig. 2 Block diagram of our proposed system

The last three pixels are used for the length (because the length is an integer number and it represented by 2 bytes so we used the last 3 pixels 2 bit from each color in the selected pixel).

A color image (RGB) is used that means each color has one byte, our key for hiding information and extracting information in and from image work as following:

The sender and receiver deal with one password in order to achieve authentication for sender and receiver, also we add a new feature to the password, we derive the key for hiding

and extracting information in our proposed system from that password.

To generate the key from the password, the password should be in numeric format, which depends on

1. The ASCII of each character in the password.
2. The position of each character in the password.

For example, the numeric value which obtained from "Ahmed" is not equal to "Hamed" and so on. The key is obtained from the relation from the ASCII of the character and its position in the password, therefore any other password which contains the same characters but in different sequence will not produce the same value of the key. So, it's not possible for everyone who is trying to produce the same key without knowing the password that agreed-upon by the sender and receiver.

Random function of our proposed system has three inputs:

1. Start point (first pixel in image).
2. End point (size of image-3)
3. The key (obtained from password).

According to this function the pixel will be chosen randomly.

As it's known, each character is represented by 8 bit, so we need a 3 pixels for each character in our text, because each pixel consists of 3 colors (RGB) and each color is represents by 8 bit and we will use one bit only from each color in least significant bit to hide one bit of character's bits.

After we chose the pixel randomly, we will chose bit from each color of this pixel randomly and the range of choosing is one bit among last three bits of each color of the pixel and this choice of bit is also depends on the password. Therefore it's impossible for anyone to uncover the hidden text without knowing the password and the hiding algorithm.

5-Hiding Algorithm

1. Start.
2. Read the Secrete message
3. Read the Password
4. Convert the Password from textual form to the numeric form and obtain from it an integer value which will be used as a seed for generating the random numbers for hiding position.
5. Call Load Cover Image Algorithm.
6. Call Hide the length of the secrete message Algorithm.
7. Call Hide the secrete message.
8. Save the new content of image file with the secrete message in new image file.
9. Close the image file
10. End.

5-1 Load Cover Image Algorithm

1. Start.
2. Display the image in image viewer.
3. Get BMP image file for reading.
4. Read the header of image file.
5. Read image data from image file.
6. Copy the bytes of image data into temporary memory.
7. End.

5-2 Hide the length of the secrete message Algorithm

1. Start.
2. Convert length from integer form to binary form.
3. Split the last three pixels of image to its color (RGB).
4. End.

5-3 Hide the secrete message

1. Start.

2. Select the positions (pixels) in which the bits of secrete message will be hide randomly.
3. Select the bits of each of the selected pixels in these bits we will hide the message bits and the bit which we will chose to hide the secrete message will be selected randomly of last three bits for each byte of the pixels which in turn selected randomly.
4. End.

6- The extracting process of the proposed system

In this section we are going to describe the extraction process of hiding data in the covered image, first of all, we read the covered image then we must specify the length of the hided message by the sender in the last three pixels of image.

We have to extract the least 2 significant bits of each color in the last three pixels of the coverd image and combine these bits together to produce a binary number which can be transformed to numeric form, this number represents the length of the message which hided in the coverd image. Then we used this length to know the last position that the sender used, to hide the message. We used this length in extraction of the secret message from the covered image.

In The extraction process of hided message we must first generate the position of the bits of the secrete message that we hide, so we use the same algorithm used in the hiding process to generate the same random position and give as input for the random function which generate these random positions :

1. Start point (first pixel in image).
2. End point (size of image-3)
3. The key (obtained from password).

According to this function the pixel will be chosen randomly.

So the extraction of the message from the image's pixels done by selecting the right pixels and the right bits of the secrete message which were hided randomly. select one of the last 3 significant bits of each color in each pixel, also select the right bit randomly from these 3bits and will used to compose the secret message by gathering each 8 bits to build a byte then we convert this byte into one character belong to the secret message and so on.

6-1 The extracting Algorithm

1. Start.
2. Call Load Cover Image Algorithm.
3. Call Find length Algorithm.
4. Call Extract the secrete message Algorithm.
5. Display the secrete message.
6. End.

6-1-1 Load Cover Image Algorithm

1. Start.
2. Display the image in image viewer.
3. Get BMP image file for reading.
4. Read the header of image file.
5. Read image data from image file.
6. Copy the bytes of image data into temporary memory.
7. End.

6-1-2 Find Length Algorithm

1. Start.
2. Select the last three pixels of the image pixels.
3. Split each of the pixels to its colors (RGB).
4. Take the last two bits from each color.
5. Form binary number from the whole bits which be extracted from the last three pixels.

6. Convert the extracted binary number from binary form to integer form which represent the length of the secrete massage.
7. End.

6-1-3 Extraction of the secrete message Algorithm

1. Start.
2. Select the positions (pixels) in which the bits of secrete message which were hided randomly.
3. Select the bits of each of the random selected pixel in which the bits of the secrete message hided in and this bits will be chosen randomly in the same way we used to select this bit to hide the bits of the secrete message in hiding algorithm.
4. Convert each 8 bits of the bits which we are extracted from the bits of image's pixels to one character so as result we will obtain on the secrete message which was hided in the hide algorithm.
5. End.

7- Characteristics of information hiding in our proposed system

7-1 Capacity:

Since the random function can generate large number in size (dimensions) as needed, so if we use an image of size 256*256 which can carry about 21842 characters message with a good perceptibility, as shown in the result table.

7-2 Perceptibility:

As the ability of the third person (human visual system) to detect the changes in the image information increased when change localized in the adjacent regions (positions) of the image. So the proposed method distribute the positions(pixels) over the image area(un adjacent positions) to minimize the perceptibility for the third person especially when some pixels not changed or change slightly after hiding process. To measure the perceptibility level of the method system used the PSNR. Measure of the result showed that the method gives good perceptibility with most experiment.

7-3 Robustness:

Robustness for the information hiding method is the measure of the method resistance to the transformation processes like (translation, scaling, and rotation) in the proposed system. If the transformation process known and reversible without any losing of the information (especially in the pixels carry the message bits), then the method can retrieve the secret message either by apply some transformation of the random function to find the new position of the cover pixels or by apply reverse transformation on the received image before extracting the hidden message and since most of transformation processing are reversible without losing some exceptions. The method will gives a good robustness when the transformation process known also if the process are unknown and the system of escrow type (not blind) the system can compare the received image with the original one to detect the transformation that applied on the cover image and reverse it to extract the hiding message.

7-4 PSNR and experimental Results:

The Results of our proposed system are shown in the table 1 and figure 3 compared with traditional methods.

Table 1 PSNR Results of our proposed system.

Picture	PSNR	TRADITIONAL PSNR
MONA LIZA	39.843	33.072
EVIL TOWER	39.175	33.377
ARAB TOWER	37.781	31.285
BIG BEN WATCH	39.157	33.109

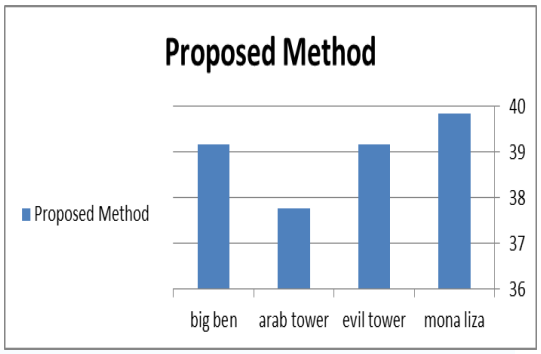

Fig. 3 PSNR of Public images were used in our proposed method.

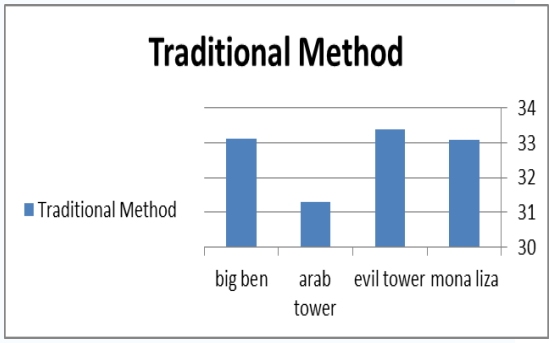

Fig. 4 PSNR of Public images were used in Traditional method.

a) Big Ben watch (Original)

b) Big Ben watch (Stego image)

Fig. 5 The original and stego image of Big Ben watch.

As you see the results of PSNR when we hide the same text in these different images, we get very good PSNR results, between 20 to 40 that mean the quality of resulted covered image is very good and we keep PSNR in range of accepted values.

If we compare these results of our proposed system with a traditional bit insertion methods we get a different gap of PSNR values as shown in figure 3; the cause of this different results is we don't need to change bits in adjacent pixels to represent hiding data in the image and if we need to change we change only one bit like traditional so the result of proposed system already less than or equal to traditional bit insertion methods but not greater. In proposed method we get quality of covered image closed to original image, distribute of hiding data in the image in a way even if the attacker detect there is an information or different in the covered image he

couldn't Know the representation of hiding data. If we compare results of our proposed method with the traditional bit insertion steganography we get a good PSNR and closed to 20 that mean the quality of resulted image (covered image) is much closer to original as shown in figure 3.

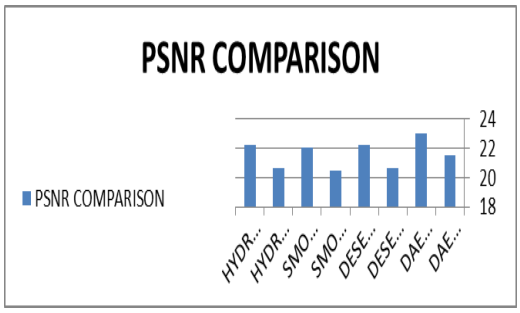

Fig. 3 PSNR Comparison

8- Conclusion:

1. The capacity of breaking hiding process is increasing, even if the pixels that carrying the secret message known when the sequence of visiting these pixels is unknown.
2. Even blind methods are less overhead than escrow one, but escrow method is necessary to overcome some weakness and increase robustness of information hiding method.

9- Future Work:

The proposed method is applied on images, but there is no restriction to apply it on other type of media like video movie, audio, and others with take care of media special criteria.

References:

[1] Alkhraisat Habes Information Hiding in BMP image Implementation, Analysis and Evaluation
Saint Petersburg Institute for Informatics and Automation, Russian Academy of Sciences,
Saint Petersburg, Russia Received February 26, 2006

[2] Fabien A. P. Petitcolas, Ross J. Anderson and Markus G. Kuhn, "Information Hiding – A Survey." Proceedings of the IEEE , vol. 87, no. 7, pp. 10621078,
July 1999.

[3]F. Rosenblatt. The perceptron a probabilistic model for information storage and organization. Brain Psych. Revue, 62:386.408, 1958.

[4] Eugene T. Lin and Edward J. Delp - A Review of Data Hiding in Digital Images.
Video and Image Processing Laboratory (VIPER), School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, 2008

[5] David salmoon. The data compression complete refrence 2008, British Library Cataloguing in Publication Data A catalogue record for this book is available from the British Library Library of Congress Control Number: 2006931789, ISBN-10: 1-84628-602-6, Springer-Verlag London Limited 2007.

[6] Peter Wayner, Dissapearing Cryptography – Information Hiding: Steganography & Watermarking – Second Edition. San Fransisco, California, U.S.A.: Elsevier Science, 2002, ISBN 1558607692.

بناء نظام اخفاء بيانات معتمد بأستخدام المفتاح العشوائي في الحيز المكاني للصورة

عامر جعفر صادق* ***زينب سلام قاسم****

*قسم علوم الحاسبات- كلية العلوم للبنات- جامعة بغداد

الخلاصة:

ان اهمية اخفاء المعلومات اصبحت متزايده مع مرور الزمن، وان كل تطبيقات اخفاء البيانات مثل العاب الحاسوب من ناحية اخفاء واظهار البيانات، او مثل رجل الشرطة واللص حيث ان اللص دائما يختفي عن رجل الشرطة بمختلف الطرق ليبقى خارج السجن. المرسل دائماً يخفي المعلومات بطريقة جديدة حتى لايتم فهمها من قبل المهاجمين والمستلم المخول هو فقط من يمكنه ان يفتح الرسالة المخفيه.

هذه الدراسة تستكشف طريقتنا العشوائية المقترحة بالتفصيل، كيفية أختيار المواقع في الصورة عشوائياً. وكيفية اختيار البت العشوائي لأخفاء البيانات في الموقع المختار، وكيفية اختلاف طريقتنا عن الطرق الاخرى في اخفاء البيانات وتطبيق مقاييس اخفاء البيانات على المشروع المقترح.