Journal of Engineering and Applied Sciences 14 (16): 5854-5863, 2019
ISSN: 1816-949X

Enhancing LSB Algorithm against Brute-Force Attack using ASCII Mapping Technique

[1]Sinan A. Naji, [2]Hussein L. Hussein, [3]A. Mustafa Ihsan and [4]Jasim H. Lafta
[1]College of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
[2]Ibn Al-Haitham College of Education, University of Baghdad, Baghdad, Iraq
[3]Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia
[4]Faculty of Informatics, Debrecen University, Debrecen, Hungary
E-mail: dr.sinannaji@uoitc.edu.iq

Abstract: The primary objective of steganographic techniques is to conceal a secret message in an innocent-looking cover media. One of the well-known techniques is Least Significant Bit (LSB) algorithm which is used in many information security applications. This current study presents an efficient solution to improve the security level of the standard LSB. The recommended solution combines two techniques; the standard LSB algorithm and the ASCII Mapping Technique (AMT). In the first step, AMT is used to encode the secret message using an image-key instead of classical stego-key. In the second step, the recursive technique along with circle equation in parametric polar form are used for embedding the message bits into pixels that are selected randomly. Furthermore, noise data are embedded into stego-image in order to render the steganalysis task more challenging. The steganalysis not only has to guess which pixels have been used to hide data but also must isolate real hidden data from noise. Compared to standard LSB algorithm, the proposed technique achieves a significant security improvement, higher capacity for hiding data with high image quality.

Key words: Steganography, data embedding, random pixel selection, LSB, ASCII mapping technique, steganalysis

INTRODUCTION

Steganography is a technique for concealing data or a secret message within an innocent-looking cover media (Senarathne and De Zoysa, 2014). It provides secret communication between the sender and receiver (Petitcolas *et al*., 1999). Steganography is derived from Greek roots and literally means 'covered writing'. It has been used for centuries. It includes a wide range of undisclosed communication methods starting from wax tablets, messenger's body, secret inks, digital signatures, etc. (Cheddad *et al*., 2010; Hussain and Hussain, 2013). Now a days, steganography plays an important role in many information security applications such as confidential communication, secret data storing, copyright protection and user authentication. In practice, steganography is very closely related to cryptography. While the primary objective of cryptography is conversion of the message into unreadable format (i.e., cipher-text), the primary objective of steganography is to embed the message into a carrier media in a manner which renders it undetectable by any unauthorized third party. Nonetheless, cryptography is employed together with steganography by the application of message encryption prior to carrying out steganography (Khosravi *et al*., 2012; Tuama *et al*., 2017). Now a days, many different digital file formats like digital images, sound files, text files and video clips are used by different steganography techniques (Tiwari *et al*., 2014). Generally, digital images are the most popular media (Singh, 2014). Numerous image-based steganography methods are presented in the literature. These methods can be classified into four main categories (Tuama *et al*., 2017).

Spatial domain techniques: These techniques are based on directly altering the intensities of image pixels to encode the message bits (Li *et al*., 2011). These methods include Least Significant Bit (LSB), edges based data embedding (Luo *et al*., 2010; Deshmukh and Pattewar,

Corresponding Author: Sinan A. Naji, College of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq,
E-mail: dr.sinannaji@uoitc.edu.iq

2014) pixel value differencing (Shen and Huang, 2015; Luo *et al*., 2011), random pixel embedding (Tiwari *et al*., 2014) etc.

Frequency domain techniques: The image is initially transformed into frequency domain followed by embedding of the message in the transform coefficients. Several transforms are available for this purpose. The most widely-used are: Discrete Wavelet Transform (DWT) (Chen and Lin, 2006; Nag *et al*., 2010; AL-Nabhani *et al*., 2015), Discrete Cosine Transform (DCT) (Kaur *et al*., 2011; Raja *et al*., 2005; Rani and Khan, 2014) and Discrete Fourier Transform (DFT) (Wang and Moulin, 2006). Although, transform domain techniques are noise tolerant, they are computationally expensive (Singh, 2014; Jindal and Kaur, 2016).

Masking and filtering techniques: These techniques are based on using masks and filters to encode the message bits (Radhakrishnan *et al*., 2005; Johnson and Jajodia, 1998).

Distortion techniques: These techniques are based on a sequence of changes corresponding to the secret message the sender wants to hide (Kim *et al*., 2007; Filler *et al*., 2011; Holub and Fridrich, 2012). The recipient determines the differences in comparison with the original cover image and reconstructs the sequence of modifications to correspond to the secret message.

When there is a need to build a steganographic system, three main questions are formulated: What technique to choose? How many message bits should be embedded in each pixel? Where exactly the actual message should be embedded? The challenges related to these methods are affected by the following factors (Alsarayreh *et al*., 2017; Kukapalli *et al*., 2014).

Undetectably: This suggests that the resulting stego- image as well as the cover-media need to be seen as nearly a like (Luo *et al*., 2010). In other words, the stego-image does not contain any detectable defects caused by message embedding (Fridrich *et al*., 2001; Luo *et al*., 2010; Nag *et al*., 2010; Tuama *et al*., 2017).

Robustness against statistical attacks: This implies that the initial cover-image as well as the resulting stego-image should have the same statistical characteristics against statistical attacks (Fridrich *et al*., 2001; Tuama *et al*., 2017; Goyal *et al*., 2015).

Capacity/Payload: The greatest amount of data that can be hidden in the cover image (Li *et al*., 2011; Kumar and Yadav, 2014).

Computational complexity: The amount of resources required for running the proposed method (Hussain and Hussain, 2013; Chen and Lin, 2006).

In computing, digital images are arrays of numbers created from the physical scene picture. These numbers represent the intensity values of the image pixels (Efford, 2000). For gray-scale image, the most common system used is 8 bits per pixel. Thus, for each pixel there are 256 intensity values. For true-color image such as RGB images, the most common system used is 24 bits per pixel. This implies that each pixel offers 256×256×256 = 16.7 million colors (Ma and Leijon, 2010). A slight alteriation in pixel intensity will produce indistinguishable change in pixel appearance and it remains difficult for the Human Vision System (HVS) to detect the slight changes.

From our point of view, Least Significant Bit (LSB) algorithm is among the most significant algorithms. It is based on a substitution procedure to replace the least significant bit (i.e., 8^{th} bit) of original pixel intensity with the secret message-bit directly (Senarathne and De Zoysa, 2014). This means that the secret message is subjected to conversion into binary stream and sequentially overwrites the least significant bit of the image pixels. The main characteristics of LSB algorithm can be summarized as follows: simple to understand, ease of implementation and less distortion. However, the main drawback of the LSB algorithm is its weakness in message protection. A brute-force attack by the intruders can identify the secret message (Tuama *et al*., 2017; Kukapalli *et al*., 2014).

This research aims to improve the security level of the standard LSB algorithm. The proposed solution combines two techniques, the standard LSB algorithm and the ASCII Mapping Technique (AMT). The implementation of different methodologies in one integrated system where one method can compensate for the weaknesses of another, enhances the general performance of the system. This study shows the efficiency of the suggested solution through experiments and comparisons with the existing standard LSB.

Literature review: Several solutions are proposed to improve the security level of the LSB algorithm. Jain and Ahirwal (2010) proposed an adaptive LSB replacement scheme along with private-key. The technique embeds adaptive number of bits of the secret message using XOR-based digital signature key. The authors claimed that the proposed method could hide (4.20 and 4.15) bits in each pixel of gray and colored images, respectively. Lee *et al*. (2009) proposed an advanced LSB embedding scheme using Pairs of Values (PoVs) concept.

The 2 bytes whose values differ only in the LSB are called (PoVs). For example, $36(00100100)_2$ and $37(00100101)_2$ are PoVs. When the counts of 1s and 0s are equal and scattered arbitrarily in the secret message, the occurrence of two values in each PoV will be the same after embedding the message. The PoVs artifact resulted from standard LSB algorithm reveals the presence of an unseen message. The proposed scheme is based on breaking the regular pattern of PoVs in the histogram domain to make the task of steganalysis more difficult. The researchers claimed that the attacks could not detect the secret messages embedded with the proposed system.

Swain and Lenka (2015) proposed an array-based LSB algorithm. The method uses 4 LSB arrays: LSB (0-3). The first array (i.e., LSB0) is constructed by combining the least significant bit from all pixels (i.e., 8^{th} bit). The second array (i.e., LSB1) is constructed by combining two least significant bits of all pixels (i.e., 7 and 8^{th} bits). The third array (i.e., LSB2) is constructed by combining together three least significant bits (i.e., 6-8^{th} bits) of all pixels. The forth array (i.e., LSB3) is constructed by combining four least significant bits (i.e., 5-8^{th} bits) of all the pixels. The system selects one of these arrays to embed the secret message. The selection is based on message size. When the message is short, the LSB0 array can be used whereas LSB3 array can be used with long messages for better matching. The secret message is fragmented into words and then the best match for each word in the binary LSB array is found. The matches are encrypted using RSA algorithm and then embedded in two least significant bits to produce the stego-image.

Wang *et al.* (2010) presented a steganography scheme based on Genetic Algorithm. The system embeds the secret message using standard LSB algorithm. Then pixel values of the stego-image are altered by Genetic algorithm to shield the statistical characteristics against statistical attacks.

Muhammad *et al.* (2016) utilized HSV color space to conceal data. First, the input image (i.e., RGB) is converted to HSV color space. Then, an adaptive LSB substitution method is applied for embedding the encrypted message within the V-channel of HSV color space. The system uses secret key-directed block LSB algorithm to enhance the robustness of the LSB steganography.

Muhammad *et al.* (2017, 2018) proposed using the I-channel of the input image in HSI color space for hiding data. The system uses MS-directed LSB substitution method along with Three-Level Encryption Algorithm (TLEA) prior to embedding. Akhtar *et al.* (2014) proposed a new bit-inversion method to lessen the embedding effect before applying the LSB algorithm. The idea of this technique is as follows: the LSBs of certain pixels are inverted, if they come with a specific pattern of some bits of the pixels. The researcher claimed that the number of pixels that would be modified would be highly reduced in comparison with the standard LSB algorithm. The proposed technique improves the security of the LSB algorithm and also maintain low effects on image quality. Other studies to enhance the security of LSB algorithm can be found by Raja *et al.* (2005), Khalaf and Sulaiman (2011), Viswanatham and Manikonda (2010), Hussain and Hussain (2013), Li *et al.* (2011), Singh (2014) and Cheddad *et al.* (2010).

MATERIALS AND METHODS

The proposed technique: The security level of the proposed system should not depend on the secrecy of the steganographic algorithm itself (Jain and Ahirwal, 2010). A number of LSB-based steganographic systems (Alsarayreh *et al.*, 2017; Jain and Ahirwal, 2010) uses stego-key to improve the security level of the system. On the other hand, brute-force attack is commonly used by attackers. One approach of the attacker is to start by going over every possible combination of letters and numbers. In this research, we propose an image-key instead of classical key in order to resist brute force attacks. The sender and receiver agree to use a secret image-key prior to communication. Using a digital image as a key instead of classical key makes the task of the attacker more difficult, since, there is no common image-database that can be searched to identify which image is the key. Now a days, we are generating a huge amount of images, more than could ever be indexed in one public database. A very significant contributing factor has definitely been the internet which provides the medium through which millions of images are involved daily. With this solution, the attacker has to start searching by going over every possible image. Figure 1 shows, the block diagram of the suggested system. The three files are required for embedding a message into an image. The first file is the message (the data to be hidden). The second file is an innocent-looking cover image that will be used as carrier. The third file is the image-key employed by AMT algorithm to encode the secret message before performing the embedding. For further security, we used a random selection of pixels that hide the bit's stream of the secret message using recursion technique along with circle curve equation in parametric polar form. As shown in Fig. 1, the receiver must have the image-key and randomization parameters in order to recover the secret message from the stego-image.

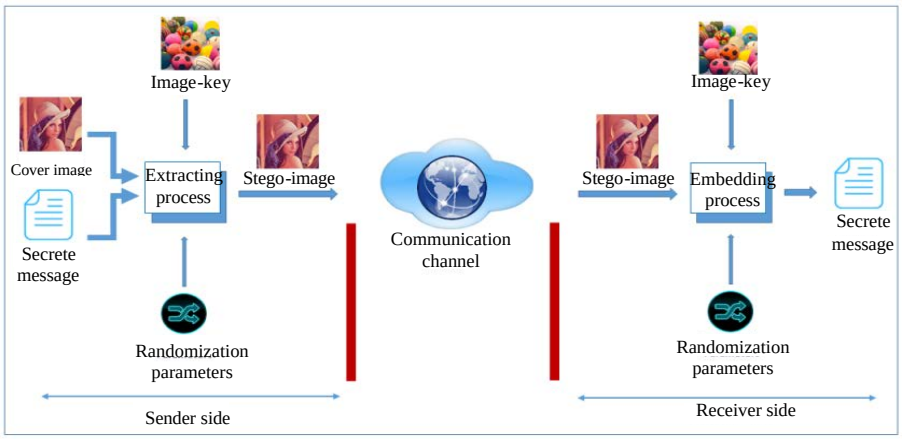

Fig. 1: The proposed steganography system

Increasing the capacity: Based on many extensive experiments (Hussain and Hussain, 2013; Li *et al*., 2011; Singh, 2014; Cheddad *et al*., 2010) it was found that even when altering two least significant bits (i.e., the 7th and 8th bits) of a pixel's intensity, the pixel appearance will retain the same look. For instance, if the intensity values of the RGB color channels are (187, 100, 165)$_{10}$ = (10111011, 01100100, 10100101)$_2$ then, altering the 2-LSBs of the Red channel to (184,100,165)$_{10}$ = (10111000, 01100100, 10100101)$_2$ will produce indistinguishable change in pixel appearance and it remains difficult for the human eye to detect the slight change in pixel color due to the fact that the amplitude of the change is still, so, small. With the goal to raise the capacity of the standard LSB algorithm we propose to use 2-bit tokens instead of 1-bit for message encoding, message embedding and message extracting. With this variation to standard LSB, the system can store up to 6 bits in each pixel (i.e., 2 bits in each color channel of RGB color space). In other words, the maximum data that can be embedded in the cover image is doubled. In general, the capacity of the steganographic method can be further raised by raising the number of the substituted bits (e.g., 3-LSBs) but there is always a tradeoff between the number of bits used for hiding data and the quality of the resulting stego-image.

AMT algorithm: In this study, we propose to use an image-key as an alternative to the traditional stego-keys in order to resist brute force attacks. The sender and receiver agree to use a digital image-key prior to communication. By assuming that the digital image can be at different scales and different resolutions, the size of image-key should be chosen, so that, the number of pixels should be large enough to encode the message. When the user selects the appropriate size, the image-key is converted into 2D-matrix of 2-bit tokens named K(N, M) where, N and M are the dimensions of the matrix K. This matrix is used as a lookup-table by AMT algorithm as follows (Bhattacharyya *et al*., 2013), the system reads the input message as a sequence of ASCII codes stream and converts it into tokens of 2-bit stream. The system takes the first token and tries to find a match with an entry K(x, y) which is randomly selected. When the token matches the entry at location K(x, y) that entry will be locked and its coordinates (x, y) are saved in a separate table. All tokens are processed in the same way until, the end of the token's stream. The idea of entry locking is that when a similar token appears again in the input stream, it will assign different coordinates. This makes the encoded tokens more secure against attacks. Table 1 shows, an example, of message encoding using AMT algorithm in conjunction with the image key. As shown in Table 1, the secret message (i.e., ASCII code format) is encoded as a sequence of random numbers. The generated table will be passed to the subsequent stage (i.e., message embedding).

Table 1: Message encoding using AMT algorithm

ASCII code	Coordinates (Bits)							
	1st and 2nd		3rd and 4th		5th and 6th		7th and 8th	
	X	Y	X	Y	X	Y	X	Y
01001101	27	14	408	361	32	500	212	12
01011001	449	212	155	405	439	97	399	462
00100000	86	364	163	491	520	44	479	4
01010011	68	22	2	15	211	23	153	68
01000101	29	4	325	91	411	10	194	9
01000011	283	230	371	492	115	8	63	2
01010010	69	24	391	51	387	11	1	326
01000101	237	25	40	495	75	359	54	243
01010100	361	207	479	280	99	33	292	321
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
01000110	25	321	481	161	116	72	63	280
01010010	164	91	476	509	387	11	1	395
01001001	16	405	92	137	75	359	54	61
01000100	331	125	30	45	175	93	455	413
01000001	309	207	502	280	99	33	190	9

Circle curve equation: Before implementing the embedding routine, the researchers have to decide where exactly the actual message should be embedded. Many approaches have been suggested in the literature to decide where to embed the secret bits: sequential pixels, zigzag path, edge points, etc. Although, these methods have been widely used, they can be effortlessly hacked by attackers as these techniques are quite straightforward and work in a systematic way (Muhammad *et al*., 2016). Accordingly, it is no big challenge for the hackers to identify the presence of a secret message within the cover media and then succeed in retrieving the message. In order to improve the security level of the standard LSB algorithm we propose using a recursion technique along with circle equation to generate random sequence of points. Recursion is an interesting technique in computer programming that is based on solving a problem by the accumulation of solving increasingly smaller instances of the same problem. This technique along with circle equation in parametric polar form can generate different sequences of points based on previously defined randomization parameters.

Expressing circle equation in parametric polar form produces the following pair of Eq. 1 and 2 (Donald and Baker, 1997):

$$x = x_c + r\cos\theta \quad (1)$$

$$y = y_c + r\sin\theta \quad (2)$$

Where:

x_c, y_c = The coordinates of the circle's center
r = The radius

Practically, Eq. 1 and 2 are used to generate adjacent sequence of points (x, y) on the circle curve. To generate non-adjacent points, a positive integer parameter s is used to define the step between points. For example, when s equals to 1, the generated points will be next to each other. On the other hand when s equals n, the generated points will be spaced by n points. So, the higher the value of s, the more scattered the points will be.

To generate the points (i.e., a traversing path), it is first required to choose a point called base center (x_0, y_0), radius, r and step s. The points are generated using recursion technique as follows, given a base circle center (x_0, y_0), radius r and step s, a loop repeatedly increases the radius r to create multiple curves at different scales using Eq. 1 and 2. The generated points represented by x and y coordinates are pushed into a stack. Then each point which is represented by x and y coordinates is popped from the stack and treated as a new base center (x_0, y_0) to generate another sequence of curves and consequently leads to generate a new sequence of points and so on. In other words, the system generates new points from previously generated points using recursion technique. Experimentally, this technique shows that the generated points are well scattered over the entire image. The number and sequence of the points on the curve depend mainly on the base randomization parameters (x_0, y_0), r and s. Using different parameters produces a set of varying sequences of points generation. So, the attacker not only needs to locate the pixels that hide data but also must determine the correct sequence of these pixels to reconstruct the secret message. Using this technique, the randomization parameters are regarded as a key and the receiver must use it to extract the embedded message as shown in Fig. 1. Accordingly, this technique provides more security where the sequence of the message bits is only known to both sender and receiver.

Message embedding: The main objective of this stage is to embed the encoded message into an innocent-looking cover image. Message embedding stage is done as follows: a traversing path is first generated based on shared randomization parameters. The points of the traversing path, each represented by x and y coordinates are used to locate the pixels on the innocent-cover image. As mentioned before, we propose increasing the capacity of standard LSB by embedding 2 bits instead of 1 bit. Accordingly the bit's stream of the message is processed sequentially as tokens of 2 bits and will be embedded into the 2 LSBs of the selected pixels one after the other. The step-by-step algorithm for the embedding process is listed as follows (Algorithm 1):

Algorithm 1; Proposed algorithm (message hiding-sender part):

```
Input: Message, Cover-image, Image-key , X0, Y0, R, S
Output: Stego-image
Begin
   Stego-image = Cover_image
   // Generate list of points using recursion technique
   List_points = Generate_ points (X0, Y0, R, S)
   // Convert image-key into 2D-Matrix lookup table
   K = Matrix (Image-key)
   // Convert input message into 2-bit tokens stream
   Tokens = Convert_to _tokens (Message)
   // Message encodin
   Encoded_table = Message_Encoding (Tokens, K)
   Tokens2 = Convert_to _tokens (Encoded_table)
   // Message embedding
   i = 1
  While not_empty (Tokens2)
   Begin
      Value = Tokens2[ i ]
      [x, y] = Get_Next_Point ( List_points )

      // Set the 2-LSBs to zero
      Stego-image(x, y) = Stego-image (x, y) && 252

      // Embed 2 bits into 2 LSBs
     Stego-image(x, y) = Stego-image (x, y)  || Value
     i = i + 1
  end
  output (Stego-image)
end
```

Adding noise: In this research, we propose to embed noise data into the stego-image in unused pixels to make the steganalysis task more difficult. The steganalysis not only needs to identify which pixels are hiding data but also must separate the ones that hide noise data from the ones that hide real data as well as its arrangement to reconstruct the message successfully.

The more noise is added to the stego-image, the more confusion is added to the steganalysis. Noise embedding process is done as follows; suppose that K is the maximum amount of noise, the sender randomly selects K points for holding the noise data. If the (x, y) coordinates of a point are already overlapped with traversing path points that point will be discarded or else, it will be utilized to hold a random 2-bit value. However, the receiver does not need to know which pixels are used for holding the noise data because the retrieving process is entirely controlled by the randomization parameters and the corresponding traversing path.

RESULTS AND DISCUSSION

Experimental results: The experimental results provided in this section define the performance of the proposed technique. Our system was carried out on Intel (R) Core i7 processor with 8-GB RAM on Windows platform. We used MATLAB R2017b package to for the implementation of both the standard LSB and the proposed algorithm and also image statistical measures for quantitative evaluation. The experiments were conducted using images from two image datasets:

Set 1: The "USC-SIPI image database" of the University of Southern California. It is maintained primarily to support researchers in image processing and machine vision. The database is divided into categories based on the contents of the images. We have used the "Miscellaneous category" that contains famous images such as Lena, mandrill, peppers, etc.

Set 2: Our own image dataset comprises a collection from the internet. This dataset consists of 260 images collected from different sources and includes various image types. For qualitative evaluation, Fig. 2 shows, examples of applying the proposed technique. Figure 2a is the original cover image; Fig. 2b shows the resulting stego-image. As shown in this figure, the original cover-image and the resulting stego-image are visually almost identical. In other words, the stego-image does not contain any detectable defects caused by message embedding. From that we infer that the proposed technique has a subordinate effect on image quality.

Fig. 2: The results of the proposed technique: a) The cover-image and b) The resulting stego-image

For quantitative evaluation, the Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) are commonly used measures to evaluate the performance of the proposed algorithm. The MSE is a representation of the cumulative squared error between the cover and stego-image. The lower the value of MSE, the lower the error. The PSNR is a measure of the peak error. A higher PSNR means better quality image (Tuama *et al*., 2017;

Table 2: The PSNR test results

Testing image	Image size	Standard LSB	Proposed algorithm
Lena	512×512	81.9715	81.9686
Mandrill	512×512	82.5877	82.5849
Peppers	512×512	82.2272	82.2302
House	512×512	81.3344	81.4764
Albert Einstein	569×523	82.8094	82.7248
Nature	624×534	82.5557	82.5542

Table 3: The MSE test results

Testing image	Image size	Standard LSB	Proposed algorithm
Lena	512×512	0.000411	0.000413
Mandrill	512×512	0.000379	0.000359
Peppers	512×512	0.000388	0.000389
House	512×512	0.000451	0.000463
Albert Einstein	569×523	0.000352	0.000341
Nature	624×534	0.000353	0.000361

Table 4: The SSIM test results

Testing image	Image size	Standard LSB	Proposed algorithm
Lena	512×512	0.999999	0.999999
Mandrill	512×512	1.000000	1.000000
Peppers	512×512	0.999999	1.000000
House	512×512	0.999999	0.999999
Albert Einstein	569×523	0.999999	0.999999
Nature	624×534	0.999999	0.999998

Rani and Khan, 2014). To calculate the PSNR we begin by computing the MSE with the following Eq. 3 (Singh, 2014; Tiwari *et al*., 2017):

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}[C(i,j)-S(i,j)]^2 \quad (3)$$

Where:

m and n = The image dimensions

C(i, j) and S(i, j) = The cover and stego-image pixels, respectively

The PSNR is computed with the following Eq. 4 (Singh, 2014; Tiwari *et al*., 2014):

$$PSNR = 10\log_{10}\frac{R^2}{MSE} \quad (4)$$

where, R is the maximum possible value in the input image's pixels. In this research, we use standard LSB algorithm as our benchmark for the comparison. Table 2 and 3 present the average PSNR and MSE of the proposed system compared to standard LSB using test images presented in Fig. 2. Inclusion of a recent metric such as Structural Similarity Index Metric (SSIM) can provide a fair comparison along with MSE and PSNR. Table 4 shows the SSIM test results of the proposed system compared to standard LSB. As shown in these tables, the PSNR, MSE and SSIM metrics for the proposed algorithm are very close to that of standard LSB.

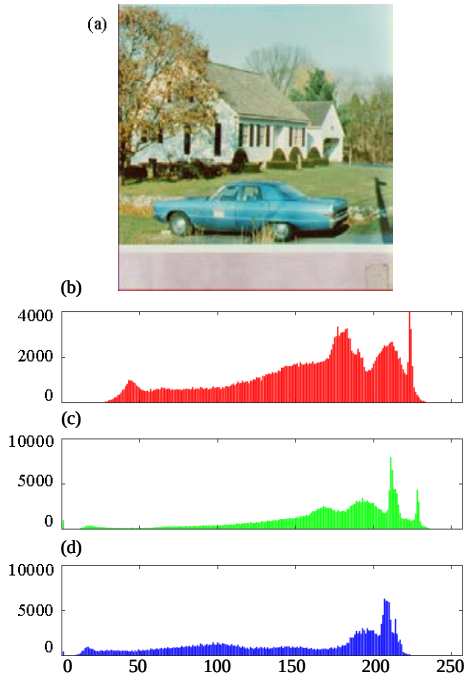

Fig. 3: a-d)Cover image and its histogram

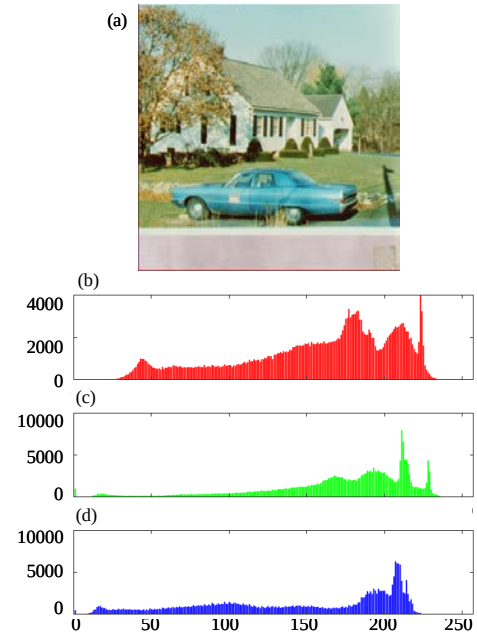

Fig. 4: a-d)Stego image and its histogram

Another performance measure that can be employed to show the quality of the embedding process is the image histogram. Figure 3 shows, a cover image and the histograms of its three color channels (Red, Green and Blue). Figure 4 shows, the corresponding stego-image and the histograms of its three color channels after embedding a message of 33 characters.

As shown in Fig. 3 and 4, the resulting histograms are very similar to those of the original histograms. From all that we conclude that the original cover-image as well as the resulting stego-image have similar statistical characteristics which cannot be differentiated when applying statistical analysis.

Compared to the work presented by Jain and Ahirwal (2010) the researchers claimed that their system could hide an average of 4.15 bits in each pixel of colored image based on adaptive number of least significant bits. In this research, the proposed technique can hide 6.0 bits in each pixel with a subordinate effect of image quality.

Generally, the researchers perform the experiments using their own test images. Furthermore, they used different secret messages for testing and evaluation. Therefore, the MSE, PSNR and SSIM measures are highly affected by the resolution of the image and the size of the secret message. Accordingly, the evaluation is subject to the system designer's judgment.

CONCLUSION

This research aims to enhance the security level of LSB algorithm. This enhancement is designed to resist brute force attacks. The main idea is based on combining two techniques to overcome the weakness of the security issues involved in the standard LSB algorithm.

Two images are used as input: the first one is used as image-key for encoding the message while the second is used as a carrier. With this solution, even though the stego-image could be hacked by attackers they still need the image-key for successful message reconstruction. Consequently, the attacker has to start searching by going over every possible image. However, since, the proposed solution uses two images, the processing time required for message embedding is greater than before.

The random selection of pixels that hold the secret bits prevents the attacker from predicting the sequence of pixels used to hide the message. Moreover, we propose embedding noise data into stego-image to make the steganalysis task more difficult. The steganalysis not only has to guess which pixels have been used to hide data but also must isolate pixels that hide noise data from the ones that hide real data.

In this study we have also used 2-LSBs for concealing data. This variation shows two advantages: first, the capacity/payload of the proposed system is increased to double compared to standard LSB. Second, the task of steganalysis becomes more confusion, since, most LSB-based applications apply one-bit substitution. The results show that the proposed solution provides an additional level of security that is relatively higher than the standard LSB algorithm with a subordinate effect of image quality.

Ethics: This research is original and unpublished material. The corresponding author confirms that there are no ethical issues involved.

REFERENCES

AL-Nabhani, Y., H.A. Jalab, A. Wahid and R.M. Noor, 2015. Robust watermarking algorithm for digital images using discrete wavelet and probabilistic neural network. J. King Saud Univ. Comput. Inf. Sci., 27: 393-401.

Akhtar, N., S. Khan and P. Johri, 2014. An improved inverted LSB image steganography. Proceedings of the 2014 IEEE International Conference on ISSUES and Challenges in Intelligent Computing techniques (ICICT'14), February 7-8, 2014, IEEE, Ghaziabad, India, ISBN:978-1-4799-2899-6, pp: 749-755.

Alsarayreh, M.A., M.A. Alia and K.A. Maria, 2017. A novel image steganographic system based on exact matching algorithm and key-dependent data technique. J. Theor. Appl. Inf. Technol., 95: 1212-1224.

Bhattacharyya, S., P. Indu and G. Sanyal, 2013. Hiding data in text using Ascii Mapping Technology (AMT). Intl. J. Comput. Appl., 70: 29-37.

Cheddad, A., J. Condell, K. Curran and P. McKevitt, 2010. Digital image steganography: Survey and analysis of current methods. Signal Process., 90: 727-752.

Chen, P.Y. and H.J. Lin, 2006. A DWT based approach for image steganography. Int. J. Applied Sci. Eng., 4: 275-290.

Deshmukh, P.U. and T.M. Pattewar, 2014. A novel approach for edge adaptive steganography on LSB insertion technique. Proceedings of the International Conference on Information Communication and Embedded Systems (ICICES2014), February 27-28, 2014, IEEE, Chennai, India, ISBN:978-1-4799-3834-6, pp: 1-5.

Donald, H. and M.P. Baker, 1997. Computer Graphics C Version. Prentice Hall Publishing Company, Upper Saddle River, NJ, USA.

Efford, N., 2000. Digital Image Processing: A Practical Introduction using Java. Addison-Wesley Company, Boston, Massachusetts, USA., ISBN:9780201596236, Pages: 340.

Filler, T., J. Judas and J. Fridrich, 2011. Minimizing additive distortion in steganography using syndrome-trellis codes. IEEE Trans. Inform. Forensics Secur., 6: 920-935.

Fridrich, J., M. Goljan and R. Du, 2001. Detecting LSB steganography in color and gray-scale images. IEEE Multimedia, 8: 22-28.

Goyal, M., Y. Lather and V. Lather, 2015. Analytical relation and comparison of PSNR and SSIM on babbon image and human eye perception using matlab. Intl. J. Adv. Res. Eng. Appl. Sci., 4: 108-119.

Holub, V. and J. Fridrich, 2012. Designing steganographic distortion using directional filters. Proceedings of the 2012 IEEE International Workshop on Information Forensics and Security (WIFS), December 2-5, 2012, IEEE, Tenerife, Spain, ISBN:978-1-4673-2285-0, pp: 234-239.

Hussain, M. and M. Hussain, 2013. A survey of image steganography techniques. Int. J. Adv. Sci. Technol., 54: 113-124.

Jain, Y.K. and R.R. Ahirwal, 2010. A novel image steganography method with adaptive number of least significant bits modification based on private stego-keys. Intl. J. Comput. Sci. Secur., 4: 40-49.

Jindal, S. and N. Kaur, 2016. Digital image steganography survey and analysis of current methods. Intl. J. Comput. Sci. Inf. Technol. Secur., 6: 10-13.

Johnson, N.F. and S. Jajodia, 1998. Exploring steganography: Seeing the unseen. Computer, 31: 26-34.

Kaur, B., A. Kaur and J. Singh, 2011. Steganographic approach for hiding image in DCT domain. Intl. J. Adv. Eng. Technol., 1: 72-78.

Khalaf, E.T. and N. Sulaiman, 2011. A robust data hiding technique based on LSB matching. World Acad. Sci. Eng. Technol., 5: 1092-1096.

Khosravi, M., S. Soleymanpour-Moghaddam and M. Mahyabadi, 2012. Improved pair-wise LSB matching steganography with a new evaluating system. Proceedings of the 6th International Symposium on Telecommunications (IST), November 6-8, 2012, IEEE, Tehran, Iran, ISBN:978-1-4673-2072-6, pp: 982-986.

Kim, Y., Z. Duric and D. Richards, 2007. Modified Matrix Encoding Technique for Minimal Distortion Steganography. In: Lecture Notes in Computer Science, Camenisch, J. et al. (Eds.). Springer, Berlin, Heidelberg, ISBN: 978-3-540-74123-7, pp: 314-327.

Kukapalli, V.R., B.T. Rao and B.S. Reddy, 2014. Image steganography by enhanced pixel indicator method using Most Significant Bit (MSB) compare. Intl. J. Comput. Trends Technol., 15: 97-101.

Kumar, M. and M. Yadav, 2014. Image steganography using frequency domain. Intl. J. Sci. Technol. Res., 3: 226-230.

Lee, Y.K., G. Bell, S.Y. Huang, R.Z. Wang and S.J. Shyu, 2009. An advanced least-significant-bit embedding scheme for steganographic encoding. Adv. Image Video Technol., 5414: 349-360.

Li, B., J. He, J. Huang and Y.Q. Shi, 2011. A survey on image steganography and steganalysis. J. Inform.Hiding Multimedia Signal Process., 2: 142-172.

Luo, W., F. Huang and J. Huang, 2010. Edge adaptive image steganography based on LSB matching revisited. IEEE Trans. Inform. Forensics Secur., 5: 201-214.

Luo, W., F. Huang and J. Huang, 2011. A more secure steganography based on adaptive pixel-value differencing scheme. Multimedia Tools Appli., 52: 407-430.

Muhammad, K., J. Ahmad, S. Rho and S.W. Baik, 2017. Image steganography for authenticity of visual contents in social networks. Multimedia Tools Appl., 76: 18985-19004.

Muhammad, K., M. Sajjad, I. Mehmood, S. Rho and S.W. Baik, 2016. A novel magic LSB substitution method (M-LSB-SM) using multi-level encryption and achromatic component of an image. Mult. Tools Appl., 75: 14867-14893.

Muhammad, K., M. Sajjad, I. Mehmood, S. Rho and S.W. Baik, 2018. Image steganography using uncorrelated color space and its application for security of visual contents in online social networks. Future Gener. Comput. Syst., 86: 951-960.

Nag, A., S. Biswas, D. Sarkar and P.P. Sarkar, 2010. A novel technique for image steganography based on block-DCT and huffman encoding. Int. J. Comput. Sci. Inform. Technol., 2: 103-112.

Petitcolas, F.A.P., R.J. Anderson and M.G. Kuhn, 1999. Information hiding-a survey. Proc. IEEE, 87: 1062-1078.

Radhakrishnan, R., M. Kharrazi and N. Memon, 2005. Data masking: A new approach for steganography?. J. VLSI Signal Process. Syst. Signal Image Video Technol., 41: 293-303.

Raja, K.B., C.R. Chowdary, K.R. Venugopal and L.M. Patnaik, 2005. A secure image steganography using LSB, DCT and compression techniques on raw images. Proceedings of 3rd International Conference on Intelligent Sensing and Information Processing, December 14-17, 2005, Bangalore, India, pp: 170-176.

Rani, J. and T.A. Khan, 2014. Performance optimized DCT domain watermarking technique with JPEG. Intl. J. Innovative Technol. Exploring Eng., 4: 20-24.

Senarathne, A.N. and K. De Zoysa, 2014. ILSB: Indexing with least significant bit algorithm for effective data hiding. Intl. J. Comput. Appl., 161: 28-42.

Shen, S.Y. and L.H. Huang, 2015. A data hiding scheme using pixel value differencing and improving exploiting modification directions. Comput. Secur., 48: 131-141.

Singh, K.U., 2014. A survey on image steganography techniques. Intl. J. Comput. Appl., 97: 10-20.

Swain, G. and S.K. Lenka, 2015. A novel steganography technique by mapping words with LSB array. Intl. J. Signal Imaging Syst. Eng., 8: 115-122.

Tiwari, A., S.R. Yadav and N. Mittal, 2014. A review on different image steganography techniques. Intl. J. Eng. Innovative Technol., 3: 121-124.

Tuama, A.Y., M.A. Mohamed, A. Muhammed and M.H. Zurina, 2017. Randomized pixel selection for enhancing LSB algorithm security against brute-force attack. J. Math. Stat., 13: 127-138.

Viswanatham, V.M. and J. Manikonda, 2010. A novel technique for embedding data in spatial domain. Intl. J. Comput. Sci. Eng., 2: 233-236.

Wang, S., B. Yang and X. Niu, 2010. A secure steganography method based on genetic algorithm. J. Inf. Hiding Multimedia Signal Process., 1: 28-35.

Wang, Y. and P. Moulin, 2006. Statistical modeling and steganalysis of DFT-based image steganography. Proceedings of SPIE International Conference on Security, Steganography and Watermarking of Multimedia Contents VIII Vol. 6072, February 15, 2006, SPIE, San Jose, California, USA., pp: 14-24.