A New Image Steganography Algorithm Based on MLSB Method with Random Pixels Selection

Odai M. Al-Shatanawi[1] and Nameer N. El. Emam[2]

[1]Department of Computer Science, Philadelphia University, Jordan
[2]Department of Computer Science, Philadelphia University, Jordan

ABSTRACT

In recent years, the rapid growth of information technology and digital communication has become very important to secure information transmission between the sender and receiver. Therefore, steganography introduces strongly to hide information and to communicate a secret data in an appropriate multimedia carrier, e.g., image, audio and video files. In this paper, a new algorithm for image steganography has been proposed to hide a large amount of secret data presented by secret color image. This algorithm is based on different size image segmentations (DSIS) and modified least significant bits (MLSB), where the DSIS algorithm has been applied to embed a secret image randomly instead of sequentially; this approach has been applied before embedding process. The number of bit to be replaced at each byte is non uniform, it bases on byte characteristics by constructing an effective hypothesis. The simulation results justify that the proposed approach is employed efficiently and satisfied high imperceptible with high payload capacity reached to four bits per byte.

KEYWORDS

Steganography; Image segmentation; Byte characteristic.

1.INTRODUCTION

Over a year's the flow of information in the twenty and twenty one century are rapid growth of information and the communication media using a large amount of data that exchanged over the Internet [1]. This growth of information encourages researchers to develop security techniques and to keep data transmission between sender and receiver safer from attackers [2].

The performance of steganography algorithms is based on many levels of security to produce stego images (stg) with high imperceptible [3]. These levels are added to be sure that the difficulties to extract the secret image (S) have been reached. Another factor that challenges the security level is the amount of payload capacities in the stego image (Stg) this factor should be calculated carefully to find the maximum number of bits from (S) that can embed into a cover image safely and more robustness. Numbers of metrics have been applied by many researchers to calculate error rate and brightness like mean square error (MSE), peak signal to noise ratio (PSNR), correlation coefficient (Corr.), Chi squire (χ^2), and standard deviation [4].

There are many Steganography algorithms proposed by many researchers, some of the algorithms are very complicated due to the long time needed to hide secret data, while the others are simple

DOI : 10.5121/ijnsa.2015.7203

methods with low complexity as in LSB (Least Significant Bit) [5, 6]. Spatial and frequency domains were used by the research to construct a steganography algorithm.

Many researchers working on frequency domain to hide secret information into JPEG images and to provide better camouflage but the embedding rate is limited [7].

Raftari, N., Moghadam, A. (2012) [8] proposed image steganography technique that combines the integer wavelet transformed (IWT) and discrete cosine transformed (DCT). This algorithm was constructed to embed a secret image in a frequency domain by using Munkres' assignment algorithm. Prabakaran et al., (2013) [9] present steganography approach in a frequency domain using DWT technique on both secret and cover images. Motamedi, H. (2013) [10] presented a wavelet-based method to perform image steganography in the frequency domain and utilize image denoising algorithms by wavelet shareholding. Steganographic algorithms are in general based on replacing noise components of a digital object to be used for hiding secret message.

In the spatial domain, the common ground of spatial steganography is directly changed the image pixel values for hiding data. The embedding rate is often measured in a bit per pixel (bpp). Ioannidou, A et al., (2012) [11] proposed a technique to produce image steganography, which belongs to techniques taking advantage of sharp areas in images in order to hide a large amount of data. Specifically, this technique is based on the edges present in an image. However, this approach cannot increase the payload capacity when the hiding process is working on smooth images or images with non sharp edges [12]. Hemalatha et al, (2013) [13]. Propose a method using two secret images to hide into one cover image to produce a high quality of a stg. However, the quality of Stg produced in this approach was not promising due to a large payload capacity (Hong, W., et al, 2010)

El-Emam, N., Al-Zubidy, R., (2013) [14] proposed steganography algorithm to hide a large amount of secret messages into a cover image by using four security layers. Moreover, this algorithm presents image segmentation algorithm and intelligent technique based on adaptive neural networks with genetic algorithm. However, this technique needs much time to produce high imperceptible Stg through four layers of security. Li, Y. et al (2010) [15], proposed a reversible data hiding method, Adjacent Pixel Difference (APD), which employs the histogram of the pixel difference sequence to increase the embedding capacity. This technique is working on gray image, and a PSNR measure is not enough to confirm the quality of Stg, in addition the author did not mention how to work against new attackers. Zhu, Y et al., (2012) [16] provide a general construction of steganography without any special assumptions and prove theoretically that the construction was a computationally secure stego system against adaptive chosen hidden text attacks. Wang et al, (2013) [17] used a reversible data hiding scheme based on histogram shifting in the spatial domain, the embedding capacity was increased, and image quality was enhanced by using wall and non-wall pixels. However, the author discussed the quality of image using PSNR and SSIM measures without attention to the effect of statically attack measures.

In this paper, we proposed new image steganography algorithm based on different size image segmentations (DSIS) and modified least significant bits (MLSB). The new hypothesis has been applied to measure byte characteristics and to fix the number of bit to be hide in the cover image.

The rest of the paper is structured as follows: In the section two, preliminary and definitions have been introduced to explain the theoretical concepts of steganography notations. The proposed steganography algorithm based on MLSB technique with new image segmentation has been presented in the section three. The prototype implementations are shown in the section four. The simulation results with their comparisons are presented in section five. Finally, the conclusion has been appeared in the section six.

2. PRELIMINARY AND DEFINITIONS

Some theoretical background to embed data into digital image has been introduced in this section to show how to improve three common requirements, (i) the security, (ii) the capacity, (iii) and the imperceptibility [18]. The performance of steganographic techniques is needed to confirm the security level with high payload and to demonstrate how to develop and implement the proposed technique to guarantee the authenticity of digital media. In Figure 1, the proposed steganography architecture has been constructed in this paper; it appears that we have two sides, the embedding and the extracting sides. In the first side, the embedding algorithm accepts three sets; these sets are: a set of non-uniform segments, a set of cover bytes, and set of integer values that represent the number of bit to be hiding at each pixel (NBTH). However, a set of non-uniform segments have been constructed by using DSIS algorithm while the set of NBTH have been estimated using new hypothesis based on byte characteristics. The output signals of the first side are a set of stego bytes Stg with high payload capacity and high imperceptible. In the second side, the system accepts the essential parameters as the input signals that represents a set of stego bytes and cipher key, whereas the output signal of this side is the set secret bytes S.

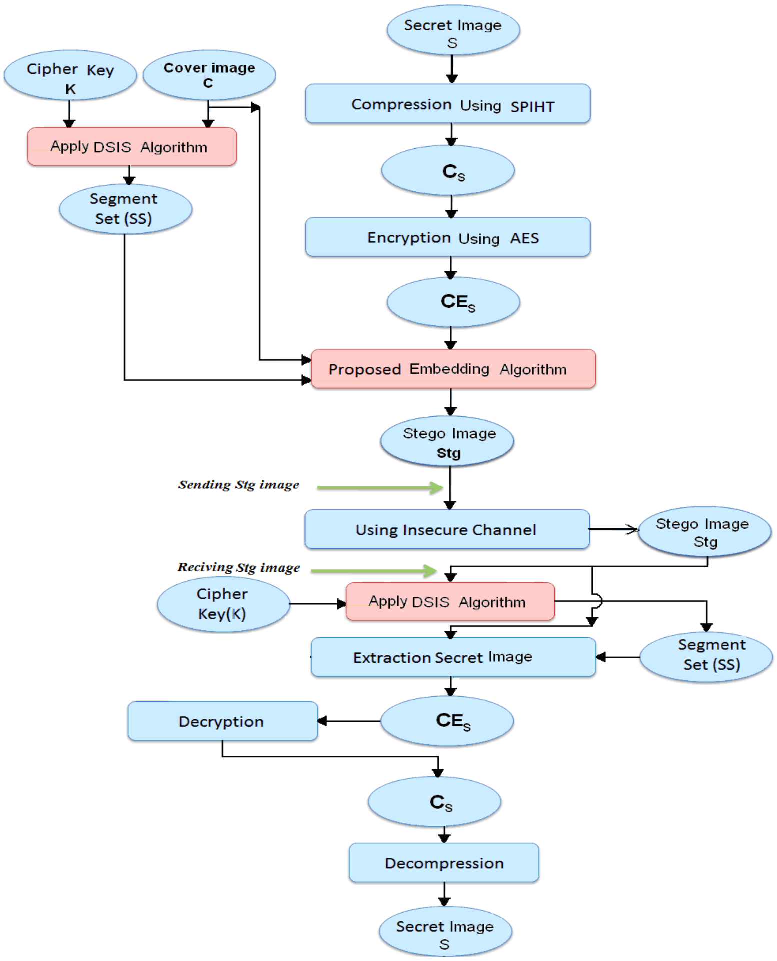


Figure1: The proposed steganography architecture

The definitions of the main components in the proposed algorithm have been discussed in the following:

Definition 1 Data compression function defines in the map $\mathrm{Comp} : S \times \hat{U}_m \times \hat{U}_S \times \breve{I}_m \rightarrow C_S$ where (Comp) is a compression technique, S is a secret image, $\hat{U}_m$ is un important image information, $\hat{U}_S$ is a list of un important sets, $\breve{I}_m$ is a list of important image information and C_S represents a secret image after compression.

An image compression is promising to save the storage and the time, in the proposed algorithm, we select lossless image compression approach based on set of partitions in hierarchal tree (SPIHT) algorithm [19, 20, 21]. The SPIHT method it provides lossless images.

Definition 2: Image encryption is defined in the map $\mathrm{Encry} : C_S \times \ell_{C_S} \rightarrow CE_S$, where C_S is a compressed secret image, ℓ_{C_S} is a length of C_S and CE_S represents a compressed secret image after encryption.

The AES algorithm has been applied to encrypt a compressed secret image. This algorithm is hard to crack, and it is well suitable to increase the security service in the applications. Moreover, AES algorithm needs low memory requirement and fast for the encryption process, so it is particularly well-suited to be used for the hiding algorithm [22].

Definition 3: Let image segmentation function define in the map $C_{DSIS} : I \times K \to SS$, where DSIS is the different size image segmentation algorithm, I is a cover or stego image, SS is the set of segments , each segment (Seg) is represented by segment's location using (x and y) coordinates with segments' edges ($X_{s,R}$ and $Y_{s,R}$) at the raster R, see Eqs. (2,3).

The purpose of using C_{DSIS} is to divide a cover image C into set non-uniform segments Seg_S and to scatter the secret bits on the segments, see Figure 2.

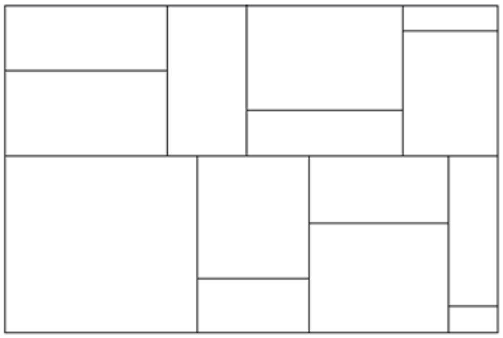


Figure 2: Non-Uniform image segmentation

Definition 4: The embedding function (EM) is represented in the map $EM : C \times SS \times CE_S \to Stg$ and it bases on byte characteristic assessment in a cover image C (to compute a number of bits to be hiding NBTH) , set of segments SS, and a secret image's compression and encryption CE_S; see section 3.

Definition 5: The extraction function (EX) is represented in the map $EX : Stg \times SS \to CE_S$ and it bases on byte characteristic assessment on stego image Stg to compute NBTH for each byte at each color and a set of segments SS, see section 3.

Definition 6: Image decryption is defined in the map $IDcry : CE_S \times \ell_{C_S} \to C_S$

Definition 7: Data decompression function DEC defines in the map $DEC : C_S \to S$, where the function domain contains a compressed secret image C_S, while the function range contains a secret image after decompression S.

Definition 8: Let (NB) represents the set of eight neighboring bytes around the target byte (TB) [23], see Eq.(1). The locations of NB are illustrated in the Figure3.

$$NB\left(TB_{i,j}\right) = \bigcup_{s=i-1}^{i+1} \bigcup_{r=j-1}^{j+1} NB_{r,s}\left(TB_{i,j}\right) \quad \text{such that if } s = i \text{ then } r \neq j \tag{1}$$

$NB_{i-1,j-1}$	$NB_{i,j-1}$	$NB_{i+1,j-1}$
$NB_{i-1,j}$	$TB_{i,j}$	$NB_{i+1,j}$
$NB_{i-1,j+1}$	$NB_{i,j+1}$	$NB_{i+1,j+1}$

Figure 3: Eight neighbour bytes

3.STEGANOGRAPHY ALGORITHM BASES ON MLSB TECHNIQUE

The proposed new steganography algorithm follows a set of rules to guide us to create a stego image Stg that produces after embedded secret image S into a cover image C. In the other hands, we implement an extraction rules to reconstruct a secret image S. In Figure1, we show the main components that are used to implement hiding/ extracting processes, where the proposed steganography algorithm (sender side) is based on two parts. The first part aims to construct different size image segmentations (DSIS) from cover image to scatter secret data randomly, while the second part aims to build an effective approach to embed a secret image into a cover image with high imperceptible to works against attacks under high payload.

3.1 Image segmentation algorithm:

Image segmentation is the process that uses to partition cover image into a set of sub images depending on a new hypothesis. Different methods proposed by many researchers had been implemented to achieve image segmentation based on the value of intensity, similarity, and variance between neighboring bytes. In the proposed algorithm, the hypothesis that is created is

based on cipher key with three operations to make hard to detect the segments edges from the attacker.

In Figure 4 we explain the proposed image segmentation based on partitioning a cover image into different segments' sizes. This cover image contains three layers red, green and blue; each layer has a two-dimensional array ($W_C \times H_C$) where W_C and H_C are the width and the height of a cover image C respectively.

The size of segment (s) is based on two variables, the first is variable is a length of width of segment s represented by ($X_{s,R}$), whereas the second variable is a length of height of segment s represented by $Y_{s,R}$, see Eq. (2-3). The cipher key K has been used to generate $X_{s,R}$ and $Y_{s,R}$ for each segment. We believe that image segmentation is an excellent approach to work against attack by hiding secret message randomly and reduced the possibility for detection with probability $\frac{1}{|SS|}$ where $|SS|$ is the number of segments in a cover image.

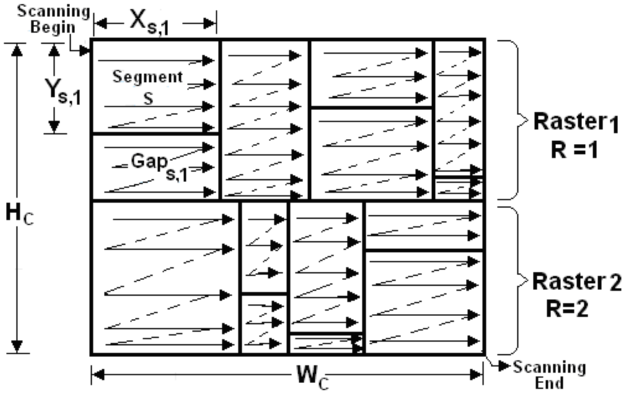


Figure 4: Using non-uniform image segmentation base on DSIS algorithm

The size of each segment s at each raster R is equal to $X_{s,R} \times Y_{s,R}$, where $X_{s,R}$ and $Y_{s,R}$ are calculated using Eqs.(2-3).

$$X_{s,R} = \left\lceil \sqrt{\frac{W_C}{\hbar}} \right\rceil + \hbar \tag{2}$$

$$Y_{s,R} = \left\lceil \frac{\sqrt{H_C + A + B}}{\bar{\lambda}} \right\rceil + \bar{\lambda} \tag{3}$$

where $\hbar$ is equal to $\mathrm{Val}(\mathrm{Str}(B)+\mathrm{Str}(C))$ (the decimal value of the concatenation of B and C strings) and $\bar{\lambda}$ is equal to $\mathrm{Val}(\mathrm{Str}(A)+\mathrm{Str}(B))$ (the decimal value of the concatenation of A and B strings) , where Str(.) function is the convertor from decimal to string value whereas Val(.) function is the convertor from string to decimal value. In addition, the variables A, B, and C are calculated using Eqs. (4-6).

$$A = \left\lfloor \frac{F}{100} \right\rfloor \tag{4}$$

$$B = \left\lfloor \frac{F - A \times 100}{10} \right\rfloor \tag{5}$$

$$C = (F - (A \times 100) - (B \times 10)) \tag{6}$$

where F is define in the Eq. 7.

$$F = \aleph - \mathrm{Val}\left(\mathrm{Str}(M_S)^r\right) \tag{7}$$

Such that $\aleph$ is the constant equal to 300, $(M_S)^r$ is reversed order of MS. and MS is defined in the Eq. 8.

$$M_S = \mathrm{Val}\left(K_S\right) \quad \forall s = 1,\ldots,|SS| \qquad (8)$$

The size of $\mathrm{Gap}_{s,R}$ shown in Figure 4 and it is appeared under the segment s at raster R is calculated using Eqs. (9).

$$\mathrm{Gap}_{s,R} = X_{s,R} \times \Big(\max_{\substack{\forall \text{ segment} \\ \text{index (i) in R}}} (Y_{i,R}) - Y_{s,R} \Big) \qquad (9)$$

The necessary conditions that should be reach is $\left(\sum_{\forall s \text{ in } R} X_{s,R} = W_c \ \wedge \sum_{\forall s \text{ in} R} \left(\max_{\forall s} \left(Y_{s,R}\right)\right) = H_c \right)$

The proposed segmentation algorithm (DSIS) is constructed to calculate the size of each segment conformity according to the following steps:

Algorithm1: Image segmentation DSIS

```
Step1: Input K, Cover image C and $\aleph$ ;
Step2: For each color in C;     // $\text{color} \in \{R, G, B\}$.
Step3: For each raster R in the C;
        Step 3-1: For each segment s in the raster R;
            Step 3-1-1: Compute MS ;     // Using Eq.(8).
        Step 3-1-2: Compute F;       // Using Eq. (7).
                Step 3-1-3: Compute A;       // Using Eq. (4).
                Step 3-1-4: Compute B;       // Using Eq. (5).
                Step 3-1-5: Compute C;       // Using Eq. (6).
                Step 3-1-6: Compute $X_{s,R}$ ;     // Using Eq. (2).
                Step 3-1-7: Compute $Y_{s,R}$ ;     // Using Eq. (3).
                Step 3-1-8: Compute $\mathrm{Gap}_{s,R}$ ; // Using Eq. (9).
        End;        // foreach segment s.
    End;        // foreach raster R.
End.// End Algorithm1
```

The time complexity measure of DSIS is defined using "Big- O" notation, where the time required for each segment is defined in Eq. (10):

$$\mathrm{TimeSeg}_s = T(Ms) + T(F) + T(A) + T(B) + T(C) + T(X_s) + T(Y_s) \approx O(7) \qquad (10)$$

Moreover, the time required for all segments for all coloris define in Eq. (11):

$$T(\mathrm{TimeSeg}_s, |SS|) = O(\mathrm{TimeSeg}_s \times 3 \times |SS|) \approx O(7 \times 3 \times |SS|) \approx O\left(21 \times |SS|\right) \qquad (11)$$

3.2 Byte characteristic assessment in the embedding algorithm:

Bytes' characteristics have been used in the proposed algorithm to find a number of bit(s) to embed secret bit(s) at each byte for each color in a cover image, these secret bits are hidden without any suspicion form steganalysis for both visual and statically attacks [23 , 25]. The proposed algorithm depends on the variance measure of the target byte ($TB_{i,j}$) and its eight neighbored byte (NB) , where the embedding process is based on scanning bytes from the upper left to the lower right of a cover image.

In this work, we apply byte value reduction function $\mathrm{BVR}(.)$ to damp byte intensity from (0-255) to (1-16), see Eq. (12). The benefits of using sixteen levels instead of 265 levels are to reduce the number of classification levels of each byte and the calculation of the variance for each byte should be faster [14].

$$\mathrm{BVR}\left(\mathrm{byte}_{i,j}\right)=\left\lfloor \frac{\mathrm{byte}_{i,j}}{16}+1 \right\rfloor \qquad (12)$$

where $\mathrm{byte}_{i,j}$ represents a target byte $\mathrm{TB}_{i,j}$ at the location (i, j) or neighbored bytes NB . Where the surrounding bytes $\mathrm{NB}\left(\mathrm{TB}_{i,j}\right)$ around the $\mathrm{TB}_{i,j}$ are defined in the set:

$$\left\{\mathrm{NB}_{i-1,j-1}\left(\mathrm{TB}_{i,j}\right), \mathrm{NB}_{i,j-1}\left(\mathrm{TB}_{i,j}\right), \mathrm{NB}_{i+1,j-1}\left(\mathrm{TB}_{i,j}\right), \mathrm{NB}_{i-1,j}\left(\mathrm{TB}_{i,j}\right), \mathrm{NB}_{i+1,j}\left(\mathrm{TB}_{i,j}\right), \mathrm{NB}_{i-1,j+1}\left(\mathrm{TB}_{i,j}\right), \mathrm{NB}_{i,j+1}\left(\mathrm{TB}_{i,j}\right), \mathrm{NB}_{i+1,j+1}\left(\mathrm{TB}_{i,j}\right)\right\}$$

$\mathrm{BVR}\left(\mathrm{NB}_{i-1,j-1}\left(\mathrm{TB}_{i,j}\right)\right)$	$\mathrm{BVR}\left(\mathrm{NB}_{i,j-1}\left(\mathrm{TB}_{i,j}\right)\right)$	$\mathrm{BVR}\left(\mathrm{NB}_{i+1,j-1}\left(\mathrm{TB}_{i,j}\right)\right)$
$\mathrm{BVR}\left(\mathrm{NB}_{i-1,j}\left(\mathrm{TB}_{i,j}\right)\right)$	$\mathrm{BVR}\left(\mathrm{TB}_{i,j}\right)$	$\mathrm{BVR}\left(\mathrm{NB}_{i+1,j}\left(\mathrm{TB}_{i,j}\right)\right)$
$\mathrm{BVR}\left(\mathrm{NB}_{i-1,j+1}\left(\mathrm{TB}_{i,j}\right)\right)$	$\mathrm{BVR}\left(\mathrm{NB}_{i,j+1}\left(\mathrm{TB}_{i,j}\right)\right)$	$\mathrm{BVR}\left(\mathrm{NB}_{i+1,j+1}\left(\mathrm{TB}_{i,j}\right)\right)$

Figure 5: the BVR (.) of the target byte (TP) and the eight neighbour bytes (NB)

New hypothesis has been proposed based on variance calculation between target bytes TB, and its eight neighbored bytes see Eq. (13). This hypothesis has been used to calculate a number of bits to be hide (NBTH),

$$\mathrm{NBTH}_{i,j}=\begin{cases}\left\lceil\left(\mathrm{EXP}\left(\dfrac{\sigma^{2}\left(\mathrm{BVR}\left(\mathrm{TB}_{i,j}\right)\right)}{\underset{\forall s,r}{\operatorname{argmax}}\left(\sigma^{2}\left(\mathrm{BVR}\left(\mathrm{TB}_{i,j}\right)\right)\cup\sigma^{2}\left(\mathrm{BVR}\left(\mathrm{NB}_{s,r}\left(\mathrm{TB}_{i,j}\right)\right)\right)\right)}\right)\times 0.6\right)^{1.7}\right\rceil & \forall i\in\left[2,\mathrm{W}_{C}-1\right]\vee\forall j\in\left[2,\mathrm{H}_{C}-1\right]\\ 4 & \forall i\in\left\{1,\mathrm{W}_{C}\right\}\vee\forall j\in\left\{1,\mathrm{H}_{C}\right\}\end{cases} \qquad (13)$$

where σ^{2} is the variance value of the target byte BVR(TB) and its neighbor eight bytes BVR(NB) , the value of these bytes are changed based on byte value reduction (BVR) to extract the high nibble of the byte that are not used by the hiding algorithm. The value of $\mathrm{NBTH}_{i,j}$ is in the range [1,4] and the proposed hypothesis checks the variations between the target byte and its surrounding eight bytes to estimate a number of bits to be replaced.

In general, steganography algorithm contains two parts; the first part is the sender that contains the embedding algorithm based on $EM(.)$ function while the second part is the receiver that contains the extracting algorithm based on $EX(.)$ function [26].

The proposed embedding algorithm EM (.) is based on MLSB and it includes the following steps:

Algorithm2: Embedding algorithm.

```
Step 1: Input Cover image (C), cipher key (K) and Secret image (S).
Step 2: Apply $Comp(S, \hat{U}_m, \hat{U}_S, \breve{I}_m)$ function to produce $C_S$ ; // see definition 1.
Step 3: Apply $IEncry(C_S, \ell_{C_S})$ function to produce $CE_S$ ; // see definition 2.
Step4: For each color in C;
   Step4-1: Apply $C_{DSIS}(C, K)$ function to produce non uniform segments Stg by
calculating $X_{s,R}$ and $Y_{s,R}$ ; // see definition 3 and Eqs. (2-9).
   Step4-2: Call Byte_Characteristic(.) to find NBTH; // see Sub-Algorithm2.
   Step4-3: Perform embedding function $EM(C, SS, CE_S)$ of the secret image's compression
and encryption ($CE_S$); // see definition 4.
Step 5: Send Stg image to insecure channel;
End. //End Algorithm2
```

Sub-Algorithm2 *// Set the intensity of each byte in the range (1, 16) and then find* $NBTH_{i,j}$ *for each byte at each color; see Eq. (13)*

```
 Byte_Characteristic( NBTH ) {
      // scanning all bytes for all segment at each color in a cover image
 For each color in C
          For each segment $Seg_s$
                 For each byte B
                    Calculate $BVR_B$ ;        // using Eq. (12).
                    Calculate $NBTH_B$ ;       // using Eq. (13).


End. // for each segment set.
         End. // for each segment set.
End. //for each color.
}// end sub-algorithm2.
```

Algorithm3- Extraction algorithm

```
Step1: Input Stg image and the K that are received from the insecure channel;
Step2: For each color in Stg;
   Step2-1: Applying $C_{DSIS}(Stg, K)$ to find the edges of each segment $Seg_S$ by
calculating $X_{s,R}$ and $Y_{s,R}$ ; // see Eq.(2-3).
   Stp2-2: Scan all bytes in Stg image and calculate NBTH; // Eq.(13).
```

Step2-3: **Apply** EX function EX(Stg , SS) for all bytes depending on byte characteristics to produce CE_S ; *// see definition 5.*

Step3: **Apply** $IDcry(CE_S \times \ell_{C_S})$ function to produce C_S ; *// see definition 6.*

Step4: **Apply** $DEC(C_S)$ function to find a secret image S; *// see definition 7.*

End. *// End Algorithm3*

The "big-O" notation has been applied to measure time complexity for data embedding (EM) and data extraction (EX), time complexity is defined in Eq. (14).

$$T(EM) = T(EX) = \sum_{color=1}^{3} \sum_{S=1}^{|SS|} \sum_{byte=1}^{\max_{\forall s,\forall R}(X_{s,R}\times Y_{s,R})} MLSB \approx O\left(3\times|SS|\times\max_{\forall s,\forall R}\left(X_{s,R}\times Y_{s,R}\right)\right) \qquad (14)$$

4. IMPLEMENTATION

The implementation of the proposed embedding algorithm has been applied by using MLSB on three colors to hide a secret image. We can hide one to four bit(s) depending on the value of NBTH by using Eq. (13). In Figure 6, we applied the proposed embedding algorithm on selective cover image (F16) since the difference in byte characteristics has been shown on three colors.

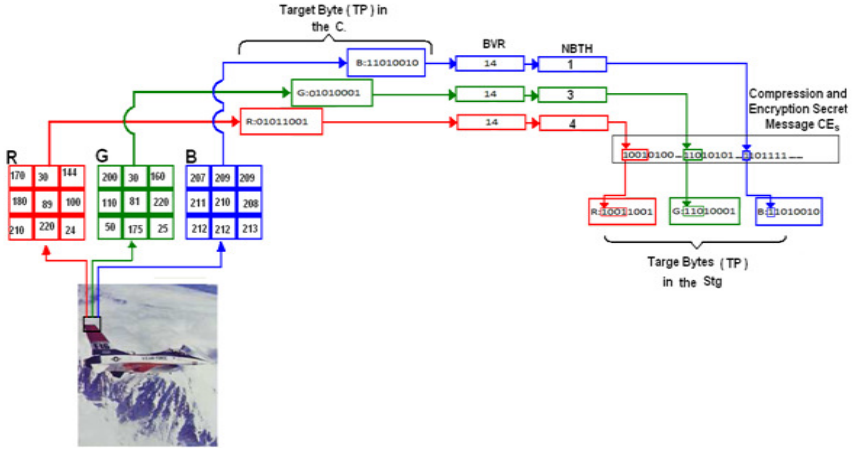


Figure 6: Find a number of bits to be replaced for each byte and at each color (R, G, B) with the embedding process

Furthermore, the proposed steganography algorithm calculates the value of NBTH for each color to minimize the distortion on stego image [27]; it appears that the read color has a highest value of NBTH equal to four due to large variance between the TB and the surrounding NBs while the blue color has the lowest value of NBTH equal to one due to small variance between TB and the surrounding NBs.

5. RESULTS AND DISCUSSIONS

The proposed algorithm using modified LSB has been implemented using MATLAB environment. The performance of the proposed approach has been studded using different kinds of measures like (amount of payload capacity, PSNR, MAE, AD, and NCC).

To confirm the performance of the proposed approach, we apply the proposed algorithm on more than 200 images from ((BOSS base version. 0.92) database. In this section we display the results using four testing color images, these are: (Lena, F16, Baboon, Peppers and Tiffany), see Fig 7.

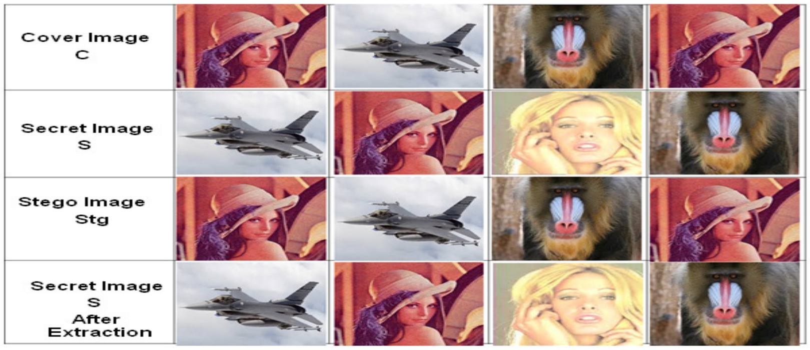


Figure 7: Five testing images.

The quality of stego image stg has been studded using peak signals to noise ratio (PSNR), see Eq.(15).

$$\mathrm{PSNR} = 10 \times \log\left(\frac{(255)^2}{\mathrm{MSE}_{\mathrm{avg}}}\right) \qquad (15)$$

where $\mathrm{MSE}_{\mathrm{avg}}$ is the average of MSE for three colors (R, G, B).

$$\mathrm{MSE}_{\mathrm{avg}} = \frac{\mathrm{MSE}_{\mathrm{R}} + \mathrm{MSE}_{\mathrm{G}} + \mathrm{MSE}_{\mathrm{B}}}{3} \qquad (16)$$

In Table 1, we display the value of PSNR for the stego image Stg after hiding the secret image's compression and encryption.

Table 1: Calculate PSNR values Eq.(15) for different cover and secret images size (256*256).

Cover image (256*256)	**Channels (R,G,B)**	**Secret Image (256X256)**	**PSNR of the Proposed Algorithm**
Lena	Red-1	F16	43.436
	Green-2		43.2875
	Blue-3		43.462
	All		43.39
Lena	Red-1	Baboon	43.6466
	Green-2		43.5839
	Blue-3		43.6
	All		43.61
F16	Red-1	Lena	44.0967
	Green-2		44.0845
	Blue-3		44.56
	All		44.2470

We observed that PSNR of the tested images using the proposed hiding algorithm has a maximum average able to (44.2470 dB) when the cover image is (F16), and the secret image is (Lena), while the minimum average is equal to (43.39 dB) when the cover image is (Lena), and secret image is (F16). Moreover, the results illustrated that blue channels have the maximum sum of PSNR equal to (131.622 dB) while the green channels have the minimum sum of PSNR equal to (130.9559 dB) about 0.5319% less than the blue channel. However, the result indicates that the amount of secret data into the green channel should be reduced to avoid perceptible of secret image by attackers.

The variances of cover and the stego images (F16) have been shown the Figure 8 when the secret image is Lena. Histograms in Figure 8 (a) and (b) refer to the variance of the cover and stego images respectively using red channel, whereas histograms in Figure 8 (c) and (d) refer to the variance of the cover and stego images respectively using Green channel and histograms in Figure 8 (e) and (f) refer to the variance of the cover and stego images respectively using blue channel.

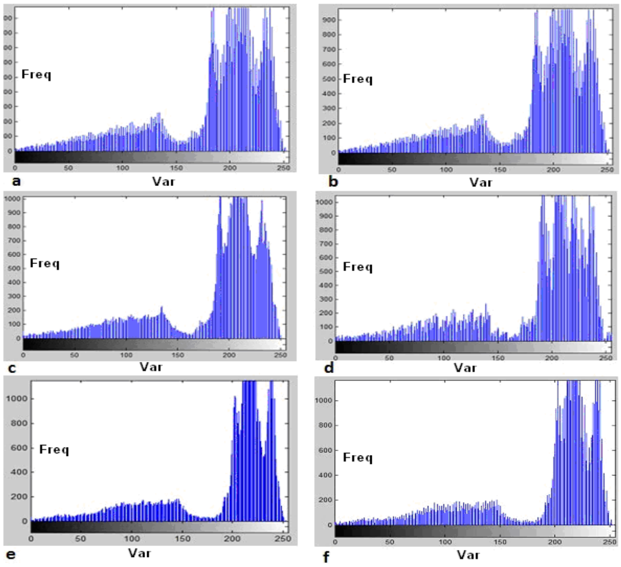


Figure 8: Histograms of different layers of the cover image and the corresponding stg image. (a) And (b) for red channel. (c) And (d) for the green channel. (e) And (f) for the blue channel.

In addition, the histograms in Figure 8 show that the matching between cover and stego images has been satisfied at the red channel while the noise are appeared at stego image in the blue channel.

Table 2 shows the PSNR values for different payload percentages on the F16 as the cover image. It appears that the percentage of the payload (amount of bits to be hidden) have highest PSNR at the payload percentage equal to 10%. In addition, the results appear that PSNR value is decreased when the payload percentage has been increased, where the percentage of PSNR has been reduced about 9.63% for the payload percentage equal to 20% and has been reduced about 12.87% for the payload percentage equal to 30% and has been reduced about 16.93% for the payload percentage equal to 50%.

Table 2: PSNR OF Stg image with different payload

Cover image	**Payload 10%**	**Payload 20%**	**Payload 30%**	**Payload 50%**
F16	53.43db	48.28db	46.55db	44.38db

In Table 3, a comparative study with other researchers has been taken up with the same circumstances (same cover images, same secret images, and same image size). These comparisons are applied between the proposed approach and the two previous according to the value of PSNR. The results confirm obviously that the proposed method is more secure and preserved secret information than the other steganographic schemes. It appears that the average of three stego images in the proposed approach is better than (EL-EMAM, N. 2013) [14] and (Chang, C., 2008) [28] about 11.23% and 14.42% respectively.

Table 3: Comparison with other researcher works

Cover image (512 X 512)	**Channels (R,G,B)**	**Secret image**	**PSNR (Chen. 2008) [28]**	**PSNR EL-EMAM, N. 2013 [14]**	**The Proposed algorithm**	**The Percentage to improve the other works**	
						EL-EMAM, 2013 [14]	**Chen. 2008 [28]**
Lena	Red	Peppers	37.97	39.01	42.97	8.12%	10.28%
	Green		37.87	39.42	42.94		
	Blue		39.78	39.98	43.05		
	All		38.54	39.47	42.96		
F16	Red	Lena	36.32	37.45	44.88	15.33%	19.02%
	Green		35.55	37.12	44.95		
	Blue		37.43	39.71	45.14		
	All		36.43	38.09	44.99		
Baboon	Red	Tiffany	37.39	39.21	42.45	10.24%	13.96%
	Green		36.38	37.98	42.4		
	Blue		35.85	37.17	42.57		
	All		36.54	38.12	42.47		

In Table 4, the performance of the proposed algorithm has been checked using five measures; these measures have been discussed through the PSNR, see Eq. (15), the mean absolute error (MAE), see Eq. (17), the average difference (AD), see Eq. (18), and normalized cross correlation (NCC), see Eq. (19).

$$\mathrm{MAE} = \frac{1}{W_C \times H_C} \sum_{\forall i,\forall j} \left| C_{i,j} - Stg_{i,j} \right| \tag{17}$$

$$\mathrm{AD} = \frac{1}{W_c \times H_C} \sum_{i=1}^{W_C} \sum_{j=1}^{H_C} \left| C_{i,j} - Stg_{i,j} \right| \tag{18}$$

$$\mathrm{NCC}(C, Stg) = \frac{\sum_{\forall r,\forall t} \left(\left(C_{r,t} - \overline{C} \right) \left(Stg_{r,t} - \overline{Stg} \right) \right)}{\left(\sum_{\forall r,\forall t} \left(w_{r,t} - \overline{w} \right)^2 \left(Stg_{r,t} - \overline{Stg} \right)^2 \right)^{0.5}} \tag{19}$$

where $\overline{C}$ is the mean of cover image while $\overline{Stg}_{i,j}$ is the mean of stego image.

Table 4: Check the performance of the proposed algorithm through different measures

Cover image	**Channels (R,G,B)**	**Secret image**	**PSNR**	**Payload**	**NCC**	**AD**	**MAE**
Lena	Red	F16	43.436	100648	0.9986	0.2563	0.0057
	Green		43.2875	102521	0.9976	0.2586	0.0107
	Blue		43.462	101424	0.9976	0.2582	0.0097
	All		43.39	304593	0.9979	0.2577	0.0087
Lena	Red	Baboon	43.6466	100648	1	0.0211	0.0056
	Green		43.5839	102521	0.999	0.0209	0.0105
	Blue		43.6	101424	0.999	0.0098	0.0096
	All		43.61	304593	0.9993	0.0172	0.0085
F16	Red	Lena	44.0967	91715	0.999	0.1708	0.005
	Green		44.0845	90258	0.9991	0.173	0.0049
	Blue		44.56	89561	0.9993	0.1451	0.0045
	All		44.2470	271534	0.999	0.1629	0.0048

The experimental results in Table 4 have been considered on many color images to check the performance using the largest amount of payload capacity. The results illustrate that the quality of stego image Stg has been reached according to those measures. In addition, results show that the high quality has been reached when AD and MAE are small, PSNR is large and NCC tends to one. Therefore, when the stego image is Lena and the secret image is Baboon the relative quality in the maximum, while when the stego image is Lena and the secret image is F16, the relative quality in the minimum. Moreover, the results show that the payload capacities for three stego images are different; they appear that the stego image Lena that holds Baboon or F16 as secret images is better than the stego image F16 that holds Lena secret image about 10.85%.

6. CONCLUSIONS

This paper presented a description of a new steganography algorithm. The algorithm is employed effectively over an insecure channel and working against attacks by producing high imperceptible steg images for both low and high payload. The proposed steganography algorithm bases on many components, these components are:

i) DSIS algorithm to generate set of non-uniform segments. These segments are employed to hide a secret image randomly instead of sequentially. This approach can decrease the probability of detection to ($\frac{1}{|SS|}$).

ii) Using DWT to get a high lossless compression ratio to increase the amount of the secret image that can be sent [29].

iii) Apply advanced encryption standard (AES) to make a secret image unreadable by attackers.

iv) Modified the traditional LSB to embed more than one bit for each byte with high imperceptible. The aim of MLSB to increase the payloads and to improve the security.

The proposed approach justifies the security according to experimental results shown in this paper.

ACKNOWLEDGMENTS

The authors would like to thank Prof. R. H. Al-Rabeh from Cambridge University for his supported and help. This support is gratefully acknowledged.

REFERENCES

[1] Abduallah, W. M., Rahma, A. M. S., Pathan, A. S. K. (2014). Mix column transform based on irreducible polynomial mathematics for color image steganography: A novel approach. Computers & Electrical Engineering, 40(4), 1390-1404.

[2] Fridrich, J., Kodovsky, J. (2012). Rich models for steganalysis of digital images. Information Forensics and Security, IEEE Transactions on, 7(3), 868-882.

[3] Ganesan, P., Bhavani, R. 2013. A High secure and robust image steganography using dual wavelet and blending model. Journal of Computer Science, 9(3).

[4] Rashid, M. K. R., Missen, S., Rashid, A., |(2014). Robust Increased Capacity Image Steganographic Scheme. Transformation, 5(11).

[5] Khalind, O., Aziz, B. (2014). Detecting 2LSB steganography using extended pairs of values analysis. Published in SPIE Proceedings Vol. 9120 Mobile Multimedia/Image Processing, Security, and Applications: (pp. 912003-912003). International Society for Optics and Photonics. doi: 10.1117/12.2048880

[6] Juneja, M., Sandhu, P. S. (2014). Improved LSB based Steganography Techniques for Color Images in Spatial Domain. IJ Network Security, 16(4), 366-376.

[7] Andriotis, P., Oikonomou, G., Tryfonas, T. (2013). JPEG steganography detection with Benford's Law. Digital Investigation, 9(3), 246-257.

[8] Raftari, N., Moghadam, A. M. E., (2012). Digital Image Steganography Based on Assignment Algorithm and Combination of DCT-IWT. InComputational Intelligence, Communication Systems and Networks (CICSyN), 2012 Fourth International Conference on (pp. 295-300).

[9] Prabakaran, G., Bhavani, R., Kanimozhi, K. (2013). Dual transform based steganography using wavelet families and statistical methods. In Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on (pp. 287-293).

[10] Mohammadi, F. G., Abadeh, M. S. (2014). Image steganalysis using a bee colony based feature selection algorithm. Engineering Applications of Artificial Intelligence, 31, 35-43.

[11] Ioannidou, A., Halkidis, S. T., Stephanides, G. (2012). A novel technique for image steganography based on a high payload method and edge detection.Expert Systems with Applications, 39(14), 11517-11524.

[12] Hong, W., Chen, T. S., Chang, Y. P., Shiu, C. W. (2010). A high capacity reversible data hiding scheme using orthogonal projection and prediction error modification. Signal Processing, 90(11), 2911-2922.

[13] Hemalatha, S., Acharya, U. D., Renuka, A., Kamath, P. R. (2013). A secure and high capacity image steganography technique. Signal & Image Processing, An International Journal, 4(1), 83-89.

[14] El-Emam, N., Al-Zubidy, R., (2013). New steganography algorithm to conceal a large amount of secret message using hybrid adaptive neural networks with modified adaptive genetic algorithm. Journal of Systems and Software, 86(6), 1465-1481, Elsevier.

[15] Li, Y. C., Yeh, C. M., Chang, C. C. (2010). Data hiding based on the similarity between neighbouring pixels with reversibility. Digital Signal Processing, 20(4), 1116-1128.

[16] Zhu, Y., Yu, M., Hu, H., Ahn, G. J., Zhao, H., (2012). Efficient construction of provably secure steganography under ordinary covert channels. Science China Information Sciences, 55(7), 1639-1649.

[17] Wang, X. T., Chang, C. C., Nguyen, T. S., Li, M. C., (2013). Reversible data hiding for high quality images exploiting interpolation and direction order mechanism. Digital Signal Processing, 23(2), 569-577.

[18] Thai, T. H., Retraint, F., Cogranne, R., (2014). Statistical detection of data hidden in least significant bits of clipped images. Signal Processing, 98, 263-274.

[19] Dubey, V., Dubey, R. (2013). A new Set Partitioning in Hierarchical (SPIHT) Algorithm and Analysis with Wavelet Filters. International Journal of Innovative Technology and Exploring Engineering, 3(3), 125-128.

[20] Reddy, K. S. N., Reddy, B. R. S., Rajasekhar, G., Rao, K. C., (2012). A Fast Curvelet Transform Image Compression Algorithm using with Modified SPIHT.International Journal of Computer Science and Telecommunications, 3(2), 1-8.

[21] Chowdhury, M. M. H., Khatun, A. (2012). Image Compression Using Discrete Wavelet Transform. IJCSI International Journal of Computer Science Issues,9(4).

[22] Shingote, P. N., Syed, A., Bhujbal, P. M., (2014). Advanced Security Using Cryptography and LSB Matching Steganography. IJCER, 3(2), 52-55.

[23] De Kok, R. (2012). Spectral Difference in the Image Domain for Large Neighborhoods, a GEOBIA Pre-Processing Step for High Resolution Imagery.Remote Sensing, 4(8), 2294-2313.

[24] Kanan, H. R., Nazeri, B. 2014. A novel image steganography scheme with high embedding capacity and tunable visual image quality based on a genetic algorithm. Expert Systems with Applications, 41(14), 6123-6130.

[25] Motamedi, H., Jafari, A. (2012). A new image steganography based on denoising methods in wavelet domain. In Information Security and Cryptology (ISCISC), 2012 9th International ISC Conference on (pp. 18-25).

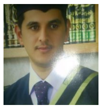
[26] Al-Bahadili, H. (2013). A Secure Block Permutation Image Steganography Algorithm. Submitted to the International Journal on Cryptography and Information Security. 3(3), 11-22.

[27] Filler, T., Judas, J., Fridrich, J. (2011). Minimizing additive distortion in steganography using syndrome-trellis codes. Information Forensics and Security, IEEE Transactions on, 6(3), 920-935.

[28] Chang, C., Lin, C., Fan, Y., (2008). Lossless data hiding for color images based on block truncation coding. Pattern Recognition 41, 2347–2357, Elsevier.

[29] Jayprkash, A., Vijay, R. (2013). Compression of MR Images Using DWT by Comparing RGB and YCbCr Color Spaces. Journal of Signal and Information Processing, 4, 364.

Authors



Odai M. Al-Shatanawi: He received the B.S. degree in computer science from the AOU, Amman, Jordan, in 2006, and the M.S. degree in computer science from Philadelphia University, Amman, Jordan, in 2015. He has CCNA certified. His current research interest is computer security using steganography.

Nameer N. EL-Emam: He completed his PhD with honor at Basra University in 1997. He works as an assistant professor in the Computer Science Department at Basra University. In 1998, he joins the department of Computer Science, Philadelphia University, as an assistance professor. Now he is an associated professor at the same university, and he works as a chair of computer science department and the deputy dean of the faculty of Information Technology, Philadelphia University. His research interest includes Computer Simulation with intelligent system, Parallel Algorithms, and Soft computing using Neural Network, GA, ACO, and PSO for many kinds of applications like Image Processing, Sound Processing, Fluid Flow, and Computer Security (Seteganography).