Journal of Computer Science

Original Research Paper

Full Capacity Image Steganography Using Seven-Segment Display Pattern as Secret Key

Mohammed A. Fadhil Al-Husainy and Hamza Abbass A. Al-Sewadi

Department of Computer Science, Middle East University, Amman, Jordan

Article history
Received: 5-01-2018
Revised: 27-02-2018
Accepted: 14-04-2018

Corresponding Author:
Hamza Abbass A. Al-Sewadi
Department of Computer Science, Middle East University, Amman, Jordan
Tel: 00962 795 90 6054 and 00962 785 35 6187
Email: hsewadi@meu.edu.jo
alsewadi46@gmail.com

Abstract: Due to the vast interchange of images over the internet, exchanging secret message hidden in these images would be encouraged and the technique is referred to as image steganography. Hiding the secret data into the least significant bits of the images pixels is a common practice. However, the choice of the hidden key and the efficiency of utilizing all pixels of the carrier image is of great importance. This paper presents a new technique for key generation and embedding/extraction processes for image steganography. The randomness of secret key required for the embedding process is achieved by using the seven segment display patterns with different dimensions. Full embedding efficiency is also achieved through. Experimental measurements and comparison with traditional steganography using LSB have confirmed the feasibility of the proposed scheme. As it produces highly imperceptible stego-images, besides the key randomness for selecting the pixels, which leads to the addition of huge difficulties against attackers and provides genuine protection for the hidden information.

Keywords: Data Hiding, Image Steganography, Hexadecimal Secret Key, Seven Segment Display

Introduction

In recent years the vast computing speed and the heavy use of the internet have encouraged researcher to develop a tremendous number of computer security algorithms. Generally, data security can be achieved either by changing the plaintext data into ciphertext using one of the cryptographic techniques (Stalling, 2017; Schneier, 1996) or hide the secret data into another media by one of the steganography methods (Ingemar *et al*., 2008), in such way that intruders are deceived by the innocence appearance of the media and does not discover the hidden data inside it. This paper is concerned with data hiding rather than cryptography and hence will consider more involvement of steganography. Three issues (or questions) are crucial in steganography; Imperceptibility (or how well the hidden secret is embedded?), Robustness (or how immune the embedded secret data against tempering?) and Payload (or what is the storage capacity of the carrier media?). The main concern of any embedding technique is to enhance all are most of these issues.

Digital data hiding methods can be achieved either in a spatial domain or frequency domain. In spatial domain methods, the secret message is embedded into the pixels of the carrier contents, either by alteration or replacement. They achieve good imperceptibility and full capacity can be achieved comparatively faster than in frequency domain, but with poor robustness. On the other side, in frequency domain methods, the secret message is embedded into the feature of the carrier media after certain transformation, however, they offer high robustness at the price of complexity and processing time. An example of spatial domain is the Least Significant Bit (LSB) technique (Raphael and Sundaram, 2011) and examples of the frequency domain algorithms are Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT) and the Discrete Fourier Transform (DFT), (Fridrich and Kodovsky, 2012; Singh and Singh, 2014).

This paper proposes an image steganography technique based on the LSB technique, but suggests a new method for embedding the secret message by generating a highly secure secret key constructed from any available digital file, while the embedding process is achieved with the aid of seven segment pattern distribution. The aims are to produce high imperceptibility and full capacity embedding of secret messages into images. A brief introduction is given in section 1,

followed by the related work in section 2. Then the suggested hiding technique is listed in section 3, its implementation and result discussion in section 4 and finally concluded in section 5.

Related Work

So many works have been published and developed in the field of image steganography, however, few of the most related digital image steganography algorithms for digital data hiding are listed in the following.

Wu and Tsai (2003) proposed a method for hiding a secret message into a gray-valued cover image by partitioning the cover image into non-overlapping blocks of two consecutive pixels and calculating the difference value, then replacing them by a new value to embed the value of the secret message. This method produces a more imperceptible result than those obtained from simple least-significant-bit substitution methods.

Chan and Chen (2004) proposed an Optimal Pixel Adjustment Procedure (OPAP) claiming a reduction in the distortion caused by the LSB substitution method. They introduced a process of adjustment to the pixel values after the secret data embedding which improve the quality of the stego-image without disturbing the data hidden.

EL-Emam (2007). Reported a method of hiding large amount of multimedia data with high security into color BMP image. It is achieved using adaptive image filtering and adaptive image segmentation with bits replacement on the appropriate pixels

Lin *et al*. (2009) suggested a hiding algorithm distortion tolerance using time domain for hiding data and gives better quality of a processed image, producing effective results than other schemes in terms of distortion tolerance.

Three steganography tools were proposed by Hossain *et al*. (2009). They utilize neighborhood information for calculating the data to be embedded in a cover image pixel without causing a noticeable change. The smooth and complicated areas of the carrier image are utilized to embed different amounts of the secret data. Therefore, the psycho visual repetition concept is implemented, since few changes can be tolerated in smooth areas as compared with complicated areas. However, only gray scale were used as carrier images. Also Al-Husainy (2009) proposed an image steganography algorithm using mapping pixels technique to letters.

Optimized True-Color Image Processing algorithm was developed by Al-Dwairi *et al*. (2010) based on the direct and inverse image alteration. They claim that the inversion time was reduced by three and eight times for images and their improvement is achieved using R'G'I design instead of HSI design. Another reversible data hiding method into gray images is proposed by Li *et al*. (2010). It adjacent pixel difference, employing the histogram of the pixel difference sequence to increase the embedding capacity. Also, around the same time, Bamatraf *et al*. (2010) reported a grayscale image for data embedding utilizing the third and the fourth LSBs for embedding and claiming more robustness achievement than the traditional LSB technique.

An embedding scheme based on intensity analysis of pixel intensity into segmented color images is proposed by Ali and Khamis (2012), relying on carrier image histogram analysis. It is claimed to be secure and robust against various types of attacks. Another image steganography algorithm that hide secret data into the sharp areas of color images was developed by Ioannidou *et al*. (2012) in the same year. It was not suitable for images with smooth edges only.

Image segmentation technique and adaptive neural networks with genetic algorithm was developed by El-Emam and Al-Zubidy (2013). Four security levels were utilized to hide the secret data in this technique which resulted into atractive embedding capacity, however, this was on the price of execution speed.

Mixing Discrete Cosine Transform with LSB in a hybrid system by Sruthi *et al*. (2014) achieved more robustness but at the price of processing complexity and long execution time. Also in this year, Gandharba Swain (2014) reported an imperceptible and high payload capacity that is based on nine-pixel differencing with modified LSB substitution. Propose variable number of bits are embedded into different blocks categorized into four levels (lower, lower-middle, higher-middle and higher) based on average of pixel value differences in nine-pixel blocks.

Ghosh *et al*. (2015) developed an Extended Hamming Code (EHC) using dual purpose spatial domain algorithm coupling steganography and cryptography. They achieved good imperceptibility and robustness. Also Al-Shatanawi and El-Emam (2015) reported an image steganography algorithm based on irregular and random segment sizes for embedding. They referred to their algorithm as "Modified Least Significant Bits (MLSB)". They claim high imperceptible with high payload capacity reaching four bits per byte.

Kumar and Dutta [(2016) reported an image steganography algorithm that couples information theory concept with LSB algorithm using maximum entropy concept for the embedding process. Such design resulted into some perceptibility and robustness improvement.

This paper reports an image steganography algorithm that couples seven-segment pattern with LSB technique and also use the pattern for the secret key generation and embedding procedure of secret data into still images. Any multimedia file can be used as the secret key and the secret message data is irregularly embedded into all the pixels of the carrier image. The choice of the segment length is also used as part of the key. The details of the proposed steganography algorithm are listed in the following section.

Materials and Methods

This work presents a secret-key image steganography technique that is using the full capacity of the carrier image. This is done through the use of the seven segment display pattern, such as that shown in Fig. 1, as a key for achieving a randomness in the selection of bytes for embedding the secret message bits.

The secret key used in this technique consists of two parts that are chosen by the user:

Part 1: Any multimedia file, such as text, image, audio and video, will be used to generate a sequence of hexadecimal digits from the bytes in it Fig. 2.

Part 2: A segment length that represents the number of bytes contained in each of the seven segments. This part determines the size of the seven segment display pattern that is used to represent each hexadecimal digit in Part 1.

The representation of the hexadecimal digits as a seven segment display pattern is used by the technique to create a random sequence of the carrier images bytes to be used for embedding the secret message bits in the Least Significant Bits (LSB) of them.

In the following, the important terms used in the proposed technique are briefly defined first, then the algorithms used for embedding and extraction processes are described.

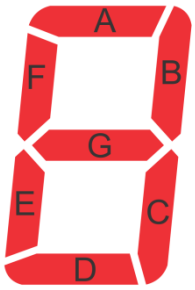


Fig. 1: Seven-segment display pattern

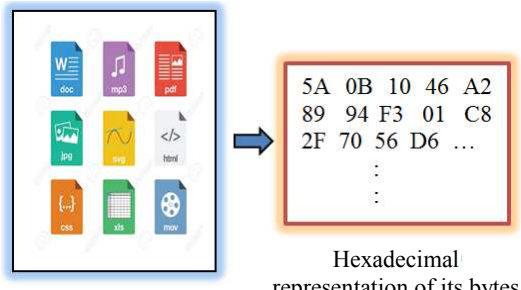


Fig. 2: Examples of Part 1 of the secret key

Terminologies

Carrier Image (*CI*): A 2D colored bitmap image that is used to hide the secret message in its pixels.

Carrier Image Size (*CIS*): The total number of bytes that represent the pixels of the carrier image ***CI,*** calculated by Equation 1:

$$CIS = Width \times Height \times Palette \quad (1)$$

Where, *Palette* equals 3 for the three colors: Red, Green and Blue for color images and equals 1 for gray images. The colored images used in this work represent each of the three colors of the pixel as a byte. This means that *CIS* equal the total number of bytes that represent the image pixels. In this work, *CI* is treated as a collection of bytes.

Segment Length (*SL*): A positive integer number > 1, represents the length of the segment (in byte) in the seven-segment display pattern.

Seven-Segment Display Pattern (***SSDP***): A 2D matrix of size $(((SL\times2)+3) \times (SL+2))$. Where $((SL\times2)+3)$ is the number of rows and $(SL+2)$ is the number of columns of *SSP*, as shown in Fig. 3.

Secret Key (*SK*): A secret key agreed upon or determined by the communicating users. It consists of the following two parts:

Part 1: Any type of digital files *DF* (such as text, image, audio and video, etc.). The steganography technique in this work treats the file as a collection of bytes and each byte represents two hexadecimal digits

Part 2: An integer number used as a segment length *SL*

Secret Message (SM)

Any type of digital files (such as text, image, audio, or video, etc.) that contains a collection of bytes. This file represents the secret information that is to be hidden into the LSBs of the pixels in the carrier image ***CI*** based on the secret key ***SK***.

Secret Message Size (SMS)

The total number of bytes in ***SM***. It should be ≤ (***CIS*/8**).

Stego Image (SI)

The 2D colored bitmap image that represents the carrier image after the embedding the secret message ***SM.*** The size of ***SI*** is the same size as ***CI***.

Embedding Phase

To embed the bytes of the secret message ***SM*** in the LSB of the bytes in the carrier image ***CI*** using the secret key ***SK***, the following steps should be applied.

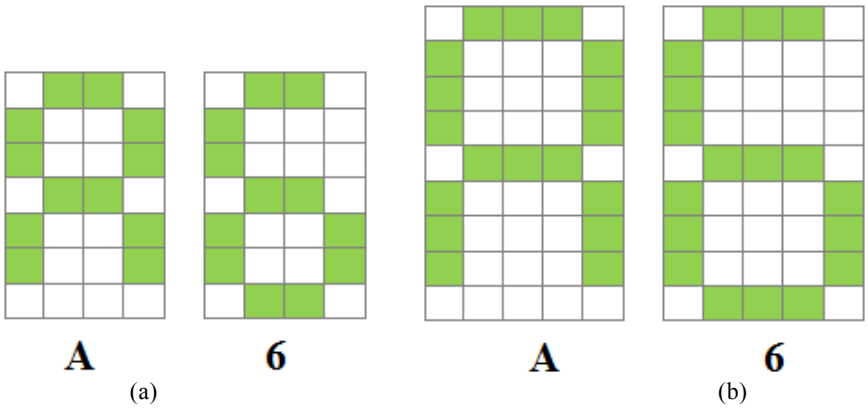


Fig. 3: Examples of Part 2 of the secret key having (a) segment length = 2 and (b) segment length = 3

Step 1: Rearrangement the bytes in the secret message ***SM*** randomly using a predetermined algorithm. This operation adds additional difficulties against the attackers

Step 2: Convert all the bytes in the secret message ***SM*** to the binary representation (where 1byte=8bits) and store all the bits in a one-dimensional list SM_{bits}. For example, if the contents of ***SM*** is: 125 24 210 …, then SM_{bits} will be: 01111101 00011000 11010010 …

Step 3: Calculate the total number of bits ***TB*** in the secret message SM_{bits}, by using equation 2

$$TB = SMS \times 8 \quad (2)$$

Step 4: Read Part 1 ***DF*** of the secret key ***SK*** and create a list ***SKL*** of hexadecimal digits from the bytes of ***DF***. The length of the list ***SKL*** is calculated by equation 3.

$$Length(SKL) = Length(DF) \times 2 \quad (3)$$

Each byte in ***DF*** is split into a pair of hexadecimal digits and then store the digits in ***SKL***. This pair represents the Most Significant Digit (MSD) and the Least Significant Digit (LSD) respectively, hence they are extracted using Equations 4 and 5:

$$MSD = DF_i \; div 16 \quad (4)$$

$$LSD = DF_i \bmod 16 \quad (5)$$

Where, DF_i is the i^{th} byte in ***DF***. For example, if DF_i = **5A**, then ***MSD*** = **5** and ***LSD*** = **A**. The ***SKL*** for the key file of Fig. 2 will be:

5	A	0	B	1		0	4	6	A	2	8	9	9	4	F	3	0	1	…

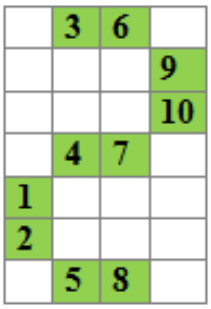


Fig. 4: Flow diagram of the embedding algorithm process

Step 5: Read Part 2 ***SL*** of the secret key ***SK***, create a table to represent the Number of Segments Used (NSU) and the total Number of Bytes Used (NBU) for each hexadecimal digit when it is represented as seven-segment display pattern based on the segment length ***SL***, as shown in Table I.

The actual number of bytes used in each hexadecimal digit is calculated using Equation 6:

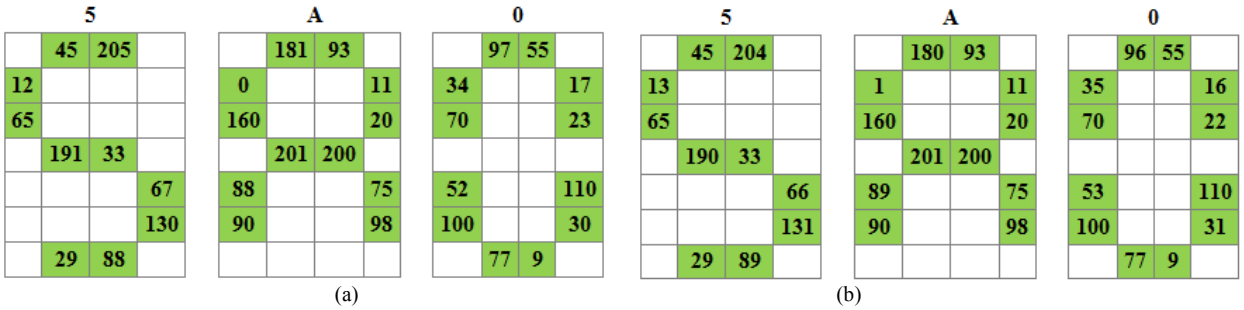
$$NBU = NSU \times SL \quad (6)$$

Step 6: The bits of the secret message SM_{bits} are embedded in the Least Significant Bit (LSB) of the bytes of ***CI*** using the following operations:

- If there are more bits in SM_{bits}, read a set of bytes from ***CI*** that are enough to fill the cells of the desired segments (A, B, C, D, F and G) and fill the cells (column by column) in the seven-segment display pattern ***SSDP*** based on the hexadecimal digits in ***SKL***
- Scan the seven-segment display pattern ***SSDP*** (row by row) and embed the required bits of SM_{bits} in the LSB of the bytes in the cells of the segments within ***SSDP*** according to the sequence of each cell visited (Fig. 4)

Table 1: *NSU* and *NBU* values of the hexadecimal digits when they represent as a seven-segment (for ***SL*** = 4)

Hexadecimal Digit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
NSU	6	2	5	5	4	5	6	3	7	6	6	7	4	6	5	4
NBU	24	8	20	20	16	20	26	12	28	24	24	28	16	24	20	16



(a) (b)

Fig. 5: (a) The values of bytes in ***SSDP*** after embedding some bits of SM_{bits} (b) An example of embedding process (***SL*** = 2)

(a) (b)

(c)

Fig. 6: Samples of carrier images used in the experiments (a) Edifice 236×157×3 (b) Fishes 256×192×3, (c) Ship 256×161×3

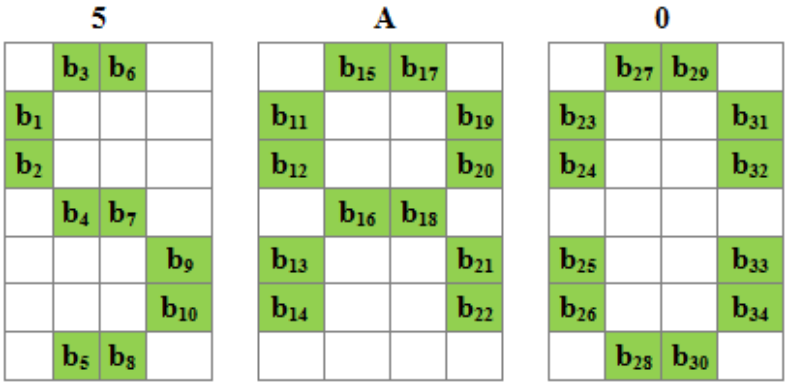


Fig. 7: Byte sequence representation of hexadecimal digits

Table 2: Sample results for PSNR values and embedding processing time for the proposed algorithm

Image Name	Type of ***DF***	***SL***	PSNR (dB)	Embedding Time (sec)
Edifice	Text	20	51.17	0.16
Fish	Image	100	51.18	0.21
Ship	Audio	8	51.14	0.19

Table 3: PSNR values and embedding processing time for the traditional LSB algorithm

Image Name	PSNR (dB)	Embedding Time (sec)
Edifice	51.13	0.13
Fish	51.15	0.15
Ship	51.43	0.14

To embed one Bit of SM_{bits} in the LSB of a Byte in ***SSDP***, one of the following four cases may occur:

- If the Bit = 0 AND the $Byte_{Current}$ is even, then $Byte_{New}$ = $Byte_{Current}$
- If the Bit = 0 AND the $Byte_{Current}$ is odd, then $Byte_{New}$ = $Byte_{Current}$ - 1
- If the Bit = 1 AND the $Byte_{Current}$ is even, then $Byte_{New}$ = $Byte_{Current}$ +1
- If the Bit = 1 AND the $Byte_{Current}$ is odd, then $Byte_{New}$ = $Byte_{Current}$

Figure 5 shows an example (in numbers) of the embedding process that is performed in this step.

Step 7: After step 6 in the embedding phase, the resulted carrier image represents the stego-image ***SI***:

SKL: **5 A 0 B ……**
SM_{bits}: **10110101110111001011000110001001111 …**
SSDP of ***SKL*** that contains bytes of ***CI*** before the embedding process

Extraction Phase

To extract the secret message bits SM_{bits} from the bytes of pixels in the stego-image ***SI***, using the secret key ***SK***, the following steps are applied

Step 1: Perform the same processes that were done in step 4 and step 5 of the embedding phase.

Step 2: The secret message bits SM_{bits} that are hidden in the LSB of the bytes of pixels based on the seven-segment display pattern of the hexadecimal digits in the ***SKL*** are extracted using the following operations:

- If there are more bytes in the ***SI***, read a set of bytes from ***SI*** that are enough to fill the cells of the desired segments (A, B, C, D, F and G) sequentially in the seven-segment display pattern ***SSDP*** based on the hexadecimal digits in ***SKL***
- Scan the seven-segment display pattern ***SSDP*** (row by row) and extract the bits in the LSB of the bytes in the cells of the segments within ***SSDP*** according to the sequence of each cell visited (Fig. 5). Any Bit from the LSB of a Byte in ***SSDP*** is extracted depending on one of the following:
  - If the Byte is even, then the Bit = 0
  - If the Byte is odd, then the Bit = 1

Step 3: Rearrange the SM_{bits} of the secret message as a one-dimensional array of bytes by converting each 8-bits to its corresponding byte, ***SM***. For example, if M_{bits} 01111101 00011000 11010010 …, then the corresponding ***M*** will be 125 24 210 ….

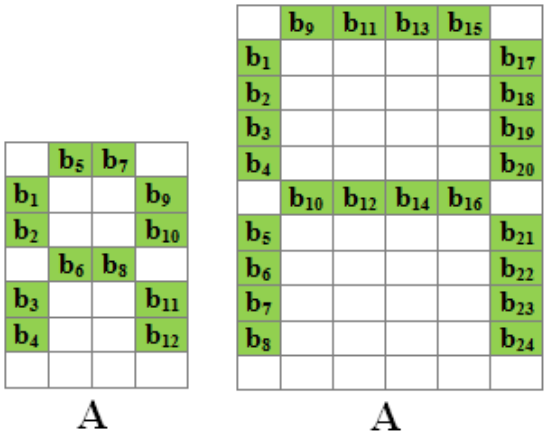
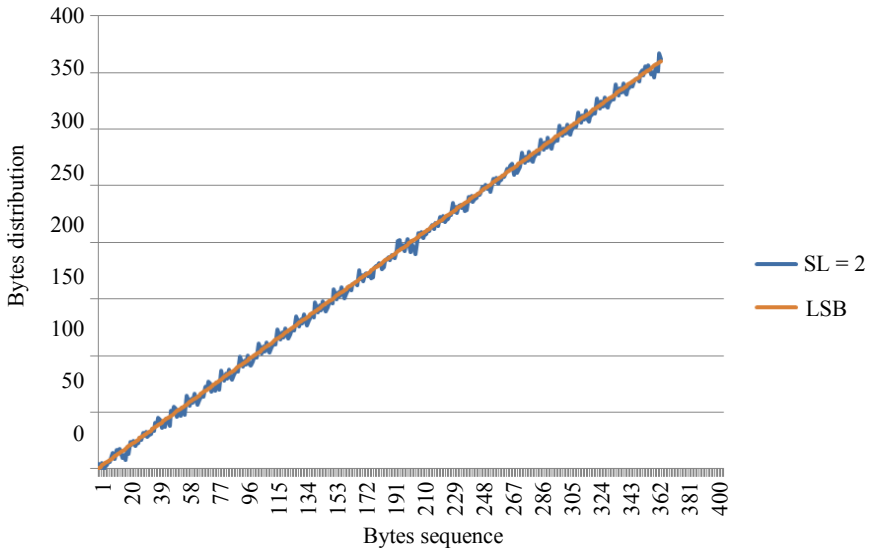
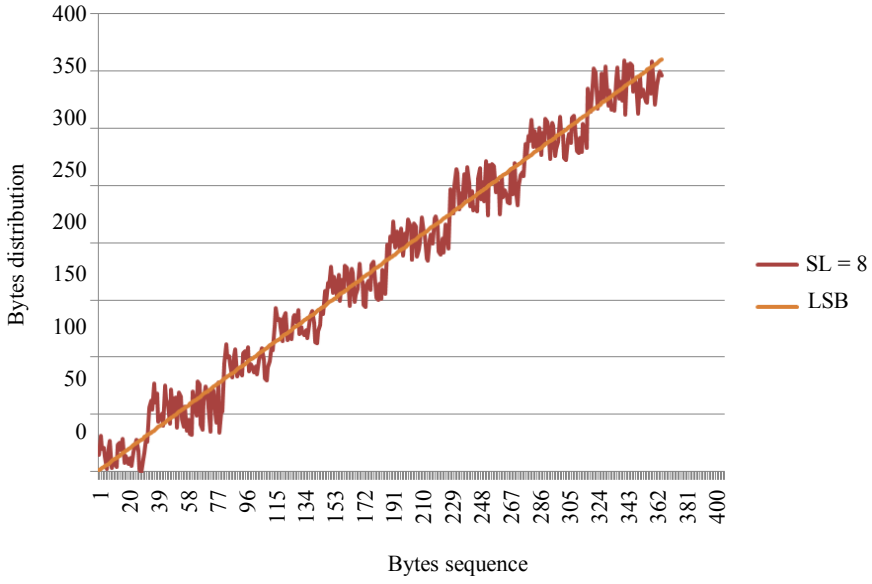


Fig. 8: A Numerical example of the random sequence of bytes based on the seven-segment display pattern used (a) The sequence of the bytes (***SL*** = 2): $b_5\ b_7\ b_1\ b_9\ b_2\ b_{10}\ b_6\ b_8\ b_3\ b_{11}\ b_4\ b_{12}$ (b) The sequence of the bytes (***SL*** = 4): $b_9\ b_{11}\ b_{13}\ b_{15}\ b_1\ b_{17}\ b_2\ b_{18}\ b_3\ b_{19}\ b_4\ b_{20}\ b_{10}\ b_{12}\ b_{14}\ b_{16}\ b_5\ b_{21}\ b_6\ b_{22}\ b_7\ b_{23}\ b_8\ b_{24}$





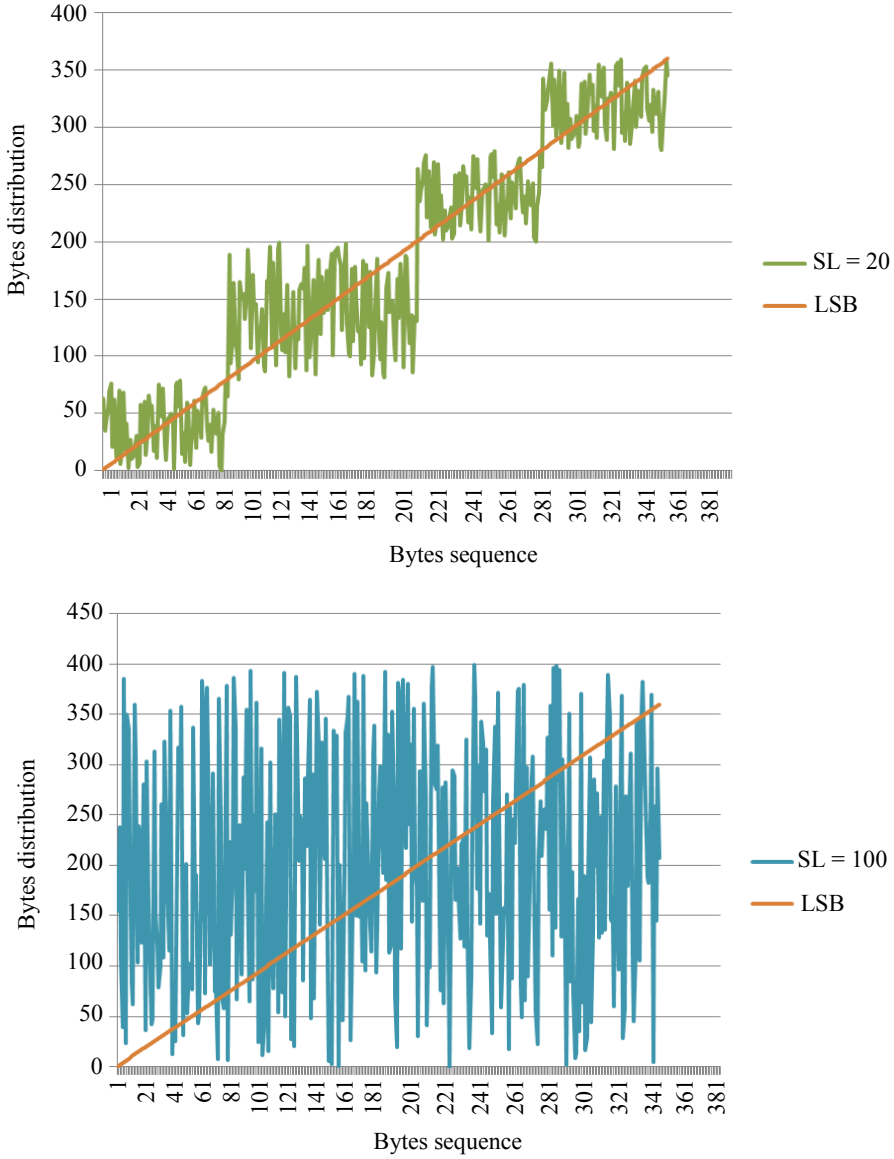


Fig. 9: The effect of using different values of SL on the random sequence of bytes when it's applied on "Ship" image

Step 4: Rearrange the bytes of the secret message ***SM*** to obtain the original sequence based on the adopted algorithm.

Results and Discussion

For the performance evaluation of the proposed algorithm, hundreds of images in various sizes, quality, scenes' and color combination contents has been experimented with for secret data hiding using. This section summarizes and discusses the obtained results. Three images with different features are selected here as samples to illustrate the performance of the technique. They are an edifice (236×157) pixels, fishes (256×192) pixels and a ship (256×161) pixels, as shown in Fig. 6.

A set of measurements have been used in the experiments to evaluate the performance of the proposed technique and compare it with the traditional LSB technique. Table 2 summarizes some sample results. It lists the computed Peak Signal to Noise Ratio (PSNR) and the embedding process time for the some selected carrier image samples. These implementations were done for different types of Data File (DF) and Segment Length (SL). Equations 7 and 8 are used to calculate the PSNR values in the experiments (Al-Husainy, 2012; 2016). It should be mentioned here that the recorded values of the embedding execution time are approximate times determined by the program based on the internal processes in the used PC.

$$NMAE = \frac{\sum_{k=0}^{(Width \times Height \times Palette)-1} |I(k) - S(k)|}{width \times Height \times Palette} \times 100 \quad (7)$$

Where, I and S are the carrier image and stego-images, respectively:

$$PSNR = 10.\log_{10}\left(\frac{MAX_I^2}{NMAE}\right) \quad (8)$$

Where, MAX represent the maximum decimal value of stego-image intensity.

To compare the results of the proposed technique listed in Table 2, with the traditional LSB image steganography technique, the traditional LSB technique has been applied to the same images (using the same PC) and the results are summarized in Table 3.

Moreover, to clarify the effect of using the seven-segment display patterns in order to achieve the random use of the pixel bytes in the carrier image for embedding the secret message bits, consider the following example.

If the list SKL for the hexadecimal digits representation shown in Fig. 5 (i.e., 5A 0) is expressed as a seven-sequence display pattern as illustrated in Fig. 7, then the original sequence of the bytes in the CI can be written as:

b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10} b_{11} b_{12} b_{13} b_{14} b_{15} b_{16} b_{17} b_{18} b_{19} b_{20} b_{21} b_{22} b_{23} b_{24} b_{25} b_{26} b_{27} b_{28} b_{29} b_{30} b_{31} b_{32} b_{33} b_{34} ……

Moreover, in the proposed technique, different segment length in the seven-segment display pattern can be used to embed the secret message bits in the bytes of the image. Figure 8 shows the effect of using a different segment length on the random order of the bytes used in the embedding process.

Then, the byte sequence of CI that is used in the embedding process is:

b_3 b_6 b_1 b_2 b_4 b_7 b_9 b_{10} b_5 b_8 b_{15} b_{17} b_{11} b_{19} b_{12} b_{20} b_{16} b_{18} b_{13} b_{21} b_{14} b_{22} b_{27} b_{29} b_{23} b_{31} b_{24} b_{32} b_{25} b_{33} b_{26} b34 b28 b30 ……

It is clear, in Fig. 8, that the selected value of ***SL*** makes a major effect on the generation of the random sequence of bytes used to embed the secret message bits $\boldsymbol{SM}_{\mathbf{bits}}$. In addition to the ***SL*** effect, the nature of the byte values contained in the secret key and their representation as hexadecimal digits adds more randomness in the sequence of bytes used.

After analyzing the performance of the proposed steganography technique, some important points can be drawn in the following:

1. The technique achieved a random selection of the bytes of pixels in the carrier images used to hide the secret message bits. This is satisfied through the implementation of randomness in three stages:
   a) Random rearrangement of the bytes sequence of the secret message SM before the embedding process
   b) The use of random patterns in the seven-segment display depending on the hexadecimal digits in DF of the secret key SK
   c) The use of different segment length SL of the seven-segment display pattern
2. Most of the existing random steganography techniques do not use the full capacity of the carrier image, while the proposed technique uses the full capacity of the carrier image
3. The distortion level occurs in the stego-image is acceptable, as compared with the traditional LSB technique
4. The time needed for the embedding process execution is practically acceptable as compared with the traditional LSB technique
5. The proposed technique uses composite secret key ***SK*** that is involving two parts ***DF*** and ***SL***. This provides a high level of security to the hidden secret message against attackers

Limitation and Future Works

Although the proposed steganography algorithm achieved acceptable results in PSNR compared with the traditional LSB algorithm and certainly improves the protection level for the hidden message in the stego-image, it can be note that the time required to complete the embedding process has been increased when the segment length SL increased in the algorithm. This may slows the algorithm somehow, however, there is a good opportunity to use the proposed algorithm in many information security applications especially when the time doesn't play the major factor in these fields .

The future trend for the authors is to implement the algorithm in other multimedia files such as video and audio. Also, a more selective strategy might be implemented to decrease the distortion in the stego-image and achieve high protection for the hidden message.

Conclusion

The proposed steganography technique has proven its ability for implementing any multimedia file to be used as the origin for the secret key generation for the steganography system used for embedding secret messages into color carrier images. The key generation implemented the seven-segment patterns representation of the hexadecimal codes of the multimedia file.

The random selection of pixels leads to the addition of huge difficulties against attackers and provides genuine protection for the hidden information. This technique introduces the flexibility of changing the segment lengths, adding extra difficulty of breaking the secret key. Moreover, it was noticed that as the segment size increases, the PSNR value improves.

Acknowledgment

The authors are grateful to the Middle East University, Amman, Jordan for the financial support granted to cover the publication fee of this research article.

Author's Contributions

Mohammed A. Fadhil Al-Husainy: Organized, designed, programmed and performed the experimentation plus contributed to the writing of the manuscript.

Hamza Abbass Al-Sewadi: Designed the research plan, coordinated the data analysis and contributed to the writing of the manuscripts.

Ethics

Participation during pilot and experiment trials in this research work are voluntary and the participants are made known that their feedbacks will be contributing to a non-profit research project.

References

Al-Dwairi, M. O., A. A. Ziad, A. A. Amjad and A. Z. Rushdi, 2010. Optimized true-color image processing. World Applied Science J., 8: 1175-1182.

Al-Husainy, M.A.F., 2009. Image steganography by mapping pixels to letters. J. Computer Science, 5: 33-38. DOI: 10.3844/jcssp.2009.33.38

Al-Husainy, M.A.F., 2012. Message segmentation to enhance the security of LSB image steganography. International J. Adv. Computer Science Applications.

Al-Husainy, M.A.F., 2016. A novel image encryption algorithm based on the extracted map of overlapping paths from the secret key. RAIRO-Theoretical Inform. Applications, 50: 241-249. DOI: 10.1051/ita/2016023

Ali, H.A. and S.A.K. Khamis, 2012. Multi image watermarking scheme based on intensity analysis. International J. Res. Rev. Inform. Science, 2: 201-206.

Al-Shatanawi, O.M. and N.N. El Emam, 2015. A new image steganography algorithm based on mlsb method with random pixels selection. International J. Network Security Applications, 7: 37-37.

Bamatraf, A., R. Ibrahim and M.N.B.M. Salleh, 2010. Digital watermarking algorithm using LSB. Proceedings of the International Conference on Computer Applications and Industrial Electronics, Dec. 5-8, IEEE Xplore Press, Kuala Lumpur, Malaysia, pp: 155-159. DOI: 10.1109/ICCAIE.2010.5735066

Chan, C.K. and L.M. Cheng, 2004. Hiding data in images by simple LSB substitution. Patten Recognition, 37: 469-474. DOI: 10.1016/j.patcog.2003.08.007

El-Emam, N.N. and R.A.S. Al-Zubidy, 2013. New steganography algorithm to conceal a large amount of secret message using hybrid adaptive neural networks with modified adaptive genetic algorithm. J. Syst. Software, 86: 1465-1481. DOI: 10.1016/j.jss.2012.12.006

EL-Emam, N.N., 2007. Hiding a large amount of data with high security using steganography algorithm. J. Computer Science, 3: 223-232.

Fridrich, J. and J. Kodovsky, 2012. Rich models for steganalysis of digital images. IEEE Trans. Information Forensic Security, 7: 868-882. DOI: 10.1109/TIFS.2012.2190402

Ghosh, S., D. Sayandip, P.M. Santi and R. Hafizur, 2015. A novel dual purpose spatial domain algorithm for digital image watermarking and cryptography using Extended Hamming Code. Proceedings of the 2nd International Conference on Electrical Information and Communication Technology, Dec. 10-12, IEEE Xplore Press, Khulna, Bangladesh, pp: 167-172. DOI: 10.1109/EICT.2015.7391940

Hossain, M., S. Al Haque and F. Sharmin, 2009. Variable rate steganography in gray scale digital images using neighborhood pixel information. Proceedings of the 12th International Conference on Computers and Information Technology, Dec. 21-23, IEEE Xplore Press, Dhaka, Bangladesh, pp: 267-272. DOI: 10.1109/ICCIT.2009.5407128

Ingemar, J.C., L.M. Matthew, A.B. Jeffrey, F. Jessica, and K. Ton, 2008. Digital Watermarking and Steganography. 2nd Ed, the Morgan Kaufmann Series in Multimedia Information and Systems, ISBN: 978-0-12-372585-1

Ioannidou, A., S.T. Halkidis and G. Stephanides, 2012. A novel technique for image steganography based on a high payload method and edge detection. Expert System Applications, 39: 11517-11524. DOI: 10.1016/j.eswa.2012.02.106

Kumar, S. and A. Dutta, 2016. A novel spatial domain technique for digital image watermarking using block entropy. Proceedings of the International Conference on Recent Trends in Information Technology, Apr. 8-9, IEEE Xplore Press, Chennai, India, pp: 1-4. DOI: 10.1109/ICRTIT.2016.7569530

Li, Y.C., C.M. Yeh and C.C. Chang, 2010. Data hiding based on the similarity between neighboring pixels with reversibility. Digital Signal Processing, 20: 1116-1128. DOI: 10.1016/j.dsp.2009.10.025

Lin, I.C., Y.B. Lin and C.M. Wang, 2009. Hiding data in spatial domain images with distortion tolerance. Computer Standards Interfaces, 31: 458-464. DOI: 10.1016/j.csi.2008.05.010

Raphael, A.J. and V. Sundaram, 2011. Cryptography and Steganography-A Survey. International J. Computer Technology Application, 2: 626-630.

Schneier, B., 1996. Applied cryptography: protocols, algorithms and source code in C. New York: Wiley.

Singh, A. and S.J. Singh, 2014. An Overview of Image Steganography Techniques. International J. Engineering Computer Science, 3: 7341-7345.

Sruthi, N., A.V. Sheetal and V. Elamaran, 2014. Spatial and spectral digital watermarking with robustness evaluation. Proceedings of the International Conference on Computation of Power, Energy, Information and Communication, Oct. 02, IEEE Xplore Press, Chennai, India, pp: 500-505. DOI: 10.1109/ICCPEIC.2014.6915415

Stalling, W., Cryptography and Network Security Principles and Practices. 7th Ed., Prentice Hall, ISBN-13: 978-0134444284.

Swain, G., 2014. Digital Image Steganography using Nine-Pixel Differencing and Modified LSB Substitution. Indian J. Science Technology, 7: 1444-1450.

Wu, D.C. and W.H. Tsai. 2003. A steganographic method for images by pixel-value differencing. Pattern Recognition Letters, 24: 1613-1626. DOI: 10.1016/S0167-8655(02)00402-6