VOL. 14, NO. 12, JUNE 2019 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

IMAGE STEGANOGRAPHY TECHNIQUE USING MULTILEVEL HASH TABLE

Mohammed A. Fadhil Al-Husainy[1] and Hamza A. A. Al-Sewadi[2]
Faculty of Information Technology, Middle East University, Jordan
E-Mail: dralhusainy@gmail.com

ABSTRACT

The use of steganography as an alternative for cryptography in the field of data security is growing. One of the main issues in steganography is the search for a strong secret key and an efficient embedding algorithm hiding confidential messages into the chosen carrier multimedia. This paper presents a new method for generating the secret embedding key employing the traditional way of representing the dates. Hash tables were generated in three levels using the date components, and are utilized for embedding the secret message into carrier images using Least Significant Bit (LSB) hiding process. The three level hashing tables would result in increased difficulty for attackers providing data security strength. Obtained results of the conducted tests manifested even distribution of the histogram and comparable Peak Signal to Noise Ratio (PSNR) with a traditional LSB scheme.

Keywords: hash functions, image steganography, key generation, data hiding.

1. INTRODUCTION

Advances in information transfer, computation power, and storage capacity of handy and popular equipment, such as smart phones have made personal and sensitive information during storage and transfer vulnerable to the danger of various hazards such as leakage, theft, modification, denial, etc. [1]. Consequently, vast amount of work and schemes were developed as countermeasures to stop or fight these hazards, such as cryptosystems, steganography, and mobile forensics [2]. Cryptosystems, work on making the information unintelligible, steganography hides the information from intruders, while forensics are concern with tracing, collecting and analyzing digital evidence to resolve mobility issues related to information flow and storage for the purpose of crime or misuse detection [3]. Both cryptography and steganography aim at the protection of confidential data against third parties such as adversaries, hackers, interceptors, intruders, interlopers, eavesdroppers, opponents, and enemies [4]. Only data hiding will be considered in this paper, hence more elaboration on steganography is relevant and will be explained in more details. The most important features of steganography algorithms are imperceptibility, payload, and robustness.

So many algorithms were developed, published, and practically applied for embedding secret information in various media such as text, audio, image, and video files, however, only embedding in still images will be included here. However, in general data hiding algorithms can be classified into two categories; spatial or time domain and transformation or frequency domain. The former is faster and simpler, but fragile, while the latter is complicated and slower, but provide robustness. In spatial domain algorithms, hiding is achieved by replacing some bits of the carrier multimedia by bits of the data to be hidden. An example is the Least Significant Bit (LSB) technique, where the replacement is done in the least significant bits of the image pixels. However, in the transformed domain algorithms, hiding is performed in a transformed version of the images. There are many examples on transformation domain algorithms such as Discrete Wavelet Transform (DWT) [5], Discrete Fourier Transform (DFT) [6], and Discrete Cosine Transform (DCT) [7].

Moreover, one of the major issues in steganography is the embedding key, hence so many algorithms were suggested with various key designs, however, the search for efficient and trusted secret key is still of great interest. The proposed steganography technique in this paper suggests a new method for generating the secret embedding key to be used for hiding secret information into still images. After the brief introduction in section one, related work is listed in section two. Then the proposed steganography technique is outlined in details. Section four includes the implementation, results and discussion of the proposed technique, and finally section five concludes the paper.

2. RELATED WORKS

Over the years, so many algorithms have been developed for data hiding, however, only some of the most related ones to the proposed scheme will be surveyed here, namely Least Significant Bit (LSB) methods in still images. The aim of all techniques is to achieve the best embedding of secret messages. For example, a steganographic technique was described by Hossain *et al.* [8] in 2019, where the quality of data that can be embedded in a gray cover image is calculated without causing a noticeable change by utilizing the feature of the image contents neighbourhoods. Hence, some complications arose due to the need for selecting the proper area for embedding, namely less bits of the secret data are hidden in the smooth areas, while more bits are hidden in the hard ones. Therefore, this embedding technique is based on the psycho visual repetition concept where changes in the smooth areas would be easily noticeable while alteration in the edges is hard to notice.

Embedding and extraction method utilizes direct and inverse image color/gray alteration is reported by Al-Dwairi *et al.* [9] in 2010. They used R'G'I design rather

than HIS claiming inversion time reduction of three and eight times.

Also, Bamatraf *et al*. [10] in 2010, implemented data embedding in the third and the fourth Least Significant Bits (LSB) in grayscale image, claiming improved robustness over traditional LSB methods.

Moreover, Li *et al* [11] in 2010, proposed a reversible embedding algorithm into gray image. They increased the embedding capacity by employing image histogram pixel differences. Also Ali and Khamis [12] in 2012 utilizes pixel intensity histogram analysis and adopting a modulating technique to alter the intensity rather than bit replacement. They claim good security and robustness.

Chutani and Goyal [13] in 2012 proposed a multi-level hiding (MLH) scheme which hide the secret message into pseudo-image first and embed it in the carrier image, using the LSB in both embedding process. They claim satisfactory results with over 60 dB of peak signal to noise ration.

Four security level embedding algorithm was proposed by El-Emam and Al-Zubidy [14] in 2013. It was suitable for embedding large amount of data into a color image. But the use of embedding into four levels has resulted in very slow execution speed. Al-Shatanawi and El-Emam [15] in 2015 modified the earlier algorithm adopting Modified Least Significant Bits (MLSB) to embed randomly into different image segments, claiming high imperceptible and payload.

Another multi-level steganography (MLS) scheme is also reported by Hussein [16] in 2017 implement a message embedding into png and jpeg images. He adopted two levels of embedding using enhanced LSB in the first level and pixel intensity embedding in the second. The imperceptibility was good enough having a PSNR of over 60%, but nothing was said about the robustness.

An algorithm proposed by Kumar and Dutta [17] in 2016, using the concept of maximum entropy in information theory with LSB algorithm, and also claiming both the perceptibility and the robustness.

A Multilevel Image Steganography Using Compression Techniques was proposed by Sayed and Wahby [18] in 2017, proposed a multilevel steganographic scheme implementing enhanced LSB embedding processes using BMP images for both levels and applying lossless data compression technique using Huffman, LZW algorithm and Winrar between First and Second level of steganography

An algorithm was proposed in 2018 by Al-Husainy and Al-Sewadi [19] that implements any multimedia file as secret key for irregularly embedding secret messages into the carrier image using the seven-segment patterns of the hexadecimal representation of the secret key content. It is claimed that it produces full embedding capacity.

This paper presents a novel way of generating the secret embedding key for image steganography. As the date representation is internationally depends on the numbers related to the date, month, and year, the algorithm implements the concept of multiple hash tables using this traditional representation for generating the secret key. Then this key is used to embed any message file into image pixels.

3. THE PROPOSED STEGANOGRAPHY TECHNIQUE

The proposed image steganography technique is outlined in details in this section. This technique utilizes the full date (Day/Month/Year) to generate the secret key that is selected by the user to be used in this scheme in order to build three levels hash tables. These tables are used to generate a random sequence of the carrier image bytes in order to embed the secret message bits into the Least Significant Bits (LSB) of these bytes. The strength of the technique comes from using an easy way for selecting the key by the user and the random sequence generation of the image bytes that is used to embed the secret message bits.

To clarify the idea of using the date in building the three-level hash tables for generating a random sequence of the carrier image bytes, for example, let the date (25/6/1983) is being selected as a secret key and the carrier image contains 3072 bytes indexed from the index (0) to the index (3071).

The first level of the hash table is generated by hashing the index of each byte*ByteIndex*to the hash table of size (1983); it names the Year hash table. The *YearHashIndex*of each *ByteIndex*is calculated using the equation (1).

$$YearHashIndex = ByteIndex \bmod Year \quad (1)$$

After the first hashing operation, the indices of the bytes *ByteIndex* in each hash index of the Year hash table are hashed again to the hash table of size (6), it names the Month hash table. The *MonthHashIndex*of each *ByteIndex*is calculated using the equation (2).

$$MonthHashIndex = ByteIndex \bmod Month \quad (2)$$

Finally, the indices of the bytes *ByteIndex* in each index of the Month hash table are hashed again to the hash table of size (25), it names the Day hash table. The *DayHashIndex*of each *ByteIndex*is calculated using the equation (3).

$$DayHashIndex = ByteIndex \bmod Day \quad (3)$$

For example, if we want to calculate the hash index of the two successive bytes (2764 and 2765) in the carrier image by using the equations 1, 2 and 3, then the hash index of the byte at the index 2764 is:

YearHashIndex = 2764 mod 1983 = 781
MonthHashIndex = 2764 mod 6 = 4
DayHashIndex = 2764 mod 25 = 14

This means that the byte at the index 2764 located at the index 781 in the Year hash table, at the index 4 in the Month hash table and at the index 14 in the Day hash table.

While, the hash index of the byte at the index 2765 is:

$YearHashIndex = 2765 \bmod 1983 = 782$
$MonthHashIndex = 2765 \bmod 6 = 5$
$DayHashIndex = 2765 \bmod 25 = 15$

This means that the byte at the index 2765 located at the index 782 in the Year hash table, at the index 5 in the Month hash table and at the index 15 in the Day hash table.

From the calculated hash indices of the two successive bytes (2764 and 2765), it is clear that the new locations of the bytes (2764 and 2765) become far from each other because many bytes at different indices in the carrier image have been located between the two bytes at the indices (2764 and 2765).

Figure-1 shows a simple diagram that depicts the three hash tables used in the proposed technique.

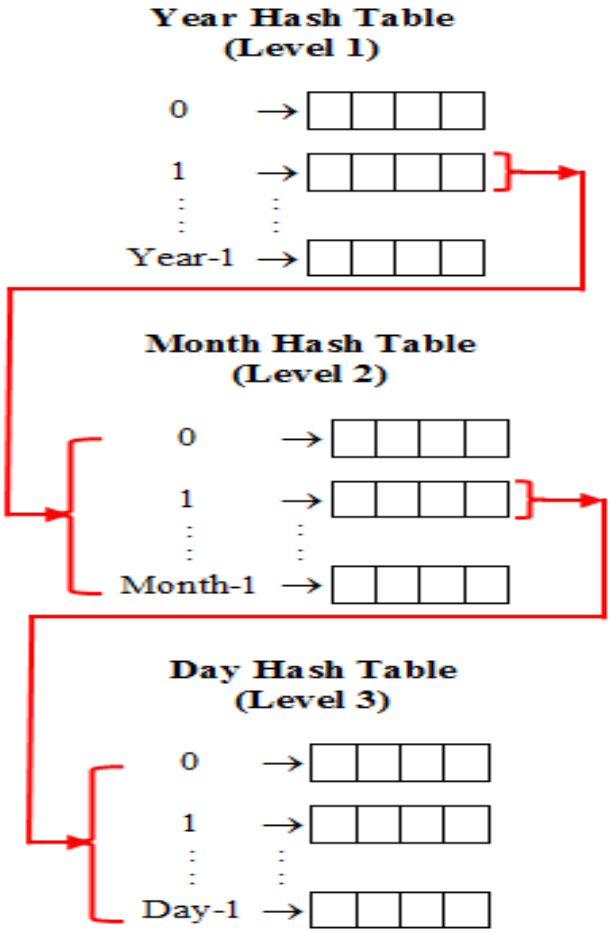


Figure-1. Three levels hash tables used in the proposed Steganography technique.

The two main phases of the proposed steganography technique (embedding and extraction) are explained in this section. The terms used in the proposed technique are defined here.

Carrier Image (*I*): It is a two-dimensional bitmap carrier colored image. The size of the carrier image *I*, called *ISize*, calculated (in pixels) by equation 4.

$$ISize = Width \times Height \times Palette \quad \text{..... (4)}$$

Where *Palette* = 3 for the RGB color images and 1 for gray images.

Secret Key (*K*): It represents a full date selected by the user and it consists of three parts: Day/Month/Year. Where Day=(01…31), Month = (01…12) and Year=(0001…9999).

Secret Message (*M*): It is a digital file that contains a collection of bytes (such as: text, image, audio, or video, etc.). This file represents the secret information which is to be hidden into the carrier image *I* based on the secret key *K*, (the size of $M \leq ((ISize \times 3)/8)$.

Stego Image (*S*): It is the two-dimensional carrier image after embedding the secret message *M* in it. The size of *S* is same as that of *I*.

Embedding process:

The main steps of the algorithm used in the embedding phase to hide the secret message *M* in the carrier image *I* are listed below.

Step 1: Represent the carrier image *I* as one-dimensional list of bytes named ***L***and it is indexed from 0 to ((*ISize*×3)-1).

Step 2: Represent the secret message as a one-dimensional array of bits. For example, if M_{bytes} is: 12524210 …, then M_{bits} will be: 01111101000110001101001O …

Step 3: Read the secret key/Date (i.e. Day/Month/Year) which is selected by the user.

Step 4: Build three hash tables: Day hash table, Month hash table and Year hash table.

Step 5: For each byte in *L*, calculate the three indices: *Year HashIndex*, *Month HashIndex and Day Hash Index* using the equations 1, 2 and 3. Then store the byte in the specific hash index based on the calculated indices.

Step 6: Store the hashed bytes, from the top index to the down index in the Day hash table (Level 3), intoa new list L_{Hahsed} which contains the same bytes of ***L*** but in a random sequence.

Step 7: Embed the bits of the secret message M_{bits} in the Least Significant Bit (LSB) of the bytes in L_{Hahsed}.

Step 8: Store the bytes of the list L_{Hahsed} to generate the stego image ***S***according to the original sequence of the bytes.

To extract the hidden message from the stego image *S*, a reverse operation has been implemented. The steps of the algorithm used in the extraction phase to extract the hidden secret message *M* from the stego image *S* are summarized below.

Step 1: Represent the stego image ***S***as a one-dimensional list of bytes named ***L***and it is indexedfrom 0 to ((*ISize*×3)-1).

Step 2: Read the secret key/Date (Day/Month/Year) from the user.

Step 3: Build three hash tables: Day hash table, Month hash table and Year hash table.

Step 4: For each byte in *L*, calculate the three indices: *YearHashIndex*, *MonthHashIndex and DayHashIndex* using the equations 1, 2 and 3. Then store the byte in the specific hash index based on the calculated indices.

Step 5: Store the hashed bytes, from the top index to the down index in the Day hash table (Level 3), into a new list L_{Hahsed} which contains the same bytes of ***L*** but in a random sequence.

Step 6: Extract the Least Significant Bit (LSB) from the bytes of ***L***and store the extracted string of bits in the secret message M_{bits}.

Step 7: Represent the secret message M_{bits} as bytes to generate M_{bytes}. For example, If M_{bits} is: 01111101000110001101010 …, then M_{bytes} will be: 12524210 …. The generated M_{bytes} represents the output extracted secret message *M*.

4. EXPERIMENTS AND DISCUSSION

To evaluate the performance of the proposed image steganography technique, and explore its advantages and pinpointing any deficiency or drawback, several experiments were conducted using various combinations of different parameters, such as secret encryption key (namely date), secret message, and carrier image. The results are listed and discussed here. Of the hundreds of color images used as carrier images for experimentation, results for the three images shown in Figure-2 are selected as examples. In order to cover wide spectrum of multimedia, these carrier images were carefully chosen in such a way that they have different contents and sizes, namely they are Airplane (256×168×3) bytes, Beach (200×300×3) bytes, and Bird (204×204×3) bytes.

The Peak Signal to Noise Ratio (PSNR) and average embedding and average extraction time for the proposed technique are computed and listed in Tables 1, 2 and 3, respectively. They are listed together with those values computed for least significant bit (LSB) steganography technique for comparison purposes. It must be noted that embedding and extraction execution times were obtained by the program, hence they are approximate time values due to the possible internal processes in the PC which may not be relevant to the intended processes. However, they can be considered useful for comparison purposes.

The PSNR values are calculated using equations 5 and 6[20, 21].

$$NMAE = \frac{\sum_{k=0}^{(Width \times Height \times Palette)} |I(k) \quad S(k)|}{Width \times Height \times Palette} \times 100 \qquad (5)$$

$$PSNR = 10.\log_{10}\left(\frac{MAX_I^2}{NMAE}\right) \qquad (6)$$

Airplane (256×168×3)	**Beach** (200×300×3)	**Bird** (204×204×3)

Figure-2. Carrier images used in the experiments.

Table-1. PSNR of the proposed and traditional LSB steganography techniques.

Image	**PSNR (dB)**	
	Proposed	**Traditional LSB**
Airplane	51.133	51.149
Beach	51.131	51.119
Bird	51.173	51.196

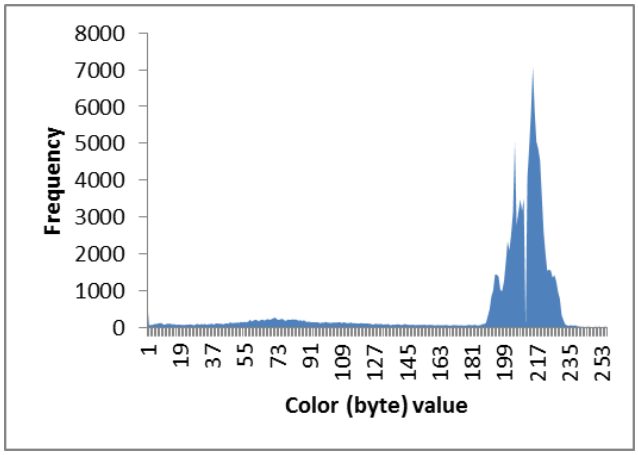
Table-2. Average embedding time measurements.

Image	**Average embedding time (msec)**	
	Proposed	**Traditional LSB**
Airplane	323.31	187.41
Beach	394.10	241.18
Bird	358.26	144.38

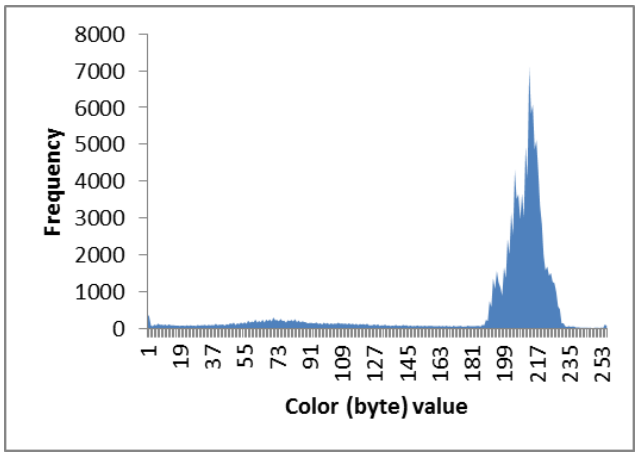
Table-3. Average extraction time measurements.

Image	**Average extraction time (msec)**	
	Proposed	**Traditional LSB**
Airplane	345.86	122.97
Beach	365.22	127.26
Bird	315.58	118.49

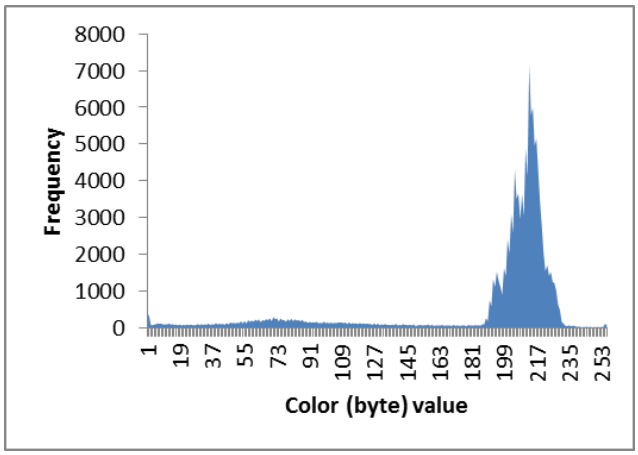
The histogram of the image bytes, before and after embedding the secret message, can be used as visual statistical test to visualize the changes that happen in the bytes of the stego image. Hence, the histograms of the original as well as the stego images for proposed steganography techniques and traditional LSB steganography technique are shown in Figure-3 for the "Airplane" carrier image as an example.



(a) Original Image (Proposed Technique)



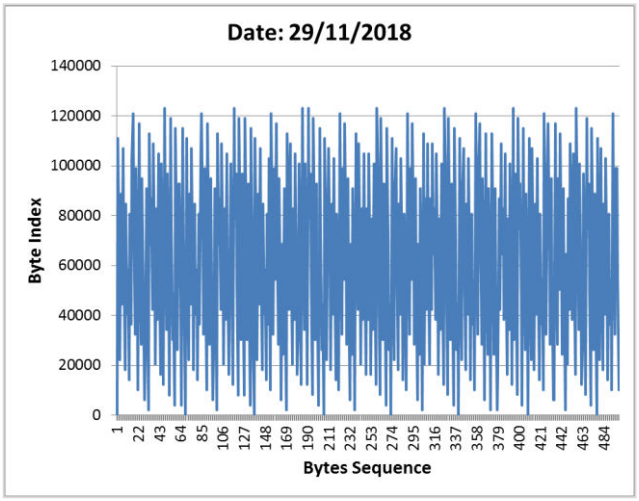
(b) Stego Image (Proposed Technique)

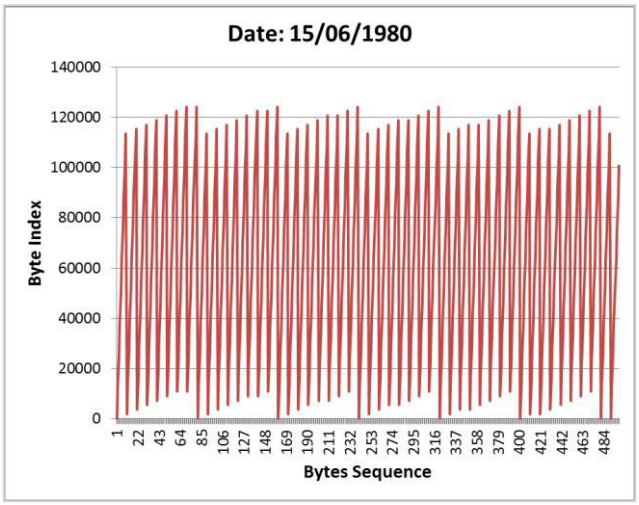


(c) Stego Image (Traditional LSB Technique)

Figure-3. Histogram of the original and stego image for the "Airplane" carrier image.

Furthermore, to investigate the effect of using different secret keys (i.e. dates) in the proposed technique on the random generation of the sequence of the carrier image bites. Many tests have been performed during the experiments; some of these tests on the "Bird" image are shown in figure 4of the first 500 indices in the stego image.





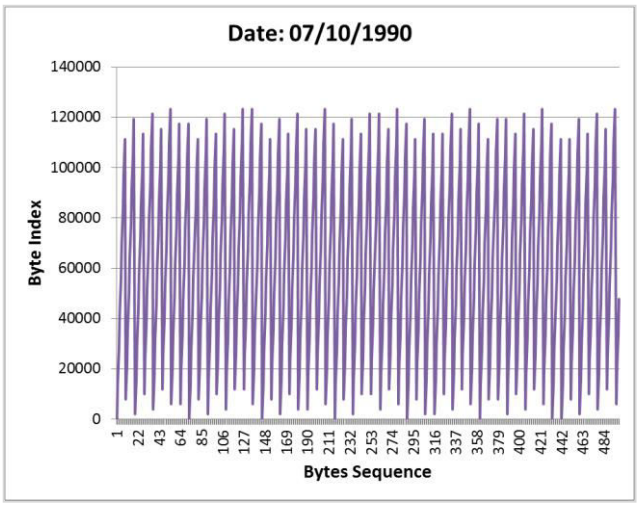


Figure-4. The effect of using different dates as a secret key on the randomness generation of the bytes sequence in the carrier image.

From the obtained implementation results of the proposed technique, the following notes may be stated:

a) The visual comparison of the histograms of Figure-3 indicates no noticeable difference between the original and the stego images, which confirms the potential ability of the proposed technique to hide the embedded content.
b) The technique achieved a random selection of pixels in the used images for hiding the secret messages due to the fact that it is mainly based on the selected date that is used as a secret key in the technique, as illustrated in figure 4.
c) Computed PSNR values for the proposed embedding technique were almost equal to those for the traditional LSB for all implemented carrier images.
d) The average embedding and extraction execution time for the proposed steganography technique was longer than those for the traditional LSB technique. This may be attributed to the added complexity in the proposed method that strengthened the method against prospected attacks, but it needs more investigation to be confirm it.

5. CONCLUSIONS

The proposed multi-level hashing algorithm of the traditional "dates" contents to be used as a secret embedding key for any digital data into color image carriers has proved the ability for using a new technique for random positioning of secret contents. As the dates are written in many different formats, more difficulty is added due to the increase in key space. Obtained results of the conducted tests manifested even distribution of the histogram and comparable PSNR with the traditional LSB scheme. Obtained results have shown an almost equal level of signal to noise ratio as compared with the traditional LSB steganographic technique, although the embedding and execution time is a bit longer, which can be attributed to the added computational complexity. More investigation is underway to improve the computation speed.

REFERENCES

[1] Stalling W. and L. Brown. 2014. Computer Security: Principles and Practice. 3rd Edition, Pearson. ISBN-13: 978-0133773927.

[2] Raphael A. J. and V. Sundaram, 2011. Cryptography and Steganography - A Survey. International Journal of Computer Technology Applications. 2: 626-630.

[3] Sloan T. and J. Hernandez-Castro. 2015. Forensic analysis of video steganography tools. PeerJ Computer Science. 1: e7, DOI.

[4] Stalling W. and L. Brown. 2006. Cryptography and Network Security: Principles and Practice. 4rd Edition, Pearson. ISBN-10: 0-13-187316-4

[5] Banik B. G. and Samir K. Bandyopadhyay. 2013. A DWT Method for Image Steganography. International Journal of Advanced Research in Computer Science and Software Engineering. 3(6): 983-989.

[6] Kaushal A. and Vineeta Chaudhary. 2013. Secured Image Steganography using Different Transform Domain. International Journal of Computer Applications (0975-8887). 77(2): 23-28.

[7] Bansal D. and Rita Chhikara, 2014. An Improved DCT based Steganography Technique. International Journal of Computer Applications (0975-8887) 102(14): 46-49.

[8] M. Hossain, S. Al Haque and F. Sharmin. 2009. Variable rate steganography in gray scale digital images using neighborhood pixel information. in Computers and Information Technology, 2009. ICCIT'09. 12th International Conference on. pp. 267-272.

[9] Al-Dwairi M. O., Z. A. Alqadi, A. A. Abujazar and R. A. Zneit. 2010. Optimized True-Color Image Processing. World Applied Sciences Journal. 8: 1175-1182.

[10] A. Bamatraf, R. Ibrahim and M. N. B. M. Salleh. 2010. Digital watermarking algorithm using LSB. in 2010 International Conference on Computer Applications and Industrial Electronics (ICCAIE) 2010. pp. 155-159.

[11] 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.

[12] Ali H. A. and S. A. K. Khamis. 2012. Multi Image Watermarking Scheme Based on Intensity Analysis. International Journal of Research and Reviews in Information Sciences (IJRRIS). 2: 201-206.

[13] Shaveta Chutani S. and Goyal H., 2012. Image Steganography using Multi Level Hiding Technique. https://www.researchgate.net/publication/277017430_Image_Steganography_using_Multi_Level_Hiding_Technique.

[14] 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. Journal of Systems and Software. 86: 1465-1481.

[15] Al-Shatanawi O. M. and N. N. El Emam. 2015. A New Image Steganography Algorithm Based on Mlsb Method with Random Pixels Selection. International Journal of Network Security & Its Applications. 7: 37.

[16] Hussein A. H. 2015. Multi-LevelImage Steganography by Using Pixel Intensity. Sudan University of Science and Technology College of Graduate Studies College of Computer Science and Information Technology.

[17] Kumar S. and A. Dutta. 2016. A novel spatial domain technique for digital image watermarking using block entropy. in 2016 International Conference on Recent Trends in Information Technology (ICRTIT). pp. 1-4.

[18] Sayed M. H. and Talaat M. Wahby T. M. 2017. Multi-Level Image Steganography Using Compression Techniques. International Journal of Computer Applications Technology and Research. 6(11): 441-450, 2017, ISSN: -2319-8656.

[19] Al-Husainy M. A. F. and Al-Sewadi H. A. 2018. Mohammed A. F. Al-Husainy, Hamza Abbass Al-Sewadi. 2018. Full Capacity Image Steganography Using Seven-Segment Display Pattern as Secret Key. Journal of Computer Science. 14(6): 753-763, DOI: 10.3844/jcssp.2018.753.763.