

NEW IMAGE STEGANOGRAPHY USING PURE AND SECRET-KEY PATTERNS BASED ON DISTANCE IN SPATIAL DOMAIN

R.A. Dihin¹ N.R. Hamza² H.T. Rashid¹

1. Computer Science Department, Faculty of Education for Women, University of Kufa, Najaf, Iraq
rashaa.aljabry@uokufa.edu.iq, hasant.kurmasha@uokufa.edu.iq

2. Computer Science Department, College of Computer Science and Information Technology, University of Al-Qadisiyah, Diwania, Iraq, nesreen.readh@qu.edu.iq

Abstract- Steganography is a technique for concealing secret messages from persons who are not authorized to see them. These days, data security during travel is critical. Using the spatial domain embedding approach, a method with many stages was devised to hide the three secret files (Text, Image, and Audio) inside the RGB images during the investigation of least significant bits LSB4. To make the recovery procedure more difficult and secure, use both pure and secret key patterns based on the distance that depends on the LSB3 of the secret files and also, rotate the cover-image 180 degree before embedding. The files embedding stage then begins within RGB, with text files going into the red channel (R), image files going into the green channel (G), and audio files going into the blue channel (B). This method was successful in hiding files without deforming the original image or causing changes to be noticed as a result of the concealment procedure, and those files were quickly recovered without losing any of their components. The method was evaluated using quality assessment tools such as SSIM, as well as the two popular new techniques HSM and HFM and FEI face dataset.

Keywords: Image Analysis, Steganography, Secret Key, LSB Security, HFM, HSM.

1. INTRODUCTION

Information security refers to a variety of approaches and processes used to keep secret information safe from those who are not allowed to see it [1]. Steganography is a science that was developed to address issues of multimedia information security. It is a data-exchange mechanism in which the secret data is hidden behind a cover (digital media type) [2]. More research initiatives have been proposed at this time to obscure information due to its importance and the development in digital connections transferred via the network. To protect the data from hackers, mechanisms for concealing information have been implemented for obtaining secure data with improved confidentiality and integrity [3]. Information hiding is a subset of steganography; it is the use of clever technology to conceal secret information in media to prevent

unwanted access. Secret contents are hidden in a file (as image) without affecting the carrier file [4].

"Image Steganography", "Video Steganography", "Network Steganography", "Text Steganography", and "Audio Steganography" are the several types of steganography techniques [5]. Due to the increasing expansion of digital multimedia consumption on the internet, digital image security has recently become a major concern. Many researchers have focused their efforts on the topic of digital picture security, as well as developing image encryption approaches to improve security. "Digital watermarks" and "image steganography" are the most important security approaches in digital images [6, 7].

Pure steganography, secret key steganography, and public key steganography are the three types of data embedding patterns. Secret key steganography is a method of concealing information by adding a key to the concealed message (file) while it is hidden inside the medium. As a result, the second party cannot extract the hidden message without knowing the password (secret key), and the concealment procedure is made safer and more sophisticated by including the password [4].

Spatial domain embedding techniques, such as least important bit "LSB" and pixel value difference "PVD" are classified as steganography techniques. Discrete wave transformation "DWT", discrete cosine transformation "DCT", Distortion technologies, diffusion spectrum techniques, coat generation techniques, and filtering and masking techniques are all examples of transform domain embedding [8, 9]. The most popular method is the least important bit injection method "LSB" where in most plots, inclusion data is placed in LSB for each pixel in a cover-image [4]. In the field of information concealment, the least valuable bits occupy a prominent position and are widely used due to their ease of implementation and simplicity [10]. Less Significant Bit embedding is the most widely utilized digital steganography technique. Confidential messages are encrypted by hiding them in the digital signal samples' is Less Significant Bit [11].

2. RELATED WORKS

Researchers have developed a variety of steganography approaches. N.S. Jinan and R.H. Haraa [5] introduced a new technique that is an enhanced technique for the least significant bits technique mixed with the Knight Tour Algorithm. The novel approach was tested using grayscale photos. I.S. Jane, et al. [12], the HLSB technology provided here is an enhanced technology for hiding information by data hiding in an image with reduced variation in image bits, which makes the method more effective and safer, and the authentication module can be used with encryption approaches. This method can be used to hide grayscale images 8 bits within RGB images 24 bits, and it can also be used to hide color images within other color images.

O. Ahmad and A. Mohammed [14], LSB steganography improves the security of messages that are sliced and delivered to the recipient. The goal of this study is to increase the security of video and text by adding text into the video and then mixing the video so that it can't be seen. O. Zahran and K. Mohamed [6] introduced many algorithms based on LSB for embedding secret data in color images (RGB) as LSB_1, LSB_2, LSB_3 and LSB 4. D.M. Stella and A.H. Alexander [13] present the option of hiding text or an image inside the cover-image using LSB. Duffing map has modified the position of the characters in the secret text and the pixels in the secret image (random number generator). The basic concept is to hide the hidden message (color image, gray image, and text) in the cover image's LSB (color or gray image).

PVD techniques and LSB substitution used in picture steganography were compared in performance by S. Sherin [15]. According to LSB substitution, the PVD approach completes the algorithm, making the embedding process more data secure. If a huge amount of data needs to be buried, the PVD approach will be more appropriate. Grover, et al. [16] proposed an edge-based adaptive or adaptable LSB alternative object that covers two bits of information in non-edgy blue channel (B) pixels and three covert bits in edgy pixels. The data is divided into two groups and incorporated in the cover image, starting with the most important pixel and progressing through the image to increase its durability.

3. RESEARCH METHOD

The proposed methodology is explained as to the following three stages.

3.1. Preparation Stage

3.1.1. A-Secret Key

To guarantee the confidentiality of the supplied files, a key is applied to the sensitive files represented by text, image, and audio files before the second stage (embedding stage). The key is added at random based on the message's least important bits. The following steps show how to add the secret key to the text to be hidden (the same steps are applied to the hidden image and audio). The following steps show how to add the secret key to the text to be hidden (the same steps are applied to the hidden image and audio):

- The text in file is: "Steganography aims to hide the messages from unauthorized persons for various purposes"

- The secret key: 6

A: Apply the steps for a single character of a text in file.

- The letter: S

- Character sequence: 1

Convert the letter to double then to binary: $(S) = (83)_{10} = (01010011)_2$

The three least significant bits are transformed to decimal: $(011)_2 = (3)_{10} = \text{DISTANCE}$

Added the key to the letter(s): $(S) + (\text{secret key}) = 83 + 6 = 89 = (Y)$

Add the distance to the sequence of the character: $3+1 = 4$

The next letter to which we add the key is its sequence: $4 = (n)$, either (t, e) is pure

The previous steps are repeated until the text file is finished.

B: Apply the steps for all the letters of the text.

1. convert text file to ASCII code:

```
83 116 101 103 97 110 111 103 114 97 112
104 121 32 97 105 109 115 32 116 111 32
104 105 100 101 32 116 104 101 32 109 101
115 115 97 103 101 115 32 102 114 111 109
32 117 110 97 117 116 104 111 114 105 122
101 100 32 112 101 114 115 111 110 115 32
102 111 114 32 118 97 114 105 111 117 115
32 112 117 114 112 111 115 101 115
```

2. Finding the distance between letters represented by the three least significant bits.

- character sequence = 1, ASCII = $(83)_{10} = (01010011)_2$
 $011 = 3$, distance the next character sequence = the current character sequence + distance = $1 + 3 = 4$

- character sequence = 4, ASCII = $(103)_{10} = (01100111)_2$
 $111 = 7$, distance the next character sequence = the current character sequence + distance = $4 + 7 = 11$

- character sequence = 11, ASCII = $(112)_{10} = (00110000)_2$
 $000 = 01$ distance the next character sequence = the current character sequence + distance = $11+1 = 12$

- character sequence = 80, ASCII = $(117)_{10} = (1110101)_2$
 $101=5$, distance the next character sequence = the current character sequence + distance = $80+5 = 85$, and so on.

- The distance between all the letters:

```
3 7 0 0 1 0 1 1 5 0 0 1 4 0 5
3 5 7 4 1 2 4 2 7 0 6 3 5 5
```

3. ASCII for all the letters to which the key will be added according to the distance:

```
83 116 101 103 97 110 111 103 114 97 112
104 121 32 97 105 109 115 32 116 111 32
104 105 100 101 32 116 104 101 32 109 101
115 115 97 103 101 115 32 102 114 111 109
32 117 110 97 117 116 104 111 114 105 122
101 100 32 112 101 114 115 111 110 115 32
102 111 114 32 118 97 114 105 111 117 115
32 112 117 114 112 111 115 101 115
```

- ASCII code after adding the key (key=100):

```
183 116 101 203 97 110 111 103 114 97 212
204 221 132 197 205 209 115 32 116 111 132
204 205 200 101 32 116 204 201 32 109 101
115 215 97 103 201 115 32 102 114 211 109
```

32 117 110 97 117 216 104 111 114 205 222
 101 200 32 112 101 214 115 211 110 115 32
 102 111 114 132 218 97 114 105 111 117 215
 32 112 217 114 112 111 115 201 115

4. The text with key in file:
 ".teĒanograĖY,,AĪÑs to,,ĪĒe tĪE mes×agĒs frĖM
 unauĖhorĪpeĒ peĖsĖns for,,Ūariou× pŪrposĒs"

However, Figures 1 and 2 demonstrate how to add a secret key to image and audio, files respectively as to the same steps of the proposed method.

Original Image (RGB) Gray Image + Add Key



Figure 1. The original image file and the image after adding the secret key to the pixels (the original image from FEI Database [19])

ORIGINALIMAGE(RGB) GRAYIMAGE+ADDKEY

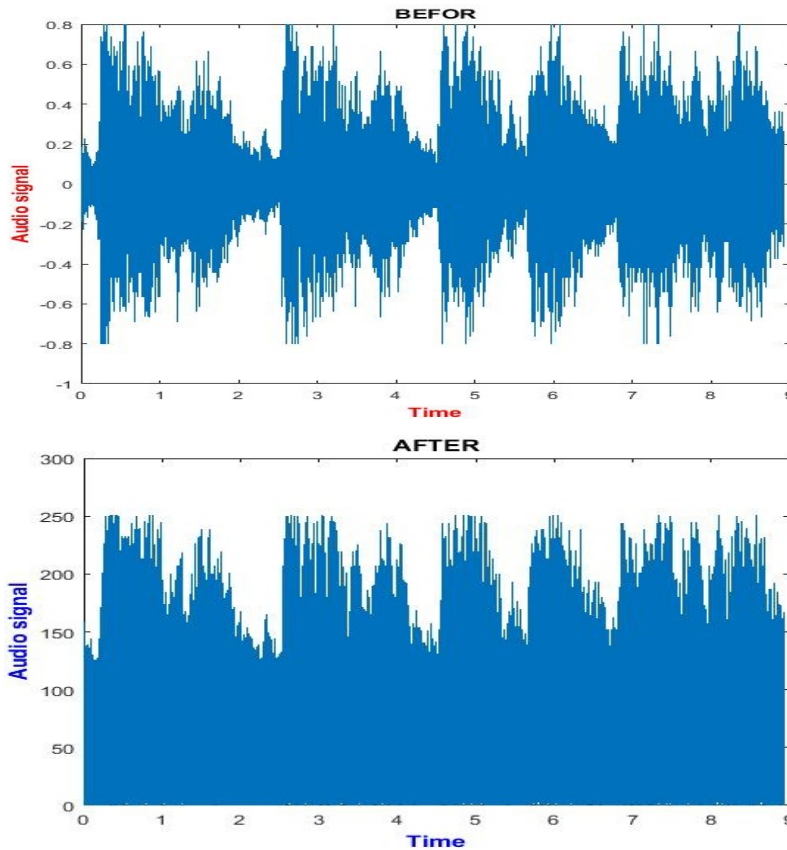


Figure 2. Original audio before and audio after adding secret key (original audio: Create 'handel.wav' from 'handel.mat' and read data back into MATLAB)

The similar strategy was used to hide the image in the least significant sections of the pixel as well. Because Gray levels range from 0 to 255, if the key is added to the pixel that has been defined, the resultant value should not exceed 255. Subtract 255 from the result if it exceeds. The audio file follows the same procedure.

3.1.2. B-Cover Image

Before embedding, the cover image is the original image that has been identified to house the secret text, image, and audio data. The proposed work uses color images as a covering medium. Before inserting the secret messages, the cover image is turned 180 degrees as in Figure 3.

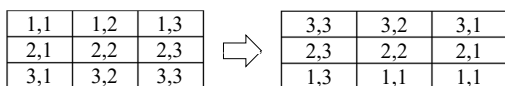


Figure 3. The cover image (rotated at 180°)

3.2. Embedding Stage

Three steps are involved in embedding confidential messages as following:

3.2.1. Red Channel-Embedding Step

The text file's secret message is encoded in the red channel, which uses four of the least essential bits (4 bits) for each pixel, effectively hiding each letter by two pixels. The embedding of a letter is demonstrated in the following steps:

The first pixel after rotating (1,1): $(100)_{10} = (01100100)_2$
 The second pixel after rotating (1,2): $(205)_{10} = (11001101)_2$
 Letter = Y = $(89)_{10} = (01011001)_2$
 First pixel after embedding (1,1): $(01100101)_2 = (101)_{10}$
 Second pixel after embedding (1,2): $(11001001)_2 = (201)_{10}$

Same way the remaining text in file is embedded

3.2.2. Green Channel-Embedding Step

The image file's secret message is encoded in the green channel, which uses four of the least essential bits (4 bits) for each pixel, resulting in two pixels embedded in each pixel of the image to be hidden, as illustrated in the example above (instead of ASCII the character, use the pixel value).

3.2.3. Blue Channel-Embedding Step

The secret message represented by the audio file is encoded in the blue channel, where each value (audio file data) is represented by two pixels using four of the least essential bits (4 bits). The data in an audio file can be negative or positive; in this case, we'll convert to positive numbers and create an array of 0 and 1 (0 for positive, 1 for negative) to aid in data retrieval. Figure 4 depicts an overview of the proposed steganography method while Figure 6 shows the graphical representation of the proposed method.

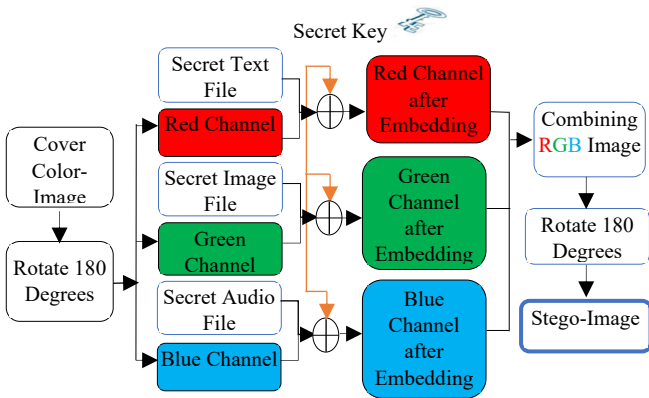


Figure 4. Overview of the proposed steganography approach

4. EXTRACTING STAGE

The identical embedding process is used in reverse in the extraction algorithm. The receiver should be aware of the essential steps in the embedding approach as well as information about the embedding process. Figure 5 depicts the extraction procedure.

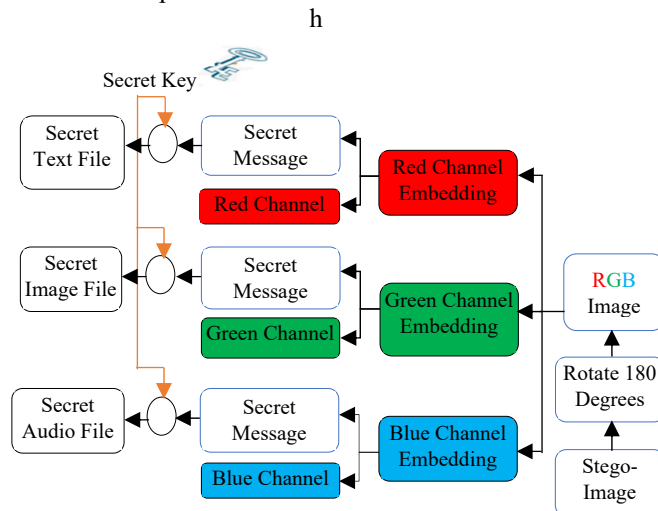


Figure 5. Extracting process of the proposed algorithm

5. EXPERIMENTAL RESULTS

For the evaluation, two measures were executed utilizing the MATLAB environment, a data set of human face images, and several types of random images. The steganography system used to analyze and measure the outcomes obtained by the proposed method. In this work, HFM and HSM performance criteria are used [17-18]. HFM consist a combination of two measures as in Equation (1). They are HSSIM which depends on information theoretical techniques and the feature similarity measure FSM which depends on statistical techniques.

$$HFM(p_1, p_2) = \sqrt{H(p_1, p_2)(K) + F(p_1, p_2)(K-1)} \quad (1)$$

HSM is a combination of two measures also. The first measure is modifying HSSIM, and the second measure is structure similarity measure SSM depends on statistical techniques as the following Equation (2).

$$HSM(p_1, p_2) = H(p_1, p_2)(1-L) + S(p_1, p_2)(L) \quad (2)$$

The cover-images and stego-images are shown in Figures 7, 8 and 9. Tables 1, 2 and 3 show comparisons of performance evaluation measures between them.

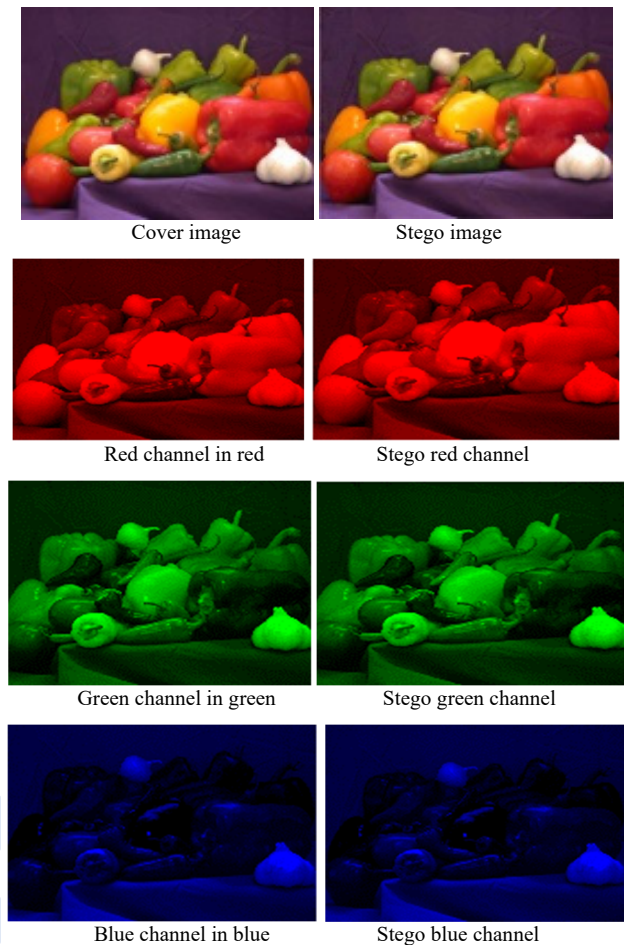


Figure 7. Cover-image (peppers.png) and stage-image with RGB channels

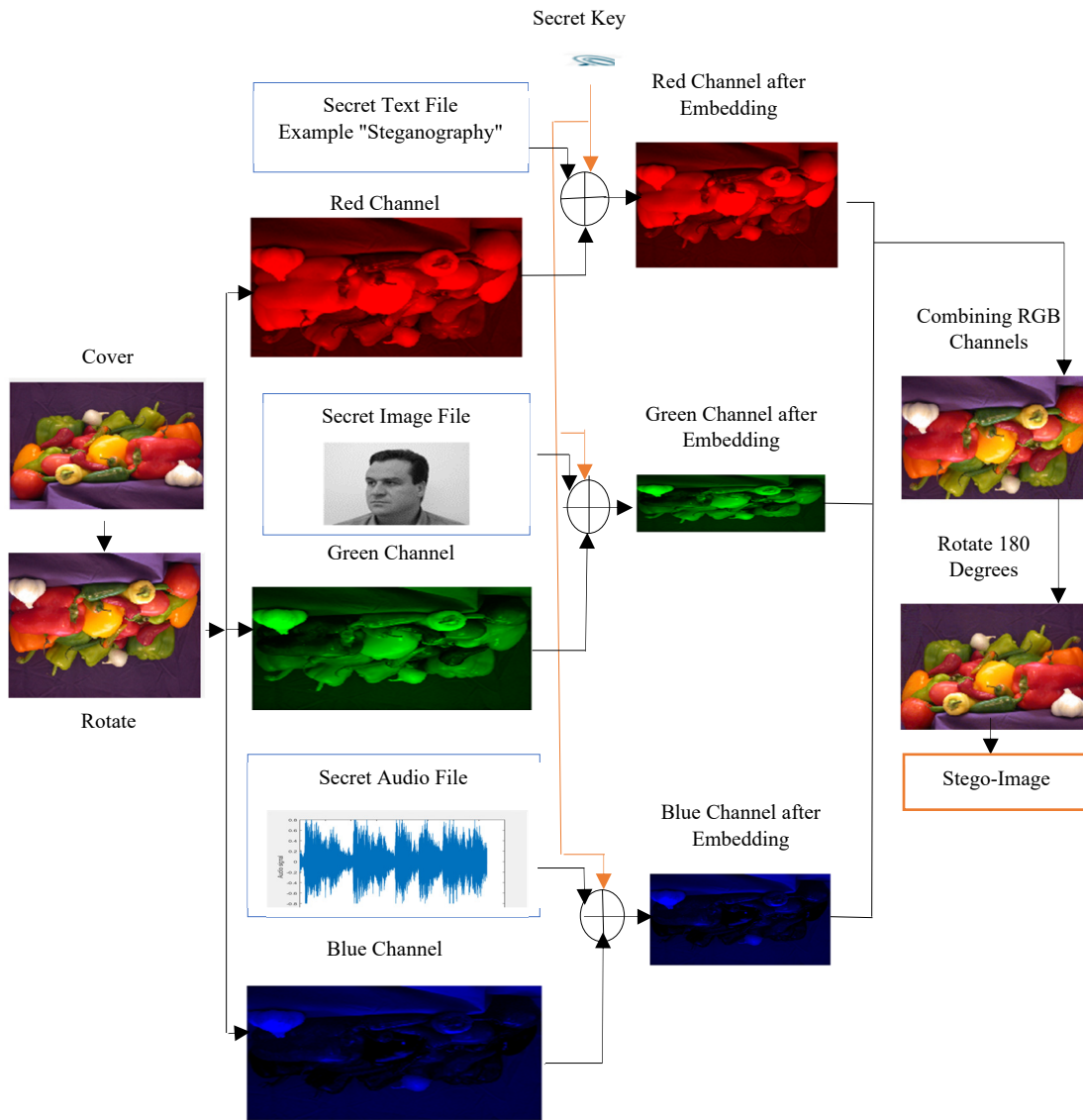


Figure 6. Graphical representation of the proposed method

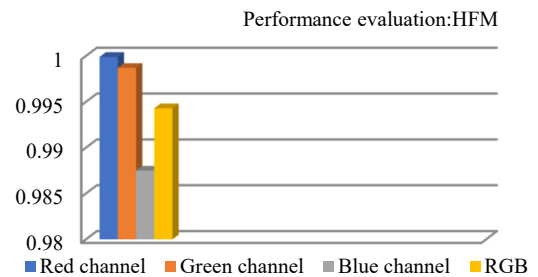
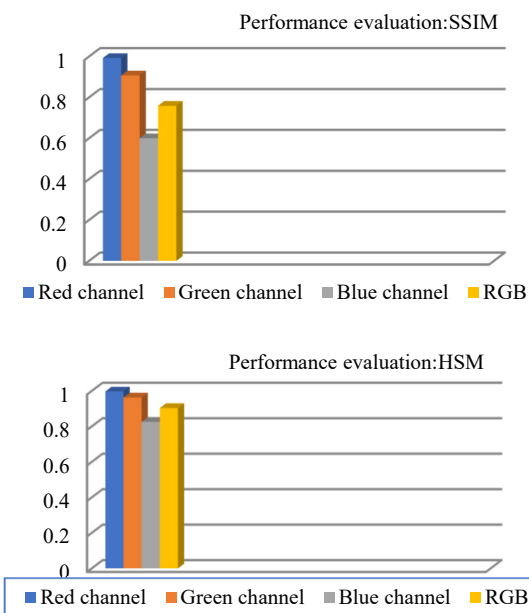
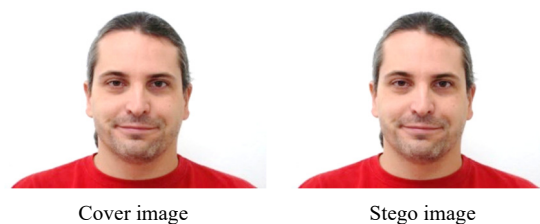


Figure 8. Comparisons of performance measures between cover-image (peppers.png) and stage-image' channels



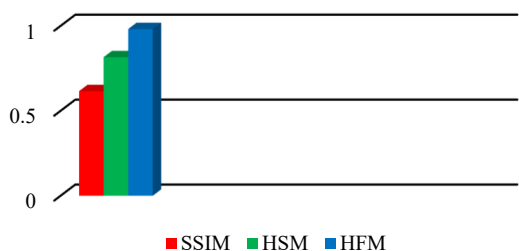


Table 1. Comparisons of performance measures between cover-image (ngc6543a.jpg) and stego-image

Measure	RGB / Stego-Cover
SSIM	0.6155
HSM	0.8162
HFM	0.9818
PSNR	38.7053

Table 2. Comparisons of performance measures between cover-image (peppers.png) and stego-image

Measure	RGB / Stego-Cover
SSIM	0.7596
HSM	0.9037
HFM	0.9943
PSNR	28.1274

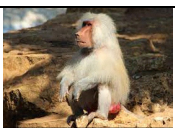


Table 3. Comparisons of performance measures between cover-image (ngc6543a.jpg) and stego-image (R/G/B)



Measure	Red/Stego-Cover	Green/Stego-Cover	Blue/Stego-Cover
SSIM	0.9973	0.9336	0.6174
HSM	0.9986	0.9596	0.8110
HFM	0.9999	0.9904	0.9737
PSNR	62.4539	42.3415	35.5033

Table 4. Comparisons of performance measures between the original image, the encrypted image (images in Figure 1), and the image after recovery and decoding

Measure	Image before encryption / image after encryption (add the key)	Original image / image after recovery and decryption
SSIM	0.0589	0.9990
HSM	0.6236	0.9996
HFM	0.8727	0.9999
PSNR	9.9817	Inf

Table 5. Comparisons of the results between the previous methods in [20, 21] and the proposed method

Cover image	Method [19]	Method [20]	Proposed Method
	PSNR	PSNR	PSNR
 Baboon	37.91	36.72	37.9370
 Average	37.93	38.04	38.2560
 Boats	37.93	38.10	38.1791

 Peppers	37.92	38.35	38.7053
 Tank	-	38.81	40.8862

However, Figures 10 and 11 illustrate the differences of histograms by analyzing a stego-image histogram and comparing it to the cover-image subjectively to determine successful steganography.

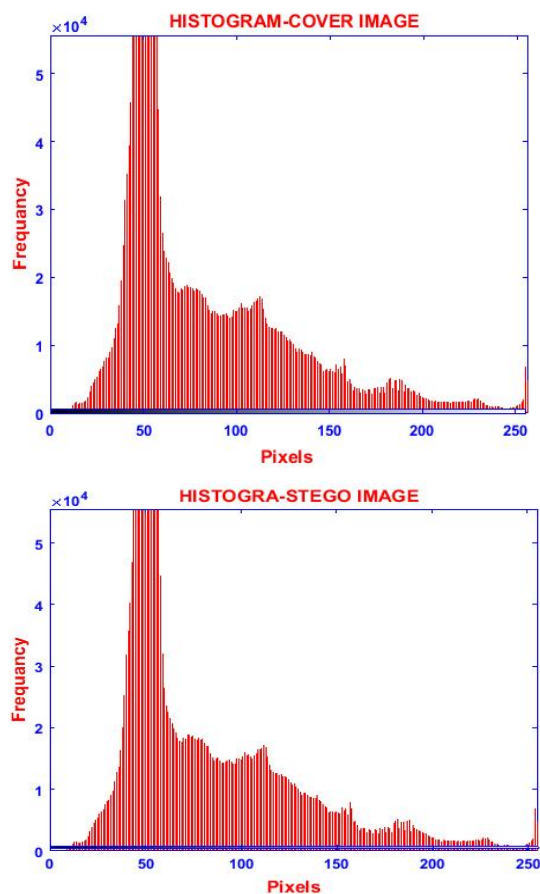
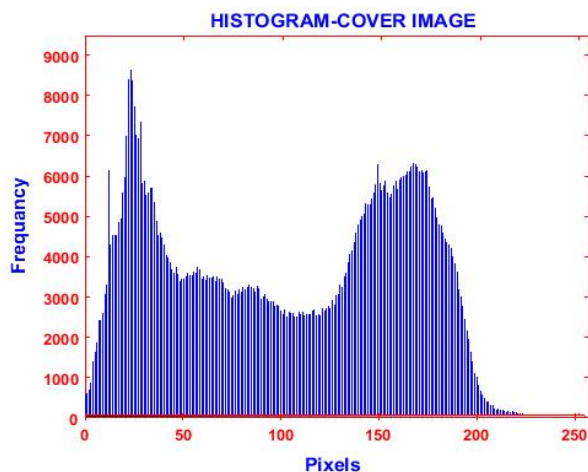


Figure 10. Cover-image histogram (peppers.png) and stego-image



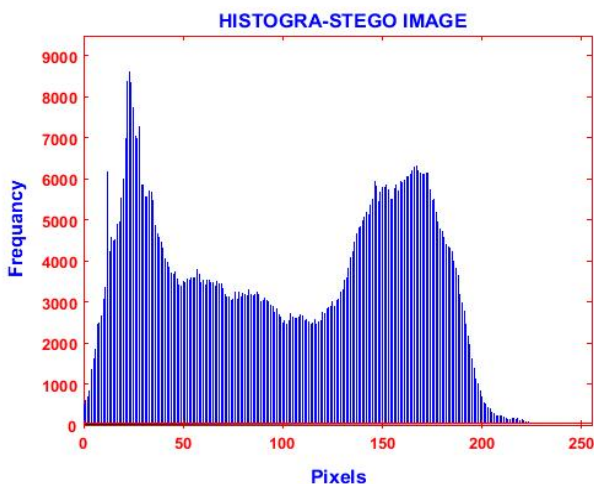


Figure 11. Cover-image histogram (boat.jpg) and stego-image histogram

6. CONCLUSION

The proposed method has been successful in hiding numerous types of files in color images without causing any changes, as well as being difficult to detect by merging two techniques which are the pure and secret key. For masking, images with a lot of detail (i.e., a lot of texture) are preferred. Any compression operation, image improvement, or extension change that affects the concealed files will result in the loss of all or part of the files and may not be fully restored. It is more secure to add the secret key (to be agreed upon) to the files and however, inserting a secret key at random and depending on the less relevant pieces of the hidden data complicates the retrieval process. Experiments and findings reveal that the proposed method is more effective with small files to be hidden. Also, the recommended scales in HFM are superior to the SSIM scale when evaluating image quality. For fare work, it is recommended to develop the method to hide big files.

ACKNOWLEDGEMENTS

The authors appreciate cooperation of Department of Computer Science, Faculty of Education for Girls, University of Kufa, Najaf, Iraq as well as College of Computer Science and Information Technology, University of Al-Qadisiya, Diwania, Iraq.

REFERENCES

[1] A. Pradhan, A.K. Sahu, G. Swain, "Performance Evaluation Parameters of Image Steganography Techniques", International Conference on Research Advances in Integrated Navigation Systems (RAINS), pp. 1-8, 2016.

[2] S. Almanasra, "Parallel Video Steganographic Method over Multi-Core Processors", TEM Journal, Vol. 9, pp. 606-612, 2020.

[3] M.H. Abood, "An Efficient Image Cryptography Using Hash-LSB Steganography with RC4 and Pixel Shuffling Encryption Algorithms", Annual Conference on New Trends in Information and Communications Technology Applications (NTICT), pp. 86-90, 2017.

[4] M.M. Hashim, M.S. Rahim, F.A. Johi, "Performance Evaluation Measurement of Image Steganography

Techniques with Analysis of LSB Based on Variation Image Formats", International Journal of Engineering and Technology, Vol. 4, pp. 3505-3514, 2018.

[5] S.A. Nie, G. Sulong, R. Ali, "The Use of Least Significant Bit (LSB) and Knight Tour Algorithm for Image Steganography of Cover Image", International Journal of Electrical and Computer Engineering, Vol. 9, p. 5218, 2019.

[6] A. Arya, S. Soni, "Performance Evaluation of Secrete Image Steganography Techniques Using Least Significant Bit (LSB) Method", Int. J. Comput. Sci. Trends Technol, Vol. 6, pp. 160-165, 2018.

[7] N.S. Soleimani, M.A. Balafar, "A Review on Data Hiding Upon Digital Images", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 14, Vol. 5, No. 1, pp. 108-113, March 2013.

[8] E.A. Elshazly, S.A. Abdelwahab, R.M. Fikry, Elaraby, "FPGA Implementation of Robust Image Steganography Technique Based on Least Significant Bit (LSB) in Spatial Domain", International Journal of Computer Applications, Vol. 12, No. 11, pp. 43-52, 2016.

[9] M. Mosleh, N. Hosseinpour, "Blind Robust Audio Watermarking Based on Remaining Numbers in Discrete Cosine Transform". International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 16, Vol. 5, No. 3, pp. 18-26, September 2013.

[10] X. Zhou, W. Gong, L. Jin, "An Improved Method for LSB Based Color Image Steganography Combined with Cryptography", IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pp. 1-4, 2016.

[11] A.M. Odat, M.A. Otair, "Image Steganography Using Modified Least Significant Bit", Indian Journal of Science and Technology, Vol. 9, pp. 1-5, 2016.

[12] W.T. Domain, I. Shahidan, "A Review and Open Issues of Diverse Text Watermarking Techniques in Spatial Domain", Journal of Theoretical and Applied Information Technology, Vol. 17, 2018.

[13] S. Sugathan, "An Improved LSB Embedding Technique for Image Steganography", The 2nd International Conference on Applied and Theoretical Computing and Communication Technology, pp. 609-612, 2016.

[14] I. Gunawan, "Use of Least Significant Bit Cryptographic Steganography Algorithm for Securing Text Messages and Video Data", J-SAKTI (Journal Sains computer dan Informatics), Vol. 2, pp. 57-65, 2018.

[15] S. Solak, U. Altinisik, "LSB Substitution and PVD Performance Analysis for Image Steganography", International Journal of Computer Sciences and Engineering, Vol. 10, pp. 1-4, 2018.

[16] S. Rahman, F. Masood, W.U. Khan, N. Ullah, "A Novel Approach of Image Steganography for Secure Communication Based on LSB Substitution Technique", Computers, Materials and Continua, Vol. 64, pp. 31-61, 2020.

[17] N. Hamza, R. Dihin, M.H. Abdul Ameer, "A Hybrid Image Similarity Measure Based on a New Combination of Different Similarity Techniques", International Journal of Electrical and Computer Engineering, Vol. 10, p. 1814, 2020.

- [18] R.A. Dihin, N.R. Hamza, Z.H. Toman, "Full-Reference Facial Image Quality Assessment and Identification by Two Proposed Measures", Journal of Southwest Jiaotong University, Vol. 55, pp. 110-118, 2020.
- [19] C.F. Lee, H.L. Chen, "A Novel Data Hiding Scheme Based on Modulus Function", J. Syst. Softw., Vol. 83, pp. 832-843, 2010.
- [20] M. Khodaei, K. Faez, "New Adaptive Steganographic Method Using Least-Significant-Bit Substitution and Pixel-Value Differencing", IET Image Processing, Vol. 6, pp. 677-686, 2012.

BIOGRAPHIES



Rasha Ali Dihin was born in Najaf, Iraq, 1988. She received her M.Sc. degree from Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq in 2018. Currently she is a Ph.D. student at the same university. Her general specialization is in computer science (artificial intelligent and image processing). Her research interests are profound contributions to different fields of computer programs, biometrics, and automatic

identification systems, statistical analyses of health, medical studies, and information technology.



Nisreen Ryadh Hamza was born in Dywania, Iraq, 1984. She received her M.Sc. degree from Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq in 2018. Her general specialization is in computer science (artificial intelligent and image processing). Her research interests are profound contributions to different fields of computer programs, biometrics, and automatic identification systems.



Hasan Thabit Rashid was born in Najaf, Iraq, 1982. He is currently a lecturer with University of Kufa, Najaf, Iraq. He received his Bachelor degree in Computer Science from Babylon University, Hillah, Iraq in 2005, and his Master degree in Information Technology from UNITEN University, Malaysia in 2013, and his Ph.D. in computer science from Babylon University, Iraq in 2021. His research interests include intelligent video surveillance systems, image and video analyzing, and computer vision.