Al-Rafidain Journal of Computer Sciences and Mathematics (RJCM)

www.csmj.mosuljournals.com

Adaptive Steganography Using Improve Bit-Plane Complexity Segmentation

Noor Gassan Abdullah[1], Shahd Abdulrhman Hasso[2, *]

[1,2]*Software Department, College of Computing and Mathematics, University of Mosul, Iraq*
Emails: noor.22csp18@student.uomosul.edu.iq, *shahd_hasso@uomosul.edu.iq*

Article Information

Article history:
Received: 31/10/2023
Accepted: 17/12/2023
Available online: 25/6/2024

Abstract

One of the primary challenges in Internet data transmission lies in safeguarding data from unauthorized access by potential attackers. The goal of content-adaptive steganography is to conceal data inside the image's intricate texture. This research introduces an improved algorithm for concealing messages within color images. The developed method incorporates bit-plane slicing and the RSA algorithm as its foundation, aiming to achieve a heightened level of security for data hiding. The algorithm's uniqueness lies in its adaptability, where the threshold is determined based on both the characteristics of text and image. Subsequently, the public key is employed for encryption the thresholds, while the private key is utilized to decrypt it. Performance criteria such as Mean Squared Error (MSE), Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index Method (SSIM) are used to assess the quality of the developed algorithm. The results indicate that when 90,000 bits are concealed in the image, it yields an acceptable PSNR value of 60.5749, MSE of 0.0569, and SSIM of 0.9996. The developed algorithm excels in data hiding, as evidenced by its favorable comparison with other studies.

Keywords:
Steganography, adaptive steganography, Bit Plane Complexity Segmentation (BPCS), RSA.

Correspondence:
Author: Shahd Abdulrhman Hasso
Email: shahd_hasso@uomosul.edu.iq

1. Introduction

As multimedia communication continues to expand, safeguarding the integrity of digital data during transmission and storage has become a critical issue [1]. Various security methods have been devised for transmitting data over networks, with cryptography and steganography being prominent examples [2]. Data hiding methods like Digital Watermarking, cryptography, and steganography ensure securing communication of sensitive information, protecting it from breaches and malicious attacks in digital networks [3]. Cryptography involves transforming data to prevent unauthorized access, whereas steganography conceals the presence of secret information [4]. Recently, certain security steganography techniques have been put forth to determine an image's degree of textural complexity [5].

Steganography finds its roots in the Greek language, where "steganos" translates to cover and "graphia" means writing. Together, these words describe the technique of concealing information within other data [6]. Steganography employs diverse techniques like invisible inks and digital signatures to discreetly embed messages within media, with the goal of making the message so well hidden that its presence is undetectable [7]. Traditional steganography methods have limited data concealment capabilities, often hiding only up to 10% of the vessel's data capacity. To overcome the limitations of traditional steganographic methods, Eiji Kawaguchi and R. O. Eason [8] introduced a novel technique named Bit-Plane

Complexity Segmentation (BPCS). BPCS markedly improves data concealment, enabling the utilization of approximately half of the container size. This method replaces the "noise-like" regions within the bit-planes of the vessel image with secret data without compromising image quality. Consequently, this research has opted for the BPCS technique for image steganography due to its superior performance.

2. Related Work

The following is some of the research related to the proposed method. In this part, some of the previous works will be illustrated from 2019 to 2023, which are related to adaptive image steganography.

Rajeev et al. [9] have proposed an innovative approach to data concealment by employing adaptive quantization and a dynamic bit-plane-based technique known as Adaptive Modified Block Truncation Coding (AMBTC). Initially, their method classifies original image blocks into four distinct categories: "Absolute Smooth," "Slightly Smooth," "Slightly Complex," and "Absolute Complex." Subsequently, they utilize the bit-planes of these blocks to hide secret data through a straightforward bit replacement strategy. The process begins by partitioning a cover image into non-overlapping blocks. For each block, the method determines an adaptive number of quantization levels and the appropriate bit-plane size based on the block's complexity level. Secret data is then embedded into the bit-planes of these blocks, taking into account their tolerance capabilities. This embedding accomplished using a simple bit replacement technique. One noteworthy aspect of this approach is its dynamic categorization of image blocks as well as its ability to determine the optimal number of quantization levels and bit-plane sizes. These aspects play a great role in ensuring the resulting marked image that maintains high quality while preserving the same bitrate as the original. Additionally, this proposed scheme boasts a substantial data hiding capacity, as nearly every block in the image utilized for concealing the secret data.

Bishwas et al. [10] have presented a new steganography method that uses the least significant bit (LSB) substitution and the pixel value differencing (PVD) method. The gray scale image is split into 3 x 3 blocks if the image size is divisible by 3 but if the image is indivisible by 3 then split it into blocks in size 3 x 3 plus 2 x 2 pixels. One pixel in the block called the reference pixel, where 4 bits of the LSB replaced. Next, the difference values between the center pixel and each neighboring pixel are calculated. Based on these dissimilarity values, adaptive LSB replacement applied in neighboring pixels.

Serdar et al. [11] have. Presented the adaptive LSB+3 algorithm. In the first step, the data to be hidden encrypted using text and digital messages received from the user. The encrypted data is then hidden inside the cover image using first-type adaptive LSB+3 or second-type adaptive LSB+3 techniques which were developed as a result of improving the LSBXYZ method.

Mohamed et al. [12] have proposed an adaptive data hiding technique that employs Histogram of Oriented Gradient (HOG) to embed secret data in digital images using PVD-LSB methods. Blocks of Interest (BOIs) are dynamically selected based on edge content using HOG. PVD conceals secret bits along the dominant edge, while LSB substitutes bits in the other two pixels, enhancing embedding capacity, visual quality and security. This method focuses on edge pixels for embedding, adapting bit counts per 2x2 BOI. It controls the PSNR versus capacity trade-off by adjusting thresholds and k-bits based on secret message length. Compared to other PVD-LSB-based methods, it excels in embedding, quality, and concealment. It extends to color images, offering adaptive channel capacity selection.

Zin et al. [13] have proposed system centers on fundamental Steganography and different requirements for data concealment. Their primary contribution is developing of the BPCS algorithm, which allows for the concealment of data within a color vessel image. It is demonstrated that the final embedded image produced by the BPCS algorithm locates noise-like regions in a cover image more precisely while appearing to be identical to the original image. The system examines image histograms to determine the embedding potential of various types. The suggested approach demonstrated, based on the histogram distribution, that the embedding capacity affects both the host image quality and size.

Bilgi et al. [14] have proposed an innovative technique for concealing data within images, employing a combination of bit plane slicing and double XOR encryption. The methodology begins by dissecting a color cover image into its individual RGB channels and focusing on the pixel values within the R channel. Specifically, they isolate the R channel and further break it down into its constituent bit planes, with the intent of concealing information within this channel. The data hiding process is initiate within the Most Significant Bit (MSB) plane of the R channel, utilizing a key that comprises as many bits as the length of the message to be conceal. In the first step, the bits of the message information are XORed with the bits of the MSB, constituting the initial XOR operation. Subsequently, a second key is generated by negating the bits of the MSB key. The result derived from the initial XOR operation then subjected to a second XOR operation with this newly generated key, constituting the secondary XOR operation. The resultant encrypted message discreetly embedded within the Least Significant Bit (LSB) plane of the R channel. To retrieve the concealed information from the stego image, the reverse sequence of operations applied, effectively decrypting and extracting the hidden data.

Kelvin et al. [15] have proposed a steganography method combines BPCS with Vigenere cipher encryption to embed text securely in color images. It leverages bit-plane complexity segmentation, binary image conjugation, and identifies informative and noise-like regions for efficient data hiding. Vigenere cipher with extended character support ensures message security and integrity.

Table 1 displays a comparison of the methods mentioned above, highlighting the best result obtained from each method.

Table 1. Comparative data for different algorithms.

Researcher	Data Size	Image	PSNR
Rajeev et al. [9]	140181	Zelda	40.12
Bishwas et al. [10]	1051375	Tank	32.91
Serdar et al. [11]	1,800,000	Baboon	42.175
Mohamed et al. [12]	800000	Lake	44.91
Zin et al. [13]	64000	Color image	43
Bilgi et al. [14]	2552	Baboon	55.929
Kelvin et al. [15]	320000	Mobil-phone	45

3. Image Block Complexity

In this paper, the method for concealing data relies on utilizing the complex portion within an image. There is no established metric for image complexity. Black-and-white (B-W) border image complexity is the measure of picture complexity that Kawaguchi [8] suggested. To gauge the complexity of the image, the binary image records the length of the black-and-white border. A complex image has a longer border; otherwise, it is simple. The amount of color shifts along all the image's rows and columns is added together to determine the border's length. For instance, a black pixel has a border length of 4 if it is bordered by white backdrop pixels. The image block complexity to a block of size $2^N * 2^N$ defined as follows:

$$\alpha = \frac{K}{2*2^N*(2^N-1)}, 0 \le K \le 2*2^N*(2^N-1) \quad (1)$$

Where:
N: The power of 2 (size of block is $2^N * 2^N$, In this paper, used a block size equal to $2^3 * 2^3$, which is 8*8, then N=3).
k: The actual length of the black and white border in the image.
It is evident that α lies in [0, 1] [13].

4. Public Key Cryptography

In public key cryptography, a key pair is utilized. The sender employs the recipient's public key to encrypt data, and the recipient uses their private key to decrypt it. This approach ensures that data can be securely encrypted with a freely distributed public key, while only the private key holder can decrypt it [16]. In the proposed method, the RSA algorithm employed among the various public key cryptography algorithms available.

4.1 RSA encryption

To clarify, RSA encryption and decryption are as follows [17]:

1. Key Generation Procedure:
   - Suppose two distinct large random prime numbers p & q such that p ≠ q.
   - Calculate n = p x q
   - Calculate Euler of n: Φ (n) = (p-1) * (q-1)
   - Choose an integer e such that the greatest common divisor (GCD) of Euler n and e equal to 1 and

     1< e <Φ (n)
   - Compute d to satisfy the congruence relation

     d = e^{-1} mod Φ (n)
   - Public Key PuK = {e, n}
   - Private Key PrK = {d, n}
   - The public key is (e, n) and the private key is (d, n).

     Keep d, p, q and Φ secret.
2. For encryption:
   - Message M < n
   - C = M^e mod n
   - Where, C is cipher text, M Message, and (e, n) public key.
3. For decryption:
   - C <n
   - M = C^d mod n.
   - Where, M is Message, C cipher text, and (d, n) private key.

The RSA algorithm was employed for encrypting/decrypting the threshold values. By encrypting, the threshold boundaries with the recipient's public key, in the proposed method guarantee only the recipient, possessing the corresponding private key, can decrypt these values. Given the crucial role of these values in retrieving information, it is imperative, as without them, no information can be retrieving.

5. Methodology

5.1 Bit-Plane Complexity Segmentation (BPCS)

The Bit-Plane Complexity Segmentation (BPCS) algorithm goes through the following steps while inputting data [8]:

Step 1: the cover image converted from Pure Binary Code (PBC) to Canonical Gray Code (CGC) system. Subsequently, the image is sliced into binary bit-planes, with each bit-plane representing a pixel's bit. The conversion formula from Pure Binary Code (PBC) to Canonical Gray Code (CGC) is applied as follows:

$$CGC_1 = PBC_i \quad (2)$$

$$CGC_i = PBC_{i-1} \oplus PBC_i \quad (3)$$

Step 2: each bit-plane of the cover image is separating into informative and noise-like regions using a threshold value (α_0), commonly set to 0.3.

Step 3: Separate the secret message bits into secret message blocks.

Step 4: If block S is too simple, conjugate it to a more complex form called S*. S* is always more complex than the limit value.
Where:

$$S^* = 1 - S$$

Step 5: Insert each secret message block within a noise-like region in the bit-plane or replace the bits within this area. The data should be stored in the conjugation map if the S block had conjugated.

Step 6: Include a conjugation map, similar to what you did for the hidden message block.

Step 7: Convert the stego image from Canonical Gray Code (CGC) to Pure Binary Code (PBC).

5.2 Proposed Method

In the proposed method, a color image used, which contains three colors (Red, Green, and Blue (RGB)) of 24 bits. Each color comprises 8 bits, and the value of each of these colors is represented using a number from 0 to 255.

Embedding algorithm

Input: Cover Image, Secret Data Message

Output: Stego Image

Step 1: Read the message and the cover image and convert it from Pure Binary Code (PBC) to Canonical Gray Code (CGC) system. Followed by split image into its channels then each channel sliced into binary bit-planes.

Step 2: Divide each bit plan into blocks in size 8*8

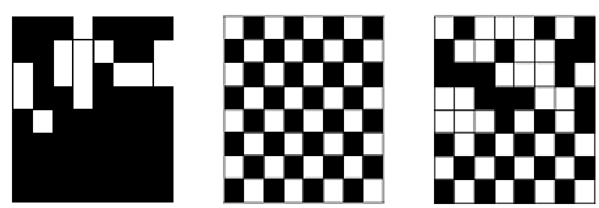
Step 3: Calculate complexity measure "alpha" (α) using the relations in equation (1) for each block of each bit plane.

Step 4: Taking the maximum threshold value (max'_α) obtained, so that max'_α is not less than 0.5, then adopting it to compute the initial minimum threshold value (min'_α) as follows:

$$min'_\alpha = \frac{max'_\alpha}{2}$$

Step 5: convert the message to the ASCII system and then to binary followed by converting the binary message into blocks of size 8 * 8 (S) and store the length of the message as the first block. Set the first bit of each block to 0.

Step 6: Calculate α for each block of secret message, if block S is out of threshold range, conjugate (XORing with Wc) it to be within acceptable limits, a block after conjugate it called S*. If min'_α=0.3526785714 and max'_α =0.7053571428, the example in **Fig. 1** illustrates the conjugation operation.



S, α=0.25892857 | Perfectly complex block, Wc, α=1 | S*, α=0.7410714285

Fig. 1. Example to illustrate the Conjugation operation for this algorithm

Step 7: Calculate α for each block of secret message, and take the maximum α as the maximum α threshold (max_α) and the minimum α as the minimum threshold (min_α).

Step 8: Encrypting threshold values using the RSA algorithm.

Step 9: Transferring the encrypted threshold values to the binary and dividing them into 8*8 blocks, then inserting them into the first blocks in LSB.

Step 10: Insert each secret message block within a noise-like region in the bit-plane.

Step 11: Convert the stego image from Canonical Gray Code (CGC) to Pure Binary Code (PBC).

These Steps to embed the text message illustrated in **Fig. 2**, and the steps for extract the message from the image illustrated in **Fig. 3**.

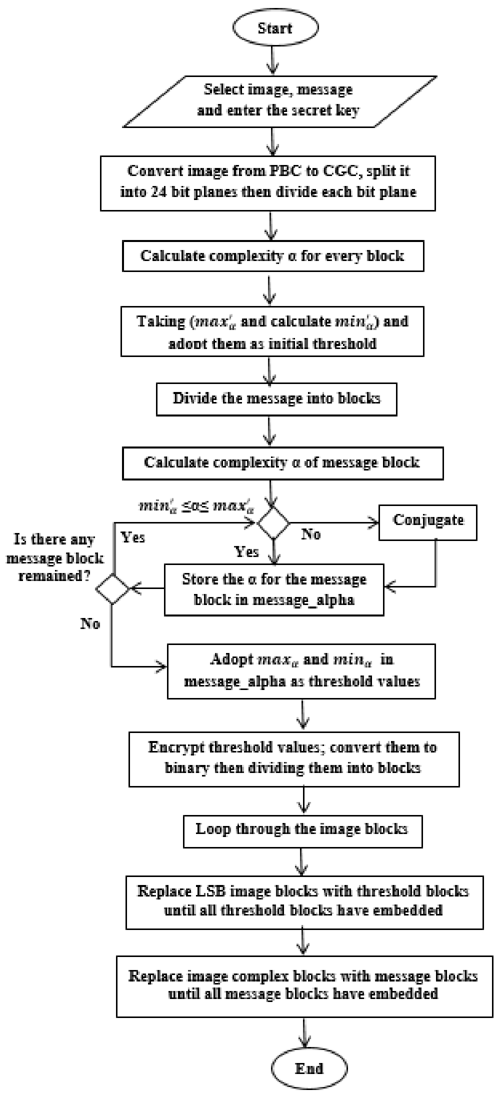


Fig. 2. The proposed algorithm for embedding procedure.

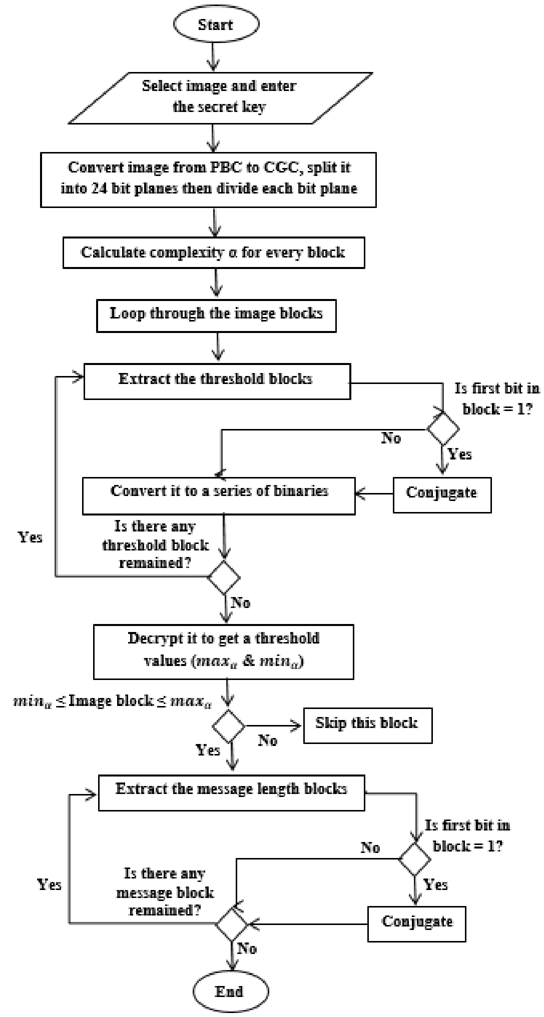


Fig. 3. The proposed algorithm for extraction procedure.

6. Results And Discussion

The proposed algorithm has been developed as a piece of software. In the proposed method created a database of 500 images over than 190 KB is downloaded from the Internet and some of them from my own photography. It can be proposed also one can use an image from the recommended image processing database [18]. The hidden messages in the proposed method used were text-based. When assessing the results, it focused upon the following metrics:

- **Mean Square Error (MSE):** For gray scale images, the MSE can be calculated based on the X × Y dimensions. However, for color images (RGB images), the MSE is determined based on the X × Y × C dimensions, as expressed below:

$$MSE = \frac{1}{XYC}\sum_{i=0}^{X-1}\sum_{j=0}^{Y-1}\sum_{k=0}^{C-1}[O(i,j,k) - S(i,j,k)]^2 \quad (4)$$

Where:

X, Y are the dimensions of the image

C is the number of image channels (Red, Green, and Blue)

O (i, j, k) is a pixel of the original image

S (i, j, k) is the pixel of the stego image [19].

- **Peak Signal to Noise Ratio (PSNR)** A common quality-measurement method in image and signal processing research is PSNR. PSNR results from the calculation of the logarithm of the mean square error (MSE) of an image.

$$PSNR = 20\, log_{10}(\frac{MAX_c}{\sqrt{MSE}}) \qquad (5)$$

Where:

MAX_c: The maximum value used to identify a color [20].

- **Structural Similarity Index (SSIM):** the Structural Similarity (SSIM) index is used to gauge how similar between two images. The SSIM index can be used to determine the quality of the second image in the comparison when an SSIM score of 1 indicates a perfect match between the compared images.

$$SSIM(O,S) = \frac{(2\mu_O\mu_S+d_1)(2\sigma_{OS}+d_2)}{(\mu_O^2+\mu_S^2+d_1)(\sigma_O^2+\sigma_S^2+d_2)} \qquad (6)$$

Where:

O: The original image

S: The stego image

μ_o, μ_s: The summations of O and S

σ_S, σ_O: The variances of O and S

σ_{SO}: The variances [21].

7. Results Display

The results obtained by applying the proposed method are shown in **Fig. 4**. In addition, **Table 2** summarizes results of the MSE, PSNR and SSIM between the original image and the stego image that are found using the relations in equations (4), (5) and (6) respectively.

Babbon

Stego image

pepper

Stego image

Cupcake

Stego image

Fig. 4. Left images are cover images while the right images contain embedded messages.

Table 2. Shows the PSNR, MSE and SSIM values vs. messages length

Image	Hidden Data (bits)	PSNR	MSE	SSIM
Babbon 512*512	900000	48.4485	0.9294	0.9979
Pepper 512*512	90000	60.5749	0.0569	0.9996
Cupcake 1920*1080	7000000	48.6344	0.9959	0.8904

Table 3 shows a comparison between our algorithm and other adaptive steganography algorithms.

Table 3. Comparative data between the proposed algorithm and different algorithms.

Cover Image	Hidden Data (bits)	Method	PSNR	MSE
Lena	116200	Proposed Method	59.3225	0.076
	115060	0R. Kumar et al. [9]	36.11	~~~
	104055	M. Hameed et al. [10]	36.32	~~~
	2552	B. Özdemir et al. [12]	55.903	0.166
	80000	K. Kusumah et al. [13]	48	~~~
Pepper	114000	Proposed Method	59.4338	0.074
	112859	0R. Kumar et al.	36.74	~~~

		[9]		
	58603	M. Hameed et al. [10]	45.65	~~~
	2552	B. Özdemir et al. [12]	55940	0,165
Babbon	130000	Proposed Method	58.8747	0.084
	125900	0R. Kumar et al. [9]	29.1	~~~
	105880	M. Hameed et al. [10]	35.40	~~~
	2552	B. Özdemir et al. [12]	55.929	0.165

7. Conclusion

In this paper, a proposed method is presented by applying image steganography using improved bit-plane complexity segmentation. It can be proposed that segmenting the image according to the BPCS algorithm, and then calculating the complexity of each block within the image, taking the maximum obtained value and adopting it as the initial maximum threshold, then taking half of this value as the initial minimum threshold. The message blocks outside the range of initial threshold values conjugated. After that, the complexity values of the message blocks are calculated and stored, and the maximum and minimum complexity values are adopted as maximum and minimum threshold for embedding data. As observed in **Table 1**, the proposed algorithm exhibits a notably high hiding capacity, specifically 900,000 bits with an acceptable PSNR value of 48.4485dB for the Babbon image with dimensions 512 × 512 pixels. Similarly, for the Cupcake image with dimensions 1920 × 1080 pixels, the hiding capacity is 7,000,000 bits, accompanied by a PSNR value of 48.6344dB after execution. The quality and size of the image determine the amount of data that can be included without affecting the quality of the image. Therefore, when including data, the size of the data should be taking into account in relation to the size of the cover image and the amount of noise present in it. Furthermore, it is verified that the proposed method is suitable for concealing information within colored cover images.

The proposed method provides a new advance in the field of steganography by providing an adaptive block selection method based on the complexity of the message block as well as the complexity of the image block. In the future, the proposed method can use other complexity metrics.

8. Acknowledgement

The authors express their gratitude to the Software Department, College of Computer Science and Mathematics, University of Mosul, Iraq, for their invaluable support in completing this work.

References

[1] Agneet, C., Sudipta, G. & Ram, S. (2020) LSB based steganography with OCR: an intelligent amalgamation, Springer Multimed Tools Appl 79 , doi: https://doi.org/10.1007/s11042-019-08472-6

[2] Ayesha, S., Fawad, Muhammad, K., Humayun, S., Syeda N., Muhammad, R., Mansoor, K., Yasar A., (2020) An Accurate Texture Complexity Estimation for Quality-Enhanced and Secure Image Steganography , IEEE Access, Vol. 8, pp. 21613 – 21630, doi: 10.1109/ACCESS.2020.2968217

[3] Sonal K., Geeta K., and Singara K., (2019) ADAPTIVE REVERSIBLE DATA HIDING SCHEME FOR DIGITAL IMAGES BASED ON HISTOGRAM SHIFTING, Computing and Informatics, Vol. 38, No. 2, doi: https://doi.org/10.31577/cai_2019_2_321.

[4] Bo O. and May, A. , (2020) Enhancing Secure Digital Communication Media Using Cryptographic Steganography Techniques, IEEE International Conference on Advanced Information Technologies (ICAIT), doi: 10.1109/ICAIT51105.2020.9261790

[5] Sakshi, A., Maruska, M., (2019) A Comparative Study of Cryptography and Image Steganography Technique, International Journal of Information and Computing Science, Vol. 5, Issue 5, https://www.ijfmr.com/papers/2023/5/8217.pdf

[6] Hidayet, Ç., Nurettin D., (2023) K-En Az Anlamlı Bitlere Dayalı Kaotik Bir Harita Kullanan Renkli Görüntü Steganografisi, Journal of Polytechnic, vol. 26, issue: 2, https://doi.org/10.2339/politeknik.1008594

[7] Auday AL-Wattar, (2023) A New Proposed Public Key Cryptography Based on Bio Strands, Technium:,Romanian Journal of Applied Sciences and Technology, vol. 5, https://doi.org/10.47577/technium.v4i3.6361

[8] Eiji, K. & Richard, 0. E., (1999) "Principle and Applications of BPCS Steganography," SPIE DIGITAL LIBRARY, Multimedia Systems and Applications Vol. 3528, doi: 10.1117/12.337436

[9] Rajeev, K., Neeraj, K. & Ki-Hyun, J., (2019) A New Data Hiding Method Using Adaptive Quantization & Dynamic Bit Plane Based AMBTC, IEEE International Conference on Signal Processing and Integrated Networks (SPIN), doi: 10.1109/SPIN.2019.8711774

[10] Bishwas, M., Anita P.& Gandharba S., (2019) Adaptive LSB substitution Steganography technique based on PVD, IEEE Xplore, International Conference on Trends in Electronics and Informatics (ICOEI), doi: 10.1109/ICOEI.2019.8862579

[11] Serdar, S. & Umut A., (2019) Image steganography based on LSB substitution and encryption method: adaptive LSB+3, Journal of Electronic Imaging, doi: 10.1117/1.JEI.28.4.043025

[12] Mohamed,. H. , M. Hassaballah , Saleh, A. , and Ali A., (2019) An Adaptive Image Steganography Method Based on Histogram of Oriented Gradient and PVD-LSB Techniques, IEEE Access, Vol. 7, pp. 185189 - 185204 doi: 10.1109/ACCESS.2019.2960254

[13] Zin, H. and Zar, Z., (2020) IMAGE STEGANOGRAPHY USING BIT PLANE COMPLEXITY SEGMENTATION, International Journal Of Creative and Innovative Research In All Studies (IJCIRAS), Vol. 2, Issue 12, http://www.ijciras.com/PublishedPaper/IJCIRAS1568.pdf

[14] Bilgi. Ö. & Nurettin, D., (2022) Data hiding to the image with bit plane slicing and double XOR, MANAS Journal of Engineering, vol. 10, Issue: 1, https://doi.org/10.51354/mjen.1049851

[15] Kelvin, K., Jeanny, P. & Novario P. , (2022) Steganography Implementation of Insertion of Confidential Data on Digital Image Media with Bit-Plane Complexity Segmentation Method and Vigenere Cipher Extended Encryption Method, EasyChair Preprint no. 8530, https://easychair.org/publications/preprint/DLjs

[16] Jyoti, N. & Ruksar, F., (2020) Adaptive Steganography technique using DIDC Model, International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, doi:10.35940/ijitee.H6138.069820

[17] K. Pavani & P. Sriramya, (2021) Enhancing Public Key Cryptography using RSA, RSA-CRT and N-Prime RSA with Multiple Keys, IEEE Xplore, International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), doi: 10.1109/ICICV50876.2021.9388621

[18] http://sipi.usc.edu/database/database.php?volume=misc

[19] De S., (2020) "PSNR vs SSIM: imperceptibility quality assessment for image steganography," Springer Multimedia Tools and Applications, doi: https://doi.org/10.1007/s11042-020-10035-z

[20] Hristo, p., Aleksandar S. & Radostin R., (2019) Image Steganography using Arduino Microcontroller, Mathematical and Software Engineering, Vol. 5, No. 2, https://www.varepsilon.com/index.php/mse/article/view/64/62

[21] Varshini, A., Tulasi, G., Divya, S., Shahana, B. and Himasri, K., (2021) Similarity Score of Two Images using Different Measures, IEEE 6th International Conference on Inventive Computation Technologies (ICICT), doi: 10.1109/ICICT50816.2021.9358789