Journal of Information Systems Engineering and Management
2025, 10(32s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Integrating Multiple Steganographic Techniques for Medical Image Data Security Enhancement in Healthcare Systems: Security-Enhanced Steganography

Salwa Mohammed Nejrs[1], Azmi Shawkat Abdulbaqi[2,*]
[1] *University of Mustansiriya, College of Art, Baghdad, Iraq*
[2]*Renewable Energy Research Center, University of Anbar, Ramadi, Iraq*
Email: qsalwaaa@uomustansiriyah.edu.iq ; azmi_msc@uoanbar.edu.iq

ARTICLE INFO

Received: 29 Dec 2024

Revised: 12 Feb 2025

Accepted: 27 Feb 2025

ABSTRACT

Steganography provides a mechanism for embedding and retrieving a message within an innocuous carrier. Image steganography is a prevalent and robust application of this technique. While standard steganography conceals a secret message within a single cover image (Cov_img), its security is limited. Mass steganography addresses this limitation by enabling concurrent data storage across multiple images. This paper proposes a novel data partitioning method for embedding data across a set of cover images. It further details the procedures for extracting the hidden information from the Cov_img at the receiving end. Dividing the data in this way enhances security by making it exceedingly difficult for an unauthorized individual to recover the information without knowledge of the specific partitioning scheme.

Keywords: Healthcare Systems; Cover Image (Cov_img); Multiple Steganographic; Medical Data Security;

INTRODUCTION

Steganography is the art of concealing secret messages by embedding them within seemingly innocuous carriers like text, images, audio, or video, thereby preventing detection by unintended recipients. Various techniques are typically employed, including steganography in documents (e.g., PDF), Steganography for videos (MP4, AVI, VOB, MPEG-2, etc.), audio files (MP3, WMA, WAV, etc.), and leveraging free space in files, executables, or operating systems. Image steganography continues to be the leading method in modern steganography, despite its widespread use[1]. Different kinds of images, including uncompressed BMP files, compressed PNG files, and JPEG files, can all be used for steganography. Various steganography methods exist, including those that operate in the spatial and transform domains, each with unique advantages and disadvantages. By systematically replacing the least significant bits, nine bits can be discreetly embedded per pixel. Improving the security of a file can be accomplished by dividing it into several image files[2]. The main disadvantages of using a single Cov_img in steganography are its very limited embedding capability and low level of security. An authentication key improves security for a distinct approach to distributing payloads through the use of image metadata[3]. The document will succinctly cover the techniques and processes, along with the support for multimedia elements utilizing JPEG images featuring 256 values in each Red, Green, and Blue channel, resulting in a depth of 24 bits. Methods and processes will be explained with a text message accompanied by 256-bit JPEG images (equivalent to 24 bits). Due to their limited capacity, images with fewer than 16 bits per pixel are difficult to steganograph. The most common medium for transferring data is high-quality color images with an 8-megapixel resolution and a bit depth of 16 to 24 bits per pixel[4]. The following sections detail the structure of this paper. Section 2 lays out the fundamental principles, followed by a description of the design and architecture in Section 3. Section 4 is dedicated to the implementation details, while Section 5 presents and analyzes the outcomes. The paper concludes with a summary of findings and concluding remarks.

LITERATURE REVIEW

Deep learning-based steganography is gaining prominence over traditional methods, as noted by Sultan et al. [5], due to its enhanced invisibility, capacity, and security. This transition reflects the effectiveness of profound steganographic models. To develop a deep learning steganography model for handling multiple datasets, this paper uses a popular deep learning model known as deep convolutional generative adversarial networks (DCGAN). A single Cov_img can conceal two messages intended for different recipients. The proposed model is comprised of

four networks: Extractor1, Steganalyzer Extractor1, Generator, and Extractor2. Two hidden messages are obscured by the Generator within one Cov_img, utilizing two distinct extractors. The steganalyzer can distinguish between stego and Cov_imgs created by the generator. The experiment utilized the CelebA dataset. Multiple tests have demonstrated that the proposed payload distribution methods enhance security. According to M. Shrivastava et al. [6], the practice of concealing private data within an image is known as image steganography to prevent hackers from finding it. With rising security vulnerabilities a major concern, researchers are actively developing methods for secure data transmission. These methods aim to ensure that only the intended recipient can access the information, building upon the vital tactic of securely storing sensitive data. Ever since then, scholars have developed different methods, like steganography, to address the continuous exchange of information. This study introduces two techniques for hiding information in images. Information bits are initially saved in the Least Significant Bit. Due to the widespread familiarity with the technique, the attacker can easily expose the information, thus making image steganography susceptible. To enhance information security, the second method utilizes RSA principles in conjunction with R-Color Channel encoding. The subsequent bits are utilized to hide RGB pixel values in cases where a red color channel is employed to hide a data bit. This paper evaluates the effectiveness of image steganography, focusing on the efficiency of the widely used LSB and RSA algorithms. G. Benedict et al. [7] indicate that steganography entails concealing a secret message within a regular message and retrieving it at the designated location. There are now many secure methods for steganography available, including image steganography. Traditional steganography techniques include hiding information within a Cov_img. Information has been hidden within multiple images using batch steganography. This research proposes cutting up sensitive data and saving it across multiple images used as cover photos. Secret information can also be retrieved from the recipient's Cov_imgs. Due to data slicing, hackers are unable to decrypt important data securely transferred without access to encryption details.

S. Mukhopadhyay et al. [8], suggest using a coordinated setup of semiconductor lasers for achieving steganography with multiple encrypted single-color images. The selection of the key scheme impacts the robustness of steganography. Chaotic synchronization could be advantageous for steganography in this area. The encryption principle of the new algorithm is analyzed using several statistical tests, and the method for creating Cov_img incorporates a visual representation of disorderly sequences. This new method enhances security by merging LSB substitution with high key space, large embedding capacity, invisibility, and strength of the hidden data. The result is important when thinking about incorporating a multiplexing system that sends multiple images simultaneously. Several Cov_imgs are used by authors A. S. Ansari and team [9] as part of their steganography technique. Different types of images can be handled by a single algorithm, offering multiple benefits. Depending on the size of the image, the speed of the internet, and distortion levels, the image may be selected differently, or security protocols may apply uniformly to all image types. JPEG, Bitmap, TIFF, and PNG Cov_imgs can be analyzed using our approach based on abstract image elements. Steganography has never been applied to various formats of Cov_imgs until now, according to our knowledge. To improve data security and protect sensitive information, a features including capacity pre-estimation, adaptive partition techniques, and distributed data storage was integrated. The proposed technique shows strong performance in robustness evaluations against steganalysis. Additionally, the outcomes of comparing the suggested technique for three distinct types of Cov_imgs are encouraging.

CONCEPTS BASED ON THE PRINCIPLES

A. An overview of image steganography

The majority of Cov_imgs incorporate message bits by employing the Least Significant Bit (LSB) insertion technique. To differentiate between subtly different hues, LSB exploits the limitations of the human visual system. Altering the least significant bit of a pixel's binary representation typically doesn't cause a noticeable change in its color intensity to the human eye. The following pseudocode illustrates a basic steganography algorithm, showing how message bits can be sequentially embedded [10][30].

1. The SecretText contains a collection of text messages.
2. When converting the SecretText into binary, each row corresponds to the message's character count, while each column denotes the bits per character used.
3. Find out the measurements of the SecretText matrix, with X denoting the rows and Y representing the columns, notably with Y being Eight for grayscale images.
4. CoverImg matrix is generated based on Cov_img assessment.
5. 5) For x equal-to One to X
6. For y equal-to One to Y
   a) *Ensure that the final important bit of each byte in Cov-Img(x, y) is cleared to zero by utilizing the bitwise AND operator.*
   b) *By performing a bitwise AND operation on Cov-Img(x, y), SecretText(x, y) can be converted smoothly.*
7. End For
8. End For

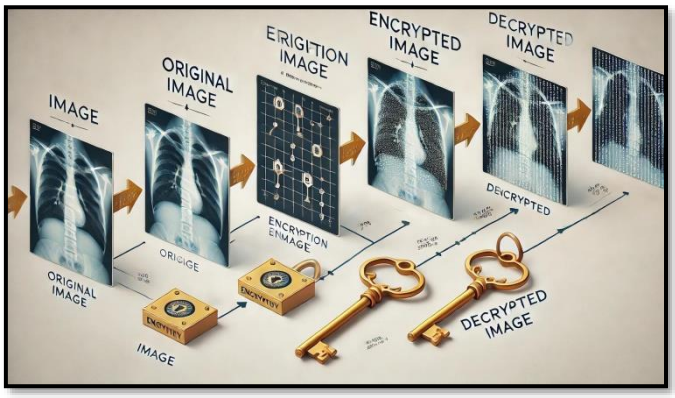
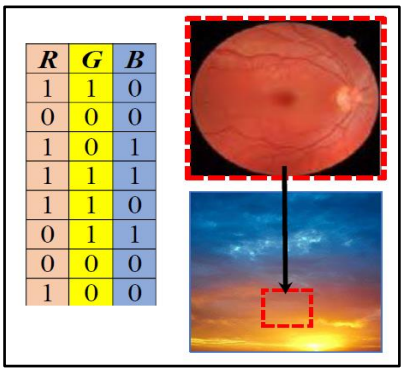


Fig. 1. General Methodology of the Proposed System

B. Analyzing color shifts

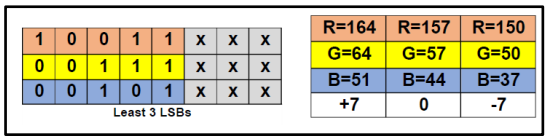
In Figure 2, the 3 least significant bits point towards a pixel embedding capacity of 9 bits. Adjusting the red, green, and blue color values of a pixel by 7 units will produce a new value that is either 7 units higher or lower than the current value. An intruder, typically unnoticed, is unable to detect this subtle pixel change. The algorithm for color shift is simply this: Each 8-bit value is shifted to the left by three, followed by applying the bitwise AND operator to the outcome and the hexadecimal value 0xF0. The preliminary result is now completely established with the incorporation of the payload file's binary information[11][12].

In Figure 2, each encrypted color pixel is plotted against its maximum color shift, albeit quite subtle and not readily apparent. Comparing the color change to the original image is the only method to determine whether it makes sense. The problem only applies to publicly available photos. This issue only affects publicly available photos[13][31].



(a) (b)

Fig. 2. (a) One pixel is highlighted in a 24-bit depth color image (b) Eight-by-three-bit color representation.



(a) (b)

Fig. 3. Shows, (a) Illustration of bits highlighting the Least Significant Bit on a hexadecimal chart, (b) The color shifts of Red, Green, and Blue with a negative or positive value of +7/-7

ARCHITECTURE AND DESIGN

A. Compression of the Payload

An image file is used as a payload for a malicious program. The payload is hidden within an image file. Files such as these can contain various file types, such as audio, video, images, and text[16]. Various file types can be included in these files. Files of this type can be compressed using ZIP's compression algorithm, which has been widely accepted worldwide as a standard compression algorithm. Using ZIP archives, you can store multiple files and directories, and you can use any lossless compression algorithm that you prefer to utilize[17][32]. By analyzing meta-data embedded in compressed directories and files without having to uncompress them, ZIP files can provide better embedding capacity predictions, as well as other benefits[18].

B. Formating Header

Although the decoding process necessitates the images to be in a specific order, the image files for encoding can be arranged in any random sequence. In addition, the payload does not need to utilize every pixel in the image, as this is dictated by its storage capacity[19]. As a result, it is advisable to exclusively decrypt the pixels that have been impacted, which will lead to a considerable decrease in time complexity while decrypting. Figure 3 details how header information, such as sequence number and maximum (x, y) values, is added to each image during encoding [3][20].

C. Techniques for distributing bits

Envision having N images saved in a folder with M encoded images present as well. To reveal the concealed files, the intruder would have to attempt decrypting them N factorial times. Therefore, deciphering encrypted files stored in concealed folders is expected to require a substantial duration, specifically following a complexity of O(n!) for every image, with 'n' denoting the quantity of images to be decrypted simultaneously[8][21]. Nevertheless, the hacker could potentially overcome the issue by utilizing supercomputers to identify any discernible patterns in the distribution of the bits. The proposal can be further improved by incorporating image hashing to bolster the security of the suggested solution. One can distribute bits in various ways, including the following options[22][33]:

1) Sequential Hashing:

This technique includes the retrieval and step-by-step allocation of binary data from a compressed payload file, in line with the principles of Mass Steganography. During the previous conversation, it was pointed out that the storage pattern is thorough, allowing for a more precise prediction of whether data is concealed within an image file without needing to scrutinize the entire file extensively[16][23].

2) Image Hashing Improvement:

Bits within a compressed payload file may be distributed haphazardly, presenting a difficulty in deciphering the slicing patterns. Slicing patterns can be meticulously examined with a considerable delay in this approach. Through this approach, one can examine slicing patterns with a notable delay. Consequently, the intruder must compare the value of each pixel with the value of every pixel in all other images[7][24]. A cutting-edge image hashing algorithm combines an exponential random bit distribution, starting with a broad acceptance of variations that slowly narrows down as the process advances. Tolerance embraces the wide variety of options available when selecting a candidate. Random value generators incorporate the dimensions and resolution of images. New values are created with each cycle. By using fixed metadata, this value stays stable across time. Whenever it operates, it reliably produces a uniform series of values[25][29]. A weight analyzer measures the bit count in each image and suggests the best image to use according to this assessment. Weight analyzers carefully examine each pixel included in the Cov_imgs. Usually, the Cov_img with the lightest weight is the one that tends to be chosen most often when distributing a particular data stream. The value displayed on the counter increases with each pixel that gets filled in the image. The weight analyzer saves the specified segments from payload files into their respective image files[11][26].

3) Password-based image hashing:

A hashing process causes an image's bits distribution to become random, which prevents the recognition of any discernible patterns. Decryption is characterized by a time complexity that is double exponential, reducing the possibility that unauthorized persons can decrypt the data. A random value generator uses this string as a seed value after the key has been hashed[12][27]. Upon seeing a folder with a total of N photos, the trespasser is supposed to perform N factorial particular activities. Once N reaches 1000, iterations increase to roughly 4 times 10 to the power of 249 per iteration. A hacker must attempt all four times the four billion possible combinations of the 12-character password. There has been a significant increase in computational overhead[28][34].

1. Implementation Stage

Our prototype example demonstrates an image steganography technique on a collection of sensitive image files that need to be transmitted in a confidential manner using encoding and decoding processes. The algorithm has been implemented in a Java-based application. The study described in this paper focused on the following projects.

A. Medical Data Encoding

Two file formats must be chosen for the encoding process. The first kind of file that hides confidential information is an image file. The term "cover file" is frequently used to describe this kind of file. Last but not least, payload files need to be encoded inside image files. ZIP files are used to archive data with payloads. Bits are allocated to each image using a hashing algorithm.[29] The payload file's segments are handled either sequentially (3–9 bits) or bit-by-bit. The value generator first generates values with a high degree of unpredictability since the created values are very different from those that were previously generated. The weight analyzer module then uses the predefined value that was previously identified to do the assigned task. As payload data is embedded within images, the weight analyzer module's counter values cause a reduction in the value generator's tolerance. The output filenames are user-definable [6][30]. A series number is automatically included in the filename by default. The camouflage feature is also available simultaneously with the cover files. Capacity plays a pivotal role in concealing information in a cover file. A payload size that has already been used is also displayed[31].

B. Medical Data Decoding

Depending on how the cover files are encoded, they can be decoded in any order. The index number is stored in the appropriate cover file, regardless of how the images were selected. An empty directory might be created if an excessive number of cover files or multiple non-cover files are selected[32]. Decoding and encoding are similar processes. Value generators and weight analyzers are analogous to encoding processes. Bits from each pixel are stored in a file stream instead of being transmitted as part of the payload. The decoding pointer, as specified in the header, is used to extract all decoded payload files[33].

C. User Interface-Design Enhancement

The job is easier to do with a graphical user interface. A number of the functional tools previously mentioned have been updated to improve the user experience[34].

FINDING AND DISCUSSION

Pixels featuring a 24-bit resolution can hold as much as nine bits of information. The camouflage capability of the image has been assessed to be 26MB, determined from a resolution of 4000x6000 pixels and 9 bits per pixel. The formats of images applicable for comparison includes; PNG, BMP, JPEG, JPG, and GIF. Table 1 examines the encoding of 24-bit depth color files, showing how different file formats achieve varying degrees of file size reduction compared to the original Lenna image. The encoding time for each format is also provided. In the simulation, evaluating performance when embedding payloads into images depends on the time required to encode each image in various file formats. For smaller payload files, GIF is favored due to its small size that leads to the elimination of excess overhead. Due to the lengthy encoding process, GIFs are not suitable for managing large payloads. JPEG/JPG is particularly ideal for creating Cov_imgs; nonetheless, all the specified file formats are very efficient. Currently, the application does not accommodate file formats that use wide gamut internal color spaces, such as RAW, ORF, DNG, and TIFF. This limitation restricts the outcome to file formats that have a 24-bit color depth. An image with 4000 pixels tall and 6000 pixels wide is likely to have an effective camouflage effect if its pixel density is high. There are nine bits of information per 24-bit pixel. Due to the image's 4000×6000 pixel resolution and 9 bits per pixel, the camouflage ability of the image is 26MB.

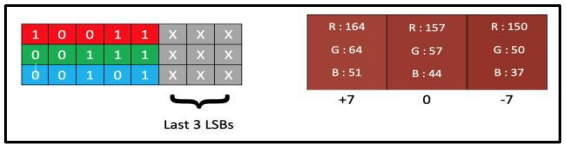


Fig. 4. The Bit Representation

TABLE I, Enhancing the encoding efficiency across various file formats

File Formats (24-bit color depth)	Simulation Parameters		
	Kilobyte size of medical image	Encoding medical images in nanoseconds	Size after sto-ring (in KB)
BMP_Med_Img	485	1.04E+08	485
JPEG_Med_Img	31	1.06E+08	32
PNG_Med_Img	323	1.6E+08	430
TIFF_Med_Img	134	1.67E+08	84

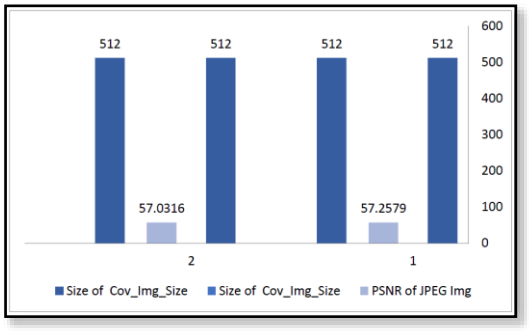


Fig. 5. An encoding speed graph for 24-bit color space file formats

Table 5, The performance measurement parameters of the proposed methodology.

Img_No	Size of Cov_Img_Size	Size of Data Length	PSNR of JPEG Img	PSNR of BMP Img	PSNR of PNG Img	PSNR of TIFF Img
1	512×512	35160	57.2579	68.30	66.67	67.02
2	512×512	35160	57.0316	70.09	67.16	68.71
3	512×512	35160	57.2975	71.40	69.45	71.40
4	200×200	55135	55.98	64.62	63.51	64.12
5	800×800	55135	59.67	63.20	62.90	63.09
6	750×750	44832	59.94	63.62	62.82	62.91
7	560×560	48504	56.10	63.09	62.09	63.01
8	450×450	50134	57.92	58.47	57.89	58.08
9	653×621	36020	55.77	66.28	61.93	63.32
10	700×700	45832	59.76	68.88	67.02	68.62

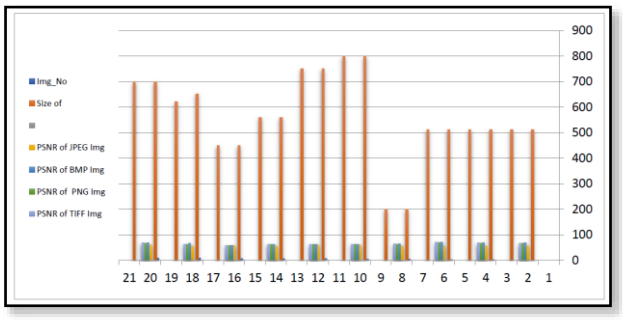


Fig. 6. Comparison of encoding speed for 24-bit color file formats

CONCLUSION

A Cov_img is concealed with the Least Significant Bit using this study. The confidential data bits are then re-encoded onto the pixels. Simulated data is recovered from Cov_img files without encountering any errors by using a Java application. The application is also being developed with an advanced user interface. With this new method, bits are randomly allocated to multiple images, metadata is considered, and advanced image hashing is applied, rendering the pattern of slicing unintelligible. Data bits are scrambled randomly, resulting in pixel bits that are

unintelligible and thus impossible to decipher. In addition to the camouflage properties of cover files, video and audio files can also be used for steganography. Gradual improvements can also be made to image hashing techniques. An application can be enhanced with drag-and-drop functionality after development. A comprehensive internal color space needs to be developed for compatibility.

REFERENCES

[1] Al-Haj A, Mohammad A, Amer A (2017) Crypto-watermarking of transmitted medical images. J Digit Imaging 30(1):26–38. https://doi.org/10.1007/s10278-016-9901-1

[2] Kasim Ö (2022) Secure medical image encryption with Walsh—Hadamard transform and lightweight cryptography algorithm. Med Biol Eng Comput 1585–1594. https://doi.org/10.1007/s11517-022-02565-5

[3] Magdy M, Hosny KM, Ghali NI, Ghoniemy S (2022) Security of medical images for telemedicine: a systematic review. Multimedia Tools and Applications

[4] Benrhouma O (2022) Cryptanalysis and improvement of a semi-fragile watermarking technique for tamper detection and recovery.

[5] B. Sultan and M. A. Wani, "Multi-data Image Steganography using Generative Adversarial Networks," 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2022, pp. 454-459, doi: 10.23919/INDIACom54597.2022.9763273.

[6] M. Srivastava, P. Dixit and S. Srivastava, "Data Hiding using Image Steganography," 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2023, pp. 1-6, doi: 10.1109/ISCON57294.2023.10112069.

[7] A. G. Benedict, "Improved File Security System Using Multiple Image Steganography," 2019 International Conference on Data Science and Communication (IconDSC), Bangalore, India, 2019, pp. 1-5, doi: 10.1109/IconDSC.2019.8816946.

[8] S. Mukhopadhyay and H. Leung, "Multi Image Encryption and Steganography Based on Synchronization of Chaotic Lasers," 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, 2013, pp. 4403-4408, doi: 10.1109/SMC.2013.751.

[9] A. S. Ansari, M. S. Mohammadi and M. T. Parvez, "A Multiple-Format Steganography Algorithm for Color Images," in IEEE Access, vol. 8, pp. 83926-83939, 2020, doi: 10.1109/ACCESS.2020.2991130.

[10] Borra S, Thanki R (2020) Crypto-watermarking scheme for tamper detection of medical images. Comput Methods Biomech Biomed Eng Imaging Vis 8(4):345–355. https://doi.org/10.1080/21681163.2019.1595730

[11] Evsutin O, Melman A, Meshcheryakov R (2020) Digital steganography and watermarking for digital images: a review of current research directions. IEEE Access 8:166589–166611. https://doi.org/10.1109/ACCESS.2020.3022779

[12] Kadian P, Arora SM, Arora N (2021) Robust digital watermarking techniques for copyright protection of digital data: a survey. Wirel Pers Commun 118(4):3225–3249. https://doi.org/10.1007/s11277-021-08177-w

[13] Bhalerao S, Ahmad I, Kumar A (2022) “Reversible ECG Watermarking for Ownership Detection, Tamper Localization, and Recovery”, Circuits. Syst Signal Process. https://doi.org/10.1007/s00034-022-02024-4

[14] Fang Y, Liu J, Li J (2022) Robust zero-watermarking algorithm for medical images based on SIFT and Bandelet-DCT, pp 16863–16879

[15] Kumar S, Panna B, Kumar R (2019) Medical image encryption using fractional discrete cosine transform with chaotic function, pp 2517–2533

[16] Kaur J (2017) An adaptive quad tree based transform domain steganography for textual data. In: 2017 international conference on energy, communication, data analytics and soft computing, pp 3194–3199

[17] Thanki R, Borra S, Dwivedi V, Borisagar K (2017) A steganographic approach for secure communication of medical images based on the DCT-SVD and the compressed sensing (CS) theory. Imaging Sci J 00:1–11. https://doi.org/10.1080/13682199.2017.1367129

[18] Arunkumar S, Subramaniyaswamy V, Vijayakumar V, Chilamkurti N, Logesh R (2019) SVD-based robust image steganographic scheme using RIWT and DCT for secure transmission of medical images. Measurement. https://doi.org/10.1016/j.measurement.2019.02.069

[19] Thabit R (2021) Review of medical image authentication techniques and their recent trends. Multimed. Tools Appl. 80(9):13439–13473. https://doi.org/10.1007/s11042-020-10421-7

[20] Kaur S, Singh S, Kaur M, Lee HN (2022) A systematic review of computational image steganography approaches. Arch Comput Methods Eng 0123456789 (2022). https://doi.org/10.1007/s11831-022-09749-0

[21] Dhawan S, Chakraborty C, Frnda J, Gupta R, Rana AK, Pani SK (2021) SSII: secured and high-quality steganography using intelligent hybrid optimization algorithms for IoT. IEEE Access 9:87563–87578. https://doi.org/10.1109/ACCESS.2021.3089357

[22] Balasamy SSVSK (2022) A review on deep learning in medical image analysis. Int J Multimed Inf Retr 11(1):19–38. https://doi.org/10.1007/s13735-021-00218-1

[23] Muhammad S, Muhammad A, Adnan M, Muhammad Q, Majdi A, Khan MK (2018) Medical image analysis using convolutional neural networks: a review, pp 1–13

[24] Journal P (2018) Noise issues prevailing in various types of medical images, vol 11, p. 1227–1237

[25] Seh AH et al (2020) Healthcare data breaches: insights and implications. Healthc 8(2):1–18. https://doi.org/10.3390/healthcare8020133

[26] Saraswat D, Chaurasia BK (2013) AHP based trust model in VANETs. In: The IEEE 5th international conference on computational intelligence and communication networks (CICN2013), Mathura, India, pp 391–393. ISBN No: 978-0-7695-5069-5. https://doi.org/10.1109/CICN.2013.86

[27] Sharma K, Soni S, Chaurasia BK (2014) Reputation and trust computation in VANETs. In: International conference on electronics engineering and computer science (IEMCON2014) organized by Elsevier, Kolkata, India, ISBN No: 9789351072485

[28] Singh MP, Rai A, Chaurasia PK (2020) Study and analysis of digital watermarking for medical images. Invertis J Sci Technol 13(1):1–7

[29] Chaurasia PK, Tiwari SK, Ansar SA, Yadav N, Soni N, Singh S (2022) The security of transforming digital medical image using RDWT, HT, and SVD techniques. Harbin Gongye Daxue Xuebao/J Harbin Inst Technol 54(10):270–276

[30] Kumar S, Srivastava A, Chaurasiya PK, Kushawaha A, Vishal V (2022) DCT and SVD-based watermarking technique for imperceptibility and robustness of medical images. In: 2022 4th international conference on advances in computing, communication control and networking (ICAC3N), pp 2335–2339. IEEE.

[31] Abdulbaqi, A. S., Obaid, A. J., & Mohammed, A. H. (2021). ECG signals recruitment to implement a new technique for medical image encryption. Journal of Discrete Mathematical Sciences and Cryptography, 24(6), 1663-1673

[32] Mahmood, S. D., Drira, F., Mahdi, H. F., Aribi, Y., & Alimi, A. M. (2023, October). Chaotic Model-Based Blind Watermarking with LSB Technique for Digital Fundus Image Authentication. In 2023 International Conference on Cyberworlds (CW) (pp. 395-402). IEEE.

[33] Al-Rubbiay, F. H., Youssef, A. Y., & Mahmood, S. D. (2023, March). Medical Image Authentication and Restoration Based on mCloud Computing: Towards Reliant Medical Digitization Era. In Doctoral Symposium on Computational Intelligence (pp. 487-500). Singapore: Springer Nature Singapore.

[34] Abdalla, I. H., & Yaser, R. F. (2023). WSN recruitments for encrypted medical image transmission securely. Journal of Discrete Mathematical Sciences & Cryptography, 26(7), 1981–1990. https://doi.org/10.47974/jdmsc-1838.