ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20240408001453
Original scientific paper

Efficient Data Security Using Differential Expansion & Metamorphic Cryptography

Gopika Rajan J.*, R. S. Ganesh

Abstract: With the advancement in information technology confidential data security and privacy in data transmission have long been a top priority because of the increase in cybercrime. A number of cryptographic and steganography methods were put up to resolve the issues. Reversible data hiding, due to its versatility and ability to be used in a variety of settings, is a great strategy for protecting sensitive information. However, it can handle only little payloads with good quality image carrying data.This paper introduces a novel spatial image steganography scheme that combines Differential Expansion (DE) and RC4 cryptography to significantly improve data security. The methodology involves embedding sensitive information into the difference values of pixel pairs using DE, followed by RC4 encryption to ensure robust data protection. Our approach leverages the strengths of DE for high-capacity data embedding and the robust encryption capabilities of RC4 to protect against unauthorized access and tampering.To evaluate the effectiveness of our proposed method, we conducted extensive experiments using a variety of image datasets. The results demonstrate that our method achieves significant improvements in payload capacity, security, and image quality compared to traditional techniques. Specifically, our method outperforms existing methods such as LSB, Hamming code, and GAN-based approaches in key metrics including Peak Signal-to-Noise Ratio (*PSNR*), Mean Square Error (*MSE*), and Structural Similarity Index Metric (SSIM). These metrics indicate that our method maintains high image quality while providing enhanced data security.

Keywords: data hiding; data security; differential expansion; *PSNR*; RC4 cryptography; steganography

1 INTRODUCTION

Since ancient times, people have been concerned in hiding concealed messages. This objective is achieved by both steganography and cryptography, although in different ways. In order to convey a secret message while hiding its presence, steganography is used. On the other hand, encryption, a technique used in cryptography, aims to conceal the content of a message rather than its mere presence. Greek word kryptos, which means "hidden," is the source of the word "cryptography." A message is routinely encrypted before being concealed in a message to obtain a better level of secrecy [1]. A cover that appears innocent, like a digital image file. The necessity for steganography is clear, but less so for additional methods. Simple steganographic methods are readily detectable, and the area of steganalysis is dedicated to combating them. Steganography is a discipline that is continually changing since improvements in steganalysis frequently counter those made in steganography.

Since digital images are used as a cover by the majority of steganographic systems, the whole discipline has taken techniques and concepts from the closely related field. Digital audio and video can be modified for copyright purposes using watermarking and fingerprinting techniques. Although many features of pictures can theoretically be altered, most stego systems strive to maintain the visual integrity of the image. Making modifications that could not be seen with the naked eye was the original aim of stego systems. Because statistical techniques may identify changes in the image even when they are not apparent, this characteristic is insufficient. The output of steganography is frequently reliant on the compression strategy employed; hence image compression is very crucial. Steganalysis techniques have foiled steganographers' efforts to develop more effective ways to conceal a message in a cover item [2]. The goal is to conceal the concealed message by embedding it in the cover item while also doing so in other cover objects. We must apply cryptography to decrypt the stego-message. Utilizing steganography and cryptography is the most effective method. Encryption converts data into an unreadable format to prevent illegal access to data without realizing the precise details of the key being used. The original data is reconstructed using a decryption method.

State vector having 256-bytes is established using RC4 encryption technique, which has key length ranging between 1 and 256 bytes. The utilized plain text has no bearing whatsoever on the key stream. Reversible text hiding strategy incorporating Difference Expansion (DE) methodology has been proposed recently. This type of approach computes the differences between adjacent two-pixel values and chooses a subset of differences for DE. Only color pictures with 8 bits per pixel (bpp) are taken into account in this research. The general architecture of system incorporating steganography and cryptography is demonstrated in Fig. 1.

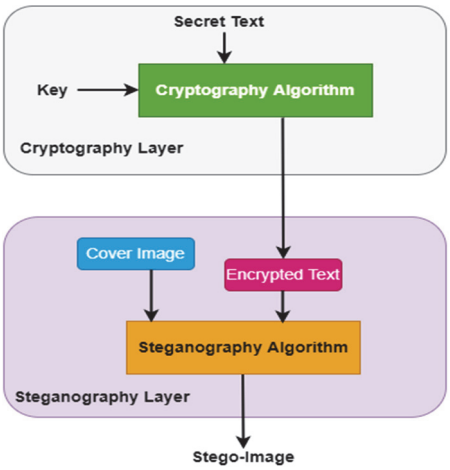

Figure 1 Data hiding framework with cryptography

Despite significant advancements in image steganography, several gaps remain in addressing security and payload capacity effectively. Existing methods often struggle with balancing high payload capacity while maintaining robust security against sophisticated attacks. Techniques such as Least Significant Bit (LSB) embedding and even some advanced cryptographic approaches fail to provide both high security and substantial payload capacity. Furthermore, many existing methods do not

adequately address the degradation of image quality, which is crucial for maintaining the imperceptibility of hidden information. Our research aims to bridge these gaps by integrating Differential Expansion (DE) and RC4 cryptography to enhance both security and payload capacity, offering a more balanced and effective solution. Major objective of this research is to associate steganography with cryptography to assure the security of secret text data transfer. The unique representation of the stego-image is concealed using a powerful encryption method that uses RC4 algorithm. The secret text can be encrypted by using DE before the text concealment phase. The recipient will then get the encrypted text that has been integrated into the chosen difference image. For recovering the original text, the stego image is applied with decryption methods.

2 RELATED WORKS

In order to hide patient data, Kim et al. [3] employed a hybrid data concealing approach that merged the LSB and Hamming code. The *PSNR* of suggested scheme retained above 50 dB. This approach offered a greater degree of concealment than previous approaches. Tang et al. [4] proposed an effective steganography approach based on fuzzy map that takes ambiguity of image into account. Furthermore, LSB replacement is used to hide data in an improved way based on fuzzy map. Alexan et al. [5] developed a security scheme based on AES-128 encryption. Then in, steganography text is concealed inside images via substitution of LSB. The image is first split into slices, and insertion involves various mathematical arrangements. This scheme generates a CT Engine image with a capacity of 295680, a *PSNR* of 62.66 dB, and an *MSE* of 0.03519. Zhang et al. [6] presented a novel technique based on Generative Adversarial Network (GAN) for improved perceptual quality. Hough Transform along with Two-Fish and LSB was utilized by Abikoye et al. [7] for text hiding. Pooja et al. [8] proposed an image steganography-based cryptography method. Cryptography (DES) techniques are based on making the content of a message garbled to unauthorized people. Ayub et al. [9] introduced data hiding based on edge in DCT transformed image. This scheme incorporated Laplacian filter for edge detection. This steganography scheme surprisingly resulted in compression of stego image

Karakkus et al. [10] proposed optimization-oriented scheme involving pixel similarities. For various cover images, mean *PSNR* is 56.5374 dB respectively. Chaeikar et al. [11] proposed a novel blind statistical method involving the flipping of LSB to detect secret text. Image filtration pixels are categorized as suspicious based on statistical evidence. Ashraf et al. [12] proposed a rule-based fuzzy system to identify relation between neighboring pixels by involving instinctive human perception. IT2 FLS similarity measure is used to identify which pixels in the image have the highest similarity values, and the LSB approach is utilized for embedding. Elmasry et al. [13] proposed a method begins by forming the codeword compression before encrypting by using AES. Fisher-Yates scheme is utilized to choose the next pixel location when embedding encrypted data and header information. To conceal one-byte, different LSB from chosen pixels are used. Ioannidou et al. [14] presented edge-based hybrid detector for choosing the pixels to deliver payload. Ahani et al. [15] discussed sparse represented wavelet transforms to hide text message in color images. *PSNR* is approximated to be around 40 dB with LSB.

Ray et al. [16] employed a CNN based edge detector with deep supervision, to identify the edges. The first step in pre-processing the cover image is to mask the final five bits of each pixel. A grayscale edge map is then produced using the edge detector model. To acquire the notable edge characteristics, the map is transformed into a binary version utilizing both global and adaptive binarization approaches. The purpose of using various binarization strategies is to show how the edge detection method is less susceptible to thresholding methods. Shree et al. [17] presented a simple data security method that makes use mixing process, to yield stego-image. When two layers of protection are required, the adoption of a cryptography method adds exceptional confidentiality to the steganography approach.

3 METHODOLOGY

As network expands, maintaining the security of data transmission, particularly for pictures, has become a top priority. Text message security is a crucial area of study in several areas. Therefore, text concealment algorithms improve the efficiency of broadcast while maintaining security from outside threats. The maximum degree of data integrity, secrecy and security, may thus be attained by the combination of cryptography and steganography. This study combines RC4 stream cypher for cryptography and DE for steganography to attain secrecy of text transmissions. The primary purpose of RC4 is to keep the text undetectable to the hackers even if the image undergoes steganalysis. Pixels are modified using difference image to produce the unique stego image. By utilizing RC4, it is possible to increase secrecy [18]. Proposed text security strategy and its embedding procedure flow is described in Fig. 2.

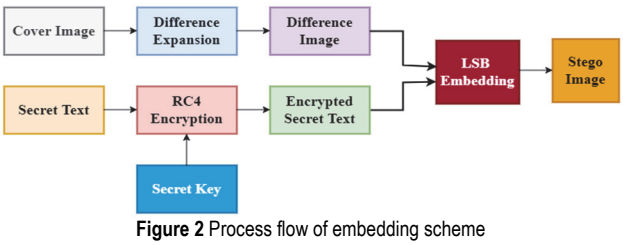

Figure 2 Process flow of embedding scheme

Color images having size of 256 × 256 are initially selected as cover image for steganography. The algorithm is then provided with confidential text message which need to be concealed inside cover image. Secret text is encrypted by utilizing RC4 technique with the inclusion of secret key that increase confidentiality of proposed scheme. The cipher text is covered among the pixels of cover by means of LSB scheme. Text may be sent in an extremely safe manner through a secure communication channel. The attacker cannot access the secret text's original form because of encryption. The method shown in Fig. 3 is utilized to obtain secret text.

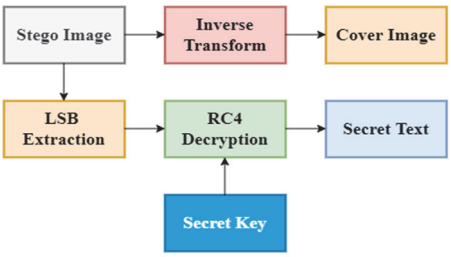

Figure 3 Process flow of extraction technique

3.1 RC4 Encryption

Symmetric cryptography system, known as the RC4 algorithm, mixes plaintext with a keystream that is often used for encryption and decoding. The same keys are used for encryption and decryption in symmetric cyphers. It is made to be simple to use even with a lot of data. Both block mode and stream mode are supported by symmetric cyphers. While in stream mode the message is encrypted from the first character to the last, in block mode the message is divided into multiple set size blocks and encrypted one at a time. [19].

Differential Expansion (DE) calculates the difference between pairs of pixel values in the cover image. These differences are then expanded to create space for embedding secret data. The DE process involves selecting pairs of pixels, calculating their differences, and then embedding the secret bits into these differences. The modified differences are adjusted to maintain image quality. RC4 is a stream cipher known for its simplicity and speed. The encryption process involves two main steps: key scheduling and pseudo-random generation. The key scheduling algorithm (KSA) initializes a permutation of all 256 possible byte values using a variable-length key. The pseudo-random generation algorithm (PRGA) then generates a pseudo-random stream of bytes, which is XORed with the plaintext to produce the ciphertext. In our method, the secret data embedded using DE is encrypted using RC4 to ensure additional security.

The Key Scheduling (KS) algorithm and the Pseudo-Random Generation (PRG) algorithm make up the RC4 algorithm. The methods are displayed in Fig. 1, where l represents the secret key's length in bytes and N represents the array S or S-size box's in words. In RC4, a typical key size ranges from 5 to 32 bytes. RC4 is often utilized in applications with word sizes of $n = 8$ and array sizes of $n = 28$. An identity permutation (0, 1, ..., $N - 1$) is inserted into the array S at the initial stage of RC4 execution. The words in S are then shuffled using a secret key K to initialize S to a randomized permutation. The PRGA generates randomized words from the permutation in S during the second stage of the procedure. The running keystream is made up of one output word from each cycle of the PRGA loop. To create the ciphertext, the plaintext and keystream are bitwise XORed. The steps involved in RC4 encryption scheme are illustrated in Fig. 4.

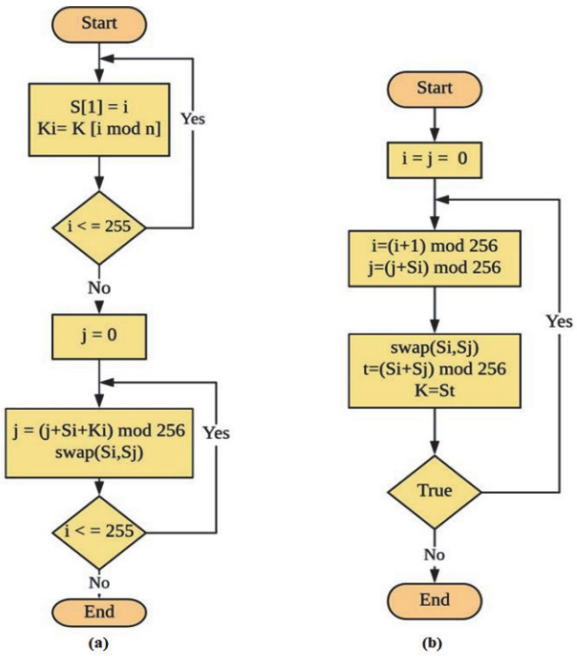

Figure 4 RC4 encryption (a) key scheduling algorithm, (b) prn generation algorithm

Algorithm 1: Key Scheduling
Input: K, m
Output: State (S)
Step 1: Consider $S[i] = i$, for $i = 0 < i < 256$.
Step 2: Initialize $j = 0$.
Step 3: $j = (j + S[i] + K [i \bmod L]) \bmod 256$.
Step 4: $S[i] \leftarrow S[j]$
)

This approach will update any changeable Vd during data embedding by including LSB using DE or changing the LSB [20]. This approach will additionally insert real values of modified LSBs to ensure a precise recovery of the original picture. Therefore, in order to prepare for the decoding procedure, the original h values for the changed Vd must be known. EN2 and CN gather the real values of LSBs using this procedure. The location map might determine the original LSB (1 or 0, respectively) for $h = 1$ or –2, in which case nothing will be gathered. For compression, JBIG2 algorithm will be used to compress the location map losslessly. L stands for the compressed bit stream. P, which stands for payload size, is then added to create B. The Vd matrix contains the bit stream B. Step 5: Output state (S).

Fig. 4 operations are all performed on bytes ($n = 8$), as shown. However, the majority of contemporary CPUs use 32-bit or 64-bit words. The size of array S increases to 232 or 264 bytes, which is impractical, if the size of word in RC4 is extended to $n = 32$ or $n = 64$ to improve speed. To hold all 32-bit or 64-bit permutations, respectively, these array widths should be noted. The KS algorithm is initiated with state "S". The key length (L) can be defined in the range "$1 \leq L \leq 256$". The key is used to initiate the state of "S" box. KS process flow is explained in Algorithm 1.

The procedure is initiated with a key having varying length ranging between 40 and 2048 bits. RC4 generates a stream of key using PRG algorithm. It compensates the criteria and yields a key stream byte. The process of PRG is explained in Algorithm 2.

Algorithm 2: Pseudo Random Generation
Input: State (*S*)
Output: *Kseq*
Step 1: Initialize $j = 0$, $i = 0$.
Step 2: $i = (i + 1) \bmod 256$
Step 3: $j = (j + S[i]) \bmod 256$
Step 4: $S[i] \leftarrow S[j]$
Step 5: $Kseq = S[(S[i] + S[j]) \bmod 256]$
Step 6: Return (*Kseq*)

3.2 Difference Expansion

The following step creates four separate groups of Difference Values (*Vd*). All inflatable $h = 0$ and $h = –1$ are present in EZ. All inflatable *h* values outside of EZ are contained in EN. All alterable *h* values that do not belong to are included in CN (EZ and EN). NC is made up of all immutable *h* values. For DE, all *h* values in EZ are always to be chosen. Depending on the payload size, EN will choose several *Vd* for DE. For ease of use, EN1 and EN2 are used to designate the subsets differences which are selected and non-selected. Location map consists of bitmap having single bit value. The location map assigns a value "1" to a *h* in EZ or ENI while assigning a value "0" to a *h* in NC, CN or EN2. Details are easily seen in Tab. 1

Table 1 Difference embedding

Criteria	Initial Set	Initial Value	Location Map	Updated Value	Updated Set
Modifiable	EN1 or EZ	*h*	1	$2h + b$	CH
Modifiable	CN or EN2	*h*	0	$h + b$	CH
Non-Modifiable	NC	*h*	0	*h*	NC

This approach will update any changeable Vd during data embedding by including LSB using DE or changing the LSB [20]. This approach will additionally insert real values of modified LSBs to ensure a precise recovery of the original picture. Therefore, in order to prepare for the decoding procedure, the original *h* values for the changed Vd must be known. EN2 and CN gathers the real values of LSBs using this procedure. The location map might determine the original LSB (1 or 0, respectively) for $h = 1$ or –2, in which case nothing will be gathered. For compression, JBIG2 algorithm will be used to compress the location map losslessly. *L* stands for the compressed bit stream. *P*, which stands for payload size, is then added to create *B*. The *Vd* matrix contains the bit stream *B*.

Only modifiable *Vd* are altered. All numerical averages and non-changing differentials remain the same. A new LSB is either incorporated via DE for a changing difference or its existing LSB is swapped out. As a result, all embedded data is stored in the LSBs of changing *Vd* after embedding. The decoder is capable of obtaining the embedded bit stream *B* by gathering the LSBs of changed *Vd*. After receiving bit stream *B*, decoder may accurately restore original picture. Fig. 5 provides a good illustration of this text embedding approach involving DE methodology.

B is obtained by gathering the LSBs of all changing *Vd*, as revealed in Tab. 1. Payload, original LSBs stream *C*, and the position map stream *L* may all be decoded from stream *B*. Each enlarged *Vd* and changing *Vd* 's location is indicated on the location map. The original values of other changeable *Vd* may be obtained from the bit stream *C*, whereas the original LSBs of enlarged difference values can be obtained by dividing the result by 2. And nothing has occurred to the non-modifiable *Vd*. The original picture may be precisely reconstructed once all modifiable *Vd* have been restored. The decoding and authentication processes share data of embedding procedure and follow a related workflow. There are five phases in the extraction process. The computation of average and *Vd* comes first. The result is the creation of two distinct groups of differences, NC and CH. The next step is to compile all of the *Vd* in CH's LSBs and create *B*. The map of location is decoded using *B* in JBIG2 decoding algorithm.

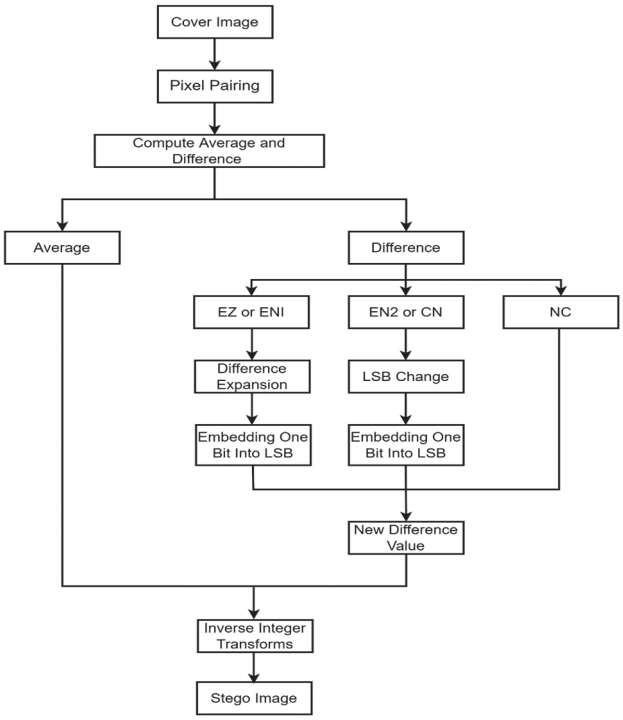

Figure 5 Text Embedding Process Flow

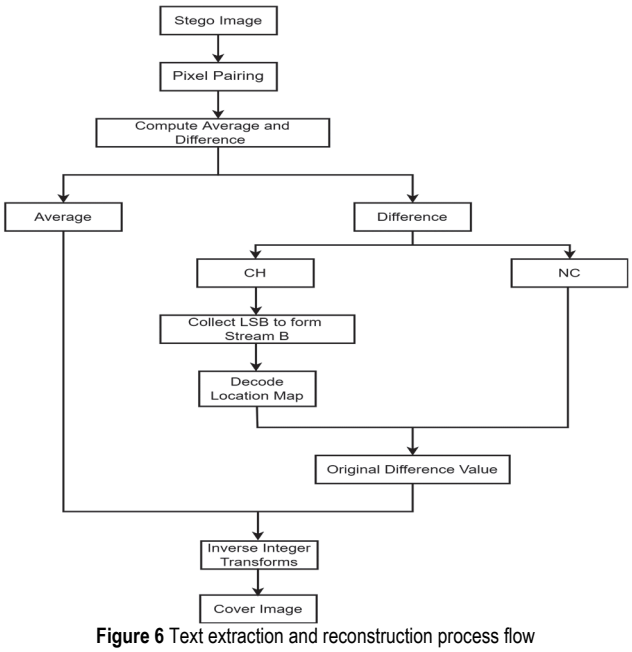

Figure 6 Text extraction and reconstruction process flow

The original picture is then restored using matrix *h*. Fig. 6 displays the flowchart of the procedure. Reversible

integer transformations can be used to return to the original picture after the original *Vd* and mean from the first stage have been obtained. This method's ability to recreate the original cover art is a key selling point. This enables the user to use the cover image without any loss in clarity.

4 RESULTS AND DISCUSSION

As explained in the methodology, the plain text is initially encrypted for the increment in security and the steganography algorithm is applied on the encrypted text. MATLAB 2018b is used to develop the proposed secure steganography algorithm. Initially, a text is selected and is read by the algorithm. Then RC4 encryption scheme having key size of 256 bits is used to encrypt the given plain text. A standard cover image is selected, and suggested DE scheme has been utilized to embed bit stream into the pixels. In order to incorporate encrypted text within the preferred cover and create a stego image, we employed DE scheme. Through a secured link between the transmitter and receiver, the resultant image is subsequently transmitted to suitable receiver. Fig. 7 presents the input text, encrypted text, cover and stego image from the developed security system. Secret text that needs to be encrypted send after embedding is represented in Fig. 7a. The encrypted secret text file obtained from the RC4 algorithm is shown in Fig. 7b and cover image is depicted in Fig. 7c. The crypto-stego integrated output is given in Fig. 7d.

Figure 7 Inputs and results (a) secret text, (b) encrypted secret text, (c) cover image, (d) stego image

Most desirable properties of a secure steganography system are nonrepudiation and capacity. In order for computer modelling to be possible, the embedded data must be unnoticeable to the viewer. Standard images (Peppers, Barbara, Baboon and Lena) are taken for evaluating the performance. Performance parameters include Mean Square Error (*MSE*), Peak Signal-to-Noise Ratio (*PSNR*) and Structural Similarity Measure (SSIM).

PSNR can be indicated as a quality measure that considers the variation in original cover and stego images. It can be cmputed as follows:

$$PSNR = 10\log\left(\frac{MAX_I^2}{MSE}\right) \tag{1}$$

where , MAX_I is the maximum possible pixel value of the image. Likewise , the *MSE* can be computed as follws:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[I(i,j) - K(i,j)\right]^2 \tag{2}$$

SSIM assesses the similarity between the original and stego images, considering luminance, contrast, and structure. SSIM values range from –1 to 1, with higher values indicating greater similarity and better image quality. In *PSNR*'s case, there is a decrement with increase in embedding rate. So, it can be showed that *PSNR* is inversely proportional to embedding rate. More number of pixels are modified because of text and visual attributes of stego decrease during the rise in embedding rate. Embedding rate is changed between 0.1 and 1 with 0.1 interval. The variation in *PSNR* according to embedding rate is exemplified in Fig. 8.

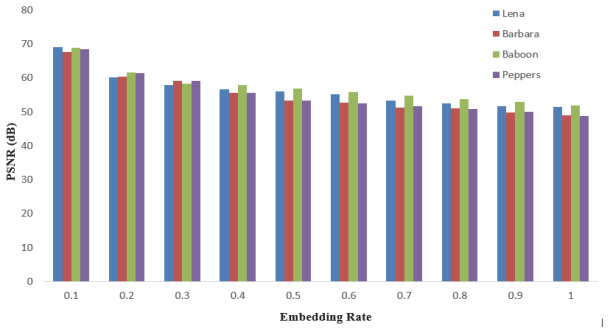

Figure 8 Variation in *PSNR* with respect to embedding rate

The maximum *PSNR* value is 68.9745 dB, which is obtained for the cover image 'Lena' when the embedding rate is 0.1. The lowermost value of *PSNR* is obtained as 48.8468 dB for the cover image 'Peppers' when the embedding rate is 1. There is 29.41% reduction in *PSNR* when the embedding rate changes from 0.1 to 1. While considering *MSE*, it varies proportionally with embedding rate. So, it can be indicated that *MSE* is directly proportional to the embedding rate. Embedding rate is changed between 0.1 and 1 with 0.1 interval. The variation in *MSE* with respect to embedding rate is illustrated in Fig. 9. Efficient algorithms provide lowest values for *MSE*.

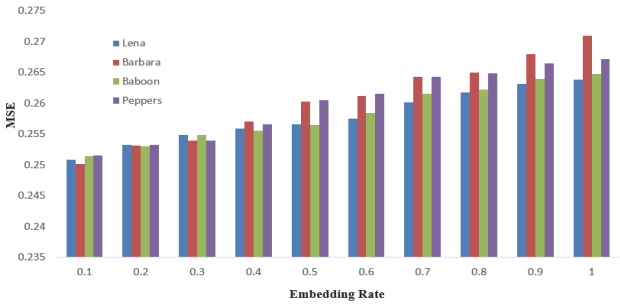

Figure 9 Variation in *MSE* with respect to embedding rate

Lowest *MSE* of 0.02501 is obtained for the cover image 'Barbara', at the embedding rate 0.1. The maximum value of *MSE* is 0.02709 obtained for the cover image 'Lena' when the embedding rate is 1. There is 7% increase in *MSE* when the embedding rate changes from 0.1 to 1. Higher embedding rate can incorporate more text inside the cover image. When SSIM is considered, its value decreases with the increase in embedding rate. So, it can be indicated that SSIM is inversely proportional to the embedding rate.

More pixels are affected by the secret image with increment in embedding rate. The image size does not have any influence on SSIM. Here the internal structural change in the image is analyzed. The variation in the embedding rate is from 0.1 to 1 with an interval of 0.1. There is no constant change in SSIM with the variation in the embedding rate. The variation in SSIM with respect to embedding rate is illustrated in Fig. 10.

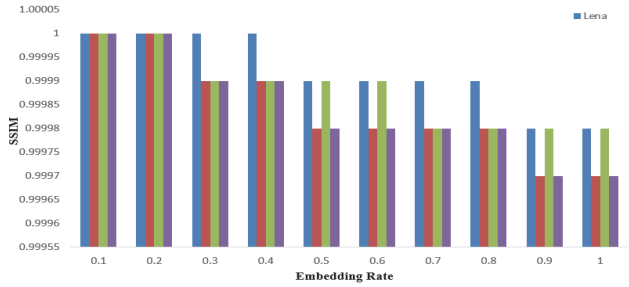

Figure 10 Variation in SSIM with respect to embedding rate

The peak value of SSIM is 1, which is obtained for lower embedding rates. The minimum value of SSIM is 0.9997 obtained for the cover images "Barbara" and "Peppers" when the embedding rate is 1. There is 0.03% decrease in SSIM when the embedding rate changes from 0.1 to 1. This indicates the ability of the proposed algorithm to provide structural stability. It is possible to incorporate more bits of text inside a cover image when the embedding rate is higher. But embedding rate has an adverse effect on the structural stability of cover image. Since there is no huge change in the structure of cover, the algorithm can provide robust results for a wide range of images.

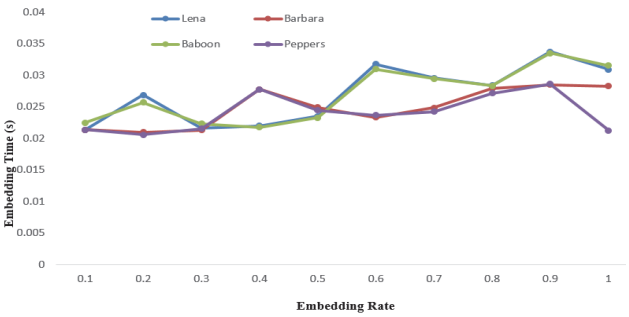

Figure 11 Variation in embedding time with respect to embedding rate

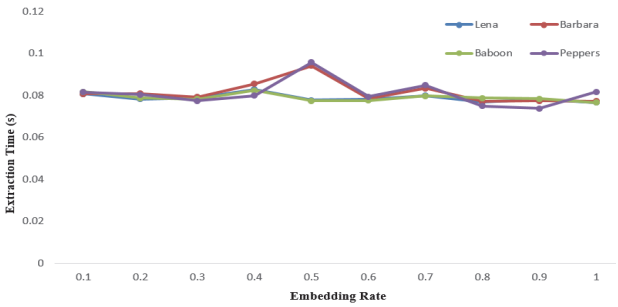

Figure 12 Variation in extraction time with respect to embedding rate

The computation time is another crucial element in the steganography process. The amount of time needed for embedding grows as the embedding rate does. Each and every pixel in stego image is examined for recovering secret encrypted message. Number of pixels that must be evaluated will rise along with the embedding rate, which will lengthen calculation time. Fig. 11 and Fig. 12 show, respectively, the variance in embedding time and extraction time with regard to embedding rate for various cover images.

When comparing the aforementioned figures, it is clear that the embedding time changes in response to the embedding rate. Regarding embedding rate, the extraction time is consistent for different cover pictures. In "Lena" image, embedding takes the longest, which is 0.033738 s at embedding rate 0.9. In "Peppers" image embedding takes the shortest, which is 0.020578 s at embedding rate 0.2. In "Peppers" image extraction takes the longest, which is 0.095796 at 0.5 embedding rate. In "Peppers" image extraction takes the shortest, which is 0.073689 at 0.9 embedding rate. In comparison to embedding time, extraction time is longer. Scanning every pixel is necessary during extraction. The computation time will rise as a result.

This research selects the method from articles which are used for comparison that have chosen 256 × 256 sized cover images. By adjusting the variation between the average of the SLT parameters, Thabit et al. [21] introduced Slantlet Transform (SLT) and embedded text changed overlayed areas of cover image. In the space domain, Kanan et al. [22] proposed GA scheme for embedding text. Adaptive steganography is possible and the stego image quality is adjustable. In addition to chaotic cypher, Sajasi et al. [23] included noise visibility function. The quality of embedding was improved through Genetic Algorithm (GA). Text concealment using 3D sinusoidal chaotic mapping, which is sensitive to beginning conditions and limitations, was proposed by Valandar et al. [24]. RDWT helps to overcome distortions caused by changes in energy allocation brought on by changes in the input signal, according to Subhedar et al. [25] description of the concept. QR factoring also offers reduced processing complexity. Fig. 13 compares these techniques with proposed steganographic scheme that is discussed on the basis of *PSNR*.

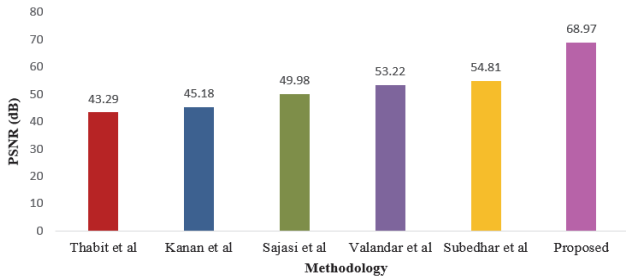

Figure 13 Comparison of *PSNR*

The proposed technique produced a peak *PSNR* of 68.97 dB which indicates high similarity between cover with the stego images. *PSNR* of 54.81 dB were provided by the method suggested by Subedhar et al. [23]. The difference between these two techniques is 14.16 dB. *PSNR* of 43.29 dB were provided by the method suggested by Thabit et al. [21], which is the lowest. When compared to the current techniques, the suggested method produces the highest *PSNR*. The *MSE* is a further factor that must be taken into account when assessing the steganography technique. The suggested approach yields the lowest *MSE*

value and indicates the least amount of error in the stego image, as seen in Fig. 14.

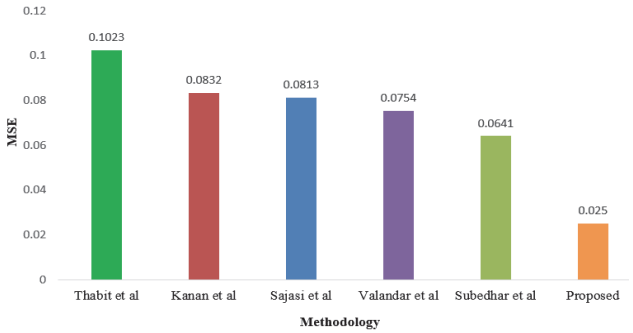

Figure 14 Comparison of *MSE*

As illustrated in Fig. 14, the proposed method provided the lowest error during the embedding process and the *MSE* value is obtained as 0.02501. The method proposed by Subedhar et al. [23] provided *MSE* of 0.0641. There is a variation of 0.0391 between these two methodologies. The proposed method yields best *MSE* (lowest error) when compared to the existing methodologies. Since the error is low and the image quality is high, this method can be incorporated in standard steganography algorithms. This method enables transmission of text messages in a secure way.

5 CONCLUSION

This research adopts a unique spatial image steganography scheme which relies on DE and RC4 cryptography. The DE increases the suggested steganography scheme's security and enhances the key space. Before embedding encrypted text message, cover image pixels are transformed into mean and difference. Analysis using *PSNR* and *MSE* demonstrates the algorithm's irrational nature. In the sequences created by algorithm, SSIM performance is better. DE is performed on cover and it is used for both embedding and extraction. Peak *PSNR*, *MSE*, and SSIM for the proposed approach were 68.97 dB, 0.02501, and 1, respectively. By computing SSIM, *PSNR*, and *MSE*, experimental trial displays good stability. Performance of embedding and extraction operations as well as stego image quality are demonstrated by comparing the developed algorithm to industry benchmarks. While our method shows promising results, it has some limitations, such as increased computational complexity, potential vulnerability to advanced attacks, possible image quality degradation with large payloads, and constrained payload capacity.

Future research should focus on optimizing computational efficiency, enhancing security measures, improving image quality preservation, expanding payload capacity, and conducting real-world application testing to validate the method's robustness and practicality.

6 REFERENCES

[1] Saleh, M. E., Abdelmgeid, A. A., & Omara, F. A. (2016). Data Security Using Cryptography and Steganography Techniques. *International Journal of Advanced Computer Science and Applications(IJACSA), 7*(6), 390-397. https://doi.org/10.14569/IJACSA.2016.070651

[2] Dalal, M. & Juneja, M. (2018). Video Steganography Techniques in Spatial Domain A Survey Proceedings of the International Conference on Computing and Communication Systems. *Lecture Notes in Networks and Systems, Springer Singapore, 24*. https://doi.org/10.1007/978-981-10-6890-4_67

[3] Kim, C., Shin, D., & Kim, B.G., & Yang, C. N. (2018). Secure medical images based on data hiding using a hybrid scheme with the Hamming code, LSB, and OPAP,*Journal of Real-Time Image Processing,* 14(5), 115–126. https://doi.org//10.1007/s11554-017-0674-7

[4] Tang, L., Wu, D., Wang, H., Chen, M., & Xie, J. (2021). An adaptive fuzzy inference approach for color image steganography. *Soft Computing, 25*(16), 10987-11004. https://doi.org/10.1007/s00500-021-05825-y

[5] Alexan, W., Elbeheiry, M., & Islam G. O. (2020). A Comparative Study Among Different Mathematical Sequences in 3D Image Steganography. *International Journal of Computing and Digital Systems, 9*(4), 545. https://doi.org/10.12785/ijcds/090403

[6] Zhang, Y., Qin, C., Zhang, W., Liu, F., & Luo, X.(2018). On the Fault-tolerant Performance for a Class of Robust Image Steganography. *Signal Processing,* 146, 99-111. https://doi.org/10.1016/j.sigpro.2018.01.011

[7] Abikoye, O. C., Ojo, U. A., Bamidele, A. J., & Ogundokun, R. (2020). A safe and secured iris template using steganography and cryptography. *Multimedia Tools and Applications, 79*(15). https://doi.org/10.1007/s11042-020-08971-x

[8] Pooja, R. & Sharma, P. (2016). Cryptography Using Image Steganography. *International Journal of Computer Science and Mobile Computing, 5*(7), 451-456.

[9] Ayub, N. & Selwal, A. (2020). An improved image steganography technique using edge based data hiding in DCT domain. *Journal of Interdisciplinary Mathematics, 23*, 357-366. https://doi.org/10.1080/09720502.2020.1731949

[10] Songul.K. & Avcı, E. (2020). A New Image Steganography Method wıth Optımum Pıxel Sımılarıty for Data Hıdıng ın Medıcal Images. *Medical Hypotheses, 139*(3), 109691. https://doi.org/10.1016/j.mehy.2020.109691

[11] Chaeikar, S. S., Zamani, M., Manaf, A. A., & Zeki, A. (2018). PSW statistical LSB image steganalysis. *Multimedia Tools and Applications, 77*(1). https://doi.org//10.1007/s11042-016-4273-6

[12] Zubair,A., Roy, M. L., Muhuri, P. K., & Lohani, Q. M. D. (2020). Interval type-2 fuzzy logic system based similarity evaluation for image steganography. *Heliyon, 6*(5). https://doi.org/10.1016/j.heliyon.2020.e03771

[13] Kasapbaşı, M. C. & Elmasry, W. (2018). New LSB-based colour image steganography method to enhance the efficiency in payload capacity. *Security and integrity check Sādhanā, 43*(5). https://doi.org/10.1007/s12046-018-0848-4

[14] Anastasia, I., Spyros, H., & George, S. (2012). A novel technique for image steganography based on a high payload method and edge detection. *Expert Systems with Applications, 39*(14), 11517-11524. https://doi.org/10.1016/j.eswa.2012.02.106

[15] Ghaemmaghami, S. & Ahani, S. (2015). Colour image steganography method based on sparse representation. *IET Image Processing, 9*(6). https://doi.org/10.1049/iet-ipr.2014.0351f

[16] Ray, B., Mukhopadhyay, S., Hossain, S., Ghosal, S. K., & Sarkar, R. (2021). Image steganography using deep learning-based edge detection. *Multimedia Tools and Applications, 80*(4), 33475-33503. https://doi.org/10.1007%2Fs11042-021-11177-4

[17] Rashmi,S. & Dinesh, S. (2021). Hybrid Secure Data Transfer Scheme Using Cryptography and Steganography. *Proceedings of the Second International Conference on Information Management and Machine Intelligence, 10*, 577-583. https://doi.org/10.1007/978-981-15-9689-6_62

[18] Kareem, S. M., & Rahma, A. M. S. (2020). A Modification on Key Stream Generator for RC4 Algorithm. *Engineering and Technology Journal, 38*(2B), 54-60. https://doi.org/10.30684/etj.v38i2B.404
[19] Deshmukh, P. R. & Rahangdale, B. (2014). Hash based least significant bit technique for video steganography. *International Journal of Engineering Research and Applications, 4*(1), 44-49.
[20] Hu, Y., Lee, H. K., Chen, K., & Li, J. (2008). Difference Expansion Based Reversible Data Hiding Using Two Embedding Directions. *IEEE Transactions on Multimedia, 10*(8), 1500-1512. https://doi.org/10.1109/TMM.2008.2007341
[21] Thabit, R. & Khoo, B. E. (2014). A new robust lossless data hiding scheme and its application to color medical images. *Digital Signal Processing, 38*, 77-94. https://doi.org/10.1016/j.dsp.2014.12.005
[22] Hamidreza, R. K. & Bahram, N. (2014). A novel image steganography scheme with high embedding capacity and tunable visual image quality based on a genetic algorithm. *Expert Systems with Applications, 41*(14), 6123-6130. https://doi.org/10.1016/j.eswa.2014.04.022
[23] Sajasi, S. & Eftekhari, M. A. M. (2015). An adaptive image steganographic scheme based on Noise Visibility Function and an optimal chaotic based encryption method. *Applied Soft Computing, 30*, 375-389. https://doi.org/10.1016/j.asoc.2015.01.032
[24] Barani,V. M. Y., Ayubi, P., & Aghazadeh, M. (2018). An integer wavelet transform image steganography method based on 3D sine chaotic map. *Multimedia Tools and Applications, 78*, 9971-9989. https://doi.org/10.1007/s11042-018-6584-2
[25] Mansi, S. & Mankar, V. H. (2016). Image steganography using redundant discrete wavelet transform and QR factorization. *Computers & Electrical Engineering, 54*, 406-422. https://doi.org/10.1016/j.compeleceng.2016.04.017

Contact information:

Gopika Rajan J., Research scholar
(Corresponding author)
Department of ECE,
Noorul Islam Centre for Higher Education,
Kumaracoil, Tamil Nadu, India
E-mail: jgopikarajan@myyahoo.com

R. S. Ganesh, PhD Professor
Department of Electronics and Communication Engineering,
PET Engineering College,
Vallioor, Tamilnadu, India
E-mail: ece.ganesh@petengg.ac.in