第47卷第8期
2020年8月

湖南大学学报（自然科学版）
Journal of Hunan University
(Natural Sciences）

Vol. 47. No. 8.
AUGUST 2020

Improving Data Security Using Modified LSB-Based Image Steganography Technique

Maryam Juma AlSinani[1†], Ahmad Kayed[2], Ghaliya Al Farsi[3]

(1,3. Department of Information Technology, Buraimi University College, Al Buraimi, Oman;

2. Department of Information Technology, Sohar University, Sohar, Oman)

(malsenani@buc.edu.om, AKayed@su.edu.om, galfarsi@buc.edu.om)

Abstract: This research improves the Savitha Bhallamudi technique by randomizing the hiding places of bits in the cover image. To enable the main quality characteristics of the image, the following steganography algorithms were used: hiding capacity, perceptual transparency, robustness, and resistance against attack. This method, using the Savitha technique, is based on the least significant bit (LSB) system. The LSB system involves sequentially hiding secret data in the LSB of the image. The main idea of the proposed method is to enhance security by embedding the bits in an image randomly, depending on the value of different bytes of the original (cover) image. This methodology was developed in stages, with the first process hiding data randomly in the images by checking the first byte of the original image only. The second process goes through some bytes in the cover image and hides data based on the value of these bytes. With this new method, results were tested on a sample of ten images. The evaluation of the effectiveness of this method's experimental results was confirmed by using the calculations of two metrics: mean squared error (MSE) and peak signal-to-noise ratio (PSNR). The achievement of better outcomes from the proposed method could improve the results of MSE, provide better visual quality (PSNR), and be used for comparison to the original Savitha method. The new results reveal the improvement of the amount of randomness that can reduce the level of capacity and enhance security (MSE average is 27.8 using the Savitha technique; MSE is 21.1 using the first enhanced method). The new results further illustrate that steganography performed by random bits-hiding is less prone to attack compared to sequence embedding as it is difficult to recognize the pattern used in hiding the secret message.

Keywords: security, steganography, least significant bit, image, randomness

使用改进的基于最低位的图像隐写技术提高数据安全性

摘要：通过随机化掩盖图像中位的隐藏位置，这项研究改进了萨维莎技术。为了实现图像的主要质量特征，使用了以下隐写术算法：隐藏能力，感知透明性，鲁棒性和抵抗攻击性。使用萨维莎技术的此方法基于最低有效位（最低位）系统。最低位系统涉及在图像的最低位中顺序隐藏秘密数据。所提出方法的主要思想是根据原始（覆盖）图像不同字节的值，通过将比特随机嵌入图像中来增强安全性。这种方法是分阶段开发的，第一个过程通过仅检查原始图像的第一个字节来在图像中随机隐藏数据。第二个过程遍历封面图像中的某些字节，并根据这些字节的值隐藏数据。使用这种新方法，可以在十张图像的样本上测试结果。通过使用两个指标的计算，证实了对该方法实验结果的有效性的评估：均方误差（微软）和峰值信噪比（信噪比）。通过提出的方法获得更好的结果可以改善微软的结果，提供更好的视觉质量（信噪比），并且可以与原始的萨维莎方法进行比较。新结果表明，随机性的改进可以降低容量级别并增强安全性（使用萨维莎技术的微软平均值为27.8；使用第一种增强方法的MSE为21.1）。新的结果进一步说明，与随机嵌入相比，通过随机位隐藏执行的隐写术更不容易受到攻击，因为很难识别用于隐藏秘密消息的模式。

关键词：安全性，隐写术，最低有效位，图像，随机性

Received (date):August 1, 2020

About the author: Maryam Juma[1†], Ahmad Kayed[2], Ghaliya Al Farsi[3], 1,3. Department of Information Technology, Buraimi University College, Al Buraimi, Oman; 2. Department of Information Technology, Sohar University, Sohar, Oman

Maryam Juma, E-mail:malsenani@buc.edu.om

Introduction

Steganography techniques have recently drawn the attention of researchers due to their use in concealing messages within mediums without leaving any tracks in the original message. The word steganography, meaning concealed writing, originated from the Greek language [1]. Steganography finds its application in hiding a message in a cover image so that its existence is not noticed [2]. This technique is countered by steganalysis, which is used to explore and estimate hidden messages or communication. A departure from the days when tattoos and invisible inks were used to uncover steganographic content, networking and computer technologies today provide easy channels for steganography [4].

A significant challenge in the internet world is maintaining privacy and anonymity. Information explosion is a boon to the security and data-hiding sector where people desperately need to maintain confidentiality and integrity of information. This demand has moved the scientific community to explore the areas of cryptography, steganography, digital signatures, fingerprinting, and watermarking [3].

Internet science and multimedia technology are growing at a speed beyond imagination; with digitized texts, audio, video, and images being exchanged rapidly in daily life. These are transmitted through a vast network of computers Internet of Things (IoT) and devices, whose security systems are still not up to the expectation and demand of users. The ability of unauthorized users to capture data is a challenge being faced by the network security team, and, steganography and cryptography help to meet it to some extent [5].

Many techniques of steganography were set to hide secret data into cover media. Least Significant Bit (LSB) substitution is a popular algorithm that is used by many techniques. Some of those algorithms are as simple as hiding & Seek algorithm, which is based on sequential hiding that makes it easy for the intruder to uncover the secret data. Randomly hiding the bits will enhance the level of security of these algorithms and it makes it hard to uncover the secret data. This research uses the Savitha algorithm to apply this idea to improve and enhance this algorithm with more security and robustness.

The main objective of this research is to enhance security in the existing technique using the LSB method with notifying the capacity value. Much work has been done in random techniques with specific patterns, and the objective of this work is to use the total pattern on a random basis. This will depend on checking the first values of bits to decide on hiding it or moving it to the next byte.

1 Method

The technique proposed in our research uses the images as cover media. Both cover and message images are read at first to store the size of them to carry out the embedding process later. Then, the embedding process is started and a specific pattern of randomization is chosen as we applied two different techniques that differentiate in the level of randomization. The main changing point between them depends on the statement location, which determines the pattern. In the end, the result of the embedding process is the stego image that contains the secret image and will be extracted in the extracting step to produce the secret message.

1.1 Framework of the Proposed Technique

1. Find appropriate images for testing the proposed method (ex. Benchmark).
2. Adapt the existing algorithm *“Savitha Bhallamudi”* which sequentially hides the secret message.
3. Manipulate the code to make the embedding look random.
4. Test the results by calculating PSNR and MSE with different images to get better comparisons.
5. After results, compare different values of PSNR and MSE from different images, to evaluate and prove the effectiveness of the proposed technique

Figure 1 indicates the main steps of the proposed technique.

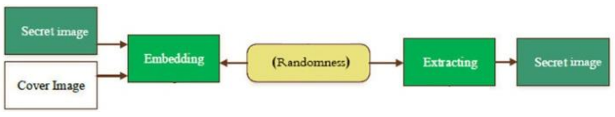


Fig.1 Main steps in proposed technique

1.2 Highlights of the Proposed Technique

- Cover and message images are converted from RGB to grayscale, to reduce the complexity in code (loss of contrast, sharpness, shadow, and structure of the color image).
- The cover and message images are extracted and divided into a new variable applying AND operation with binary numbers, to produce less affected results.
- Both cover and message will be in binary format for matching.
- To achieve randomness, condition statements are based on two levels (image level and image byte level).

- A stego image is the result of an embedding process that is less likely to be detectable by visual imperceptibility and the PSNR metric.

1.3 Embedding Process

In our research of hiding data, we conceal the secret data bits into the least four significant bits (right-most bits), depending on the method used, the first with less randomness or second where more randomness is applied. Images are chosen as cover mediums. Before the embedding process starts, variables are initialized to store the sizes of cover and message image and converted both to grayscale to get rid of any sharpness, shadow, or any other image that might affect the results of the embedding process. Further, the part of the randomness process is initiated by choosing specific values for the new variable responsible for not hiding in some location. These variables identify the capacity values. The variable will contain two values, first, the initialized one and the second one based on the value of byte changed. Later, a checking process will go on in the first randomness technique where less randomization is applied. Only the value of the first byte in the image is checked. If it is more than 128, the variable of not hiding will be changed to another value that leads to not hiding in the first byte of an image. The second enhanced technique then applies the checking process of the value of each byte in the image. So the process is conducted by iteration by going through all the bytes of image size. If the value of the bytes is more than 128, the hiding variable is not changed to another value. Essentially, inside the loop statement, a MOD operation is calculated by taking the value “J” which refers to the matrix index of an image that undergoes constant change with not hide variable. The result is then checked. If the result value is not equal to zero, the hiding process case is applied. Otherwise, the hiding process will be in original bits of the cover image (not hidden).

The embedding process and value checking go through all image sizes until the loop finishes. Then, the four bitplanes of the cover image are extracted to combine the data of the cover and message image acquired in the last step. This process is done by creating a new blank space, with a size equivalent to that of the cover image. Bit operations are applied to combine the cover and message images.

The enhanced technique generally consists of two major phases: embedding the secret message inside the cover image using randomized factors, and extracting the message from the cover media.

1.4 Randomness

Most of the Least Significant Bit techniques involve embedding the secret data within the pixels of an image in a sequenced pattern. The proposed techniques are hiding or not hiding randomly pixels in the image bits, based on the algorithm used. Two steps are added to enhance the hiding process. The first step is the initialization of the variable to hold the values of capacity that indicate the location of not hiding; these can be of any value. The variable then is placed two times. For example, in the first place, a default value of 50 is assigned, then it is changed based on the method used. The second step is applying the Mod function. The two factors used in the Mod formula are the value of the "j" field, which indicates the counter of an image matrix. The second factor is the "no hiding" variable. The function counts and checks the result until the counter ends upon reaching the last row of the image size.

A randomness pattern is applied on two levels during the embedding process: first in the level of the image, and then in the level of the bytes of an image. Both methods embed the secret message bits in a normal (sequential) manner, but in some specific locations not hidden.

The first method uses the randomness pattern by checking only the value of the first byte. If it is higher than the (specific value), the "no hide" variable is changed to a different value, while the second enhanced method does not just check the first value of the byte, However, the checking process runs in a loop to iterate through all image byte values. The "no hide" variable is changed and stored based on that specific value, and each byte has a different value. Therefore, for example, the variable value could be either 50 or 25. After that, the Mod operation occurs to decide between hiding or not hiding, by calculating the Mod of the row's matrix values with the "no hide" variable value. If the results are not equal to 0, the bit hiding process continues, otherwise it ends.

1.5Pseudo Code

1.5.1 Embedding Phase Pseudo Code

```
Embedding Input: Cover image C, secret image M.
Convert cover and message to grayscale.
Calculate the size of C and M.
Initialize condition variable (no hide mode) with value.
If first byte greater than 128 then
Set condition variable to other capacity value.
Duplicate the size of the message and cover to new variables
For row counter to the height of message image
For column counter to the width of message image
If result of Mod (column counter and not hide mode) Not equal to 0
hide normal case
else
do not hide, use the original bit from the cover
```

```
    End
    End
```

1.5.2 Extracting Phase Pseudo Code

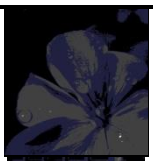
```
Extracting Input: Stego image
Calculate the width and height of cover and message images
Initialize condition variable (no hide mode) with value
If first byte greater than 128 then
Set condition variable to other capacity value
For row counter to the height of message image
For column counter to the width of message image
 If the result of Mod (column counter and not hide mode) Not equal to 0
Extract Eighth-bit blanes of stego image
Initialize message variable with 8-bit blans multiplied with binary values
```

2 Experiments

In this part of the research, we present the experiments that have been done and discuss the results. To measure how efficient our enhanced algorithms are, the algorithms have been tested over a dataset of secret images. Our dataset consists of ten grayscale images, which are gathered from two sources. The first source is the BOSSbase1.01 datasets, which contain the famous images globally used for steganographic algorithms evaluation such as Lyena. The sizes of the images are modified to make them suitable for the Savitha algorithm, as it required the cover message image to be in same size. Secondly, some random images were collected from the Internet.

Since we embed the secret data inside the bytes that consist of the pixels, the number of bytes that can be used depends on the number of pixels in the cover image. Since the algorithm embeds in the spatial domain, cover and message images used the grayscale function to lose any effect of the images. All the images chosen were of the same type, JPG, to make the type of image a fixed factor through the experiments. The images consist of images with dimensions of 449×455. All the cover images are shown below in Table 1.

Table 1 Sample of images used in testing (BOSSbase1.01 dataset)

No	Type	Dimensions (pixel)	Number of Bytes	Image
1	JPG	449 × 455	449 × 455	
2	JPG	449 × 455	449 × 455	
3	JPG	449 × 455	449 × 455	
4	JPG	449 × 455	449 × 455	
5	JPG	449 × 455	449 × 455	
6	JPG	449 × 455	449 × 455	
7	JPG	449 × 455	449 × 455	
8	JPG	449 × 455	449 × 455	
9	JPG	449 × 455	449 × 455	
10	JPG	449 × 455	449 × 455	

3 Imperceptibility and Detectability Results

For performing the experiments, we embedded all the secret data made up of images into the cover images. As all the secret images were not used in the hiding process, only the first four bits of each byte are considered for the embedding process because only the important bits that hold the important parts of an image are considered. Besides, as we are applying the randomness technique, not every byte is included in the secret data. So, it is necessary to leave some parts of secret images that are considered as not important data and located in the last bits of the byte.

For each secret image, the data that was embedded decreased by approximately 20% of the capacity in the enhanced algorithm instead of the 50% of the original algorithm (Savitha) to achieve a better secure hiding process by applying the randomness. Then, MSE and PSNR values are calculated for each secret image to measure the imperceptibility and the impact of the hiding process. Tables 2, 3 and 4 show the results of different capacity values. In this experiment, the image (Lena.jpg), shown in Figure 2, is used as a cover image. Ten images are used in parallel as the image of a message for the hiding process and to test the effectiveness of the proposed technique.

Table 2 MSE, SNR and PSNR of capacity (97%)

	MSE			SNR			PSNR		
	Old	R1	R2	Old	R1	R2	Old	R1	R2
Img1	10.1	9.17	9.99	7.79	14.2	13	7.78	14.4	14.1
Img2	55.2	49.8	54.3	13.2	6.88	6.51	15.3	6.98	6.98
Img3	37.09	33.4	36.71	10.5	8.61	8.21	12.0	9.03	8.45
Img4	11.17	10.0	10.98	9.49	13.8	13.4	9.54	13.9	13.6
Img5	19.87	17.8	19.7	9.83	11.3	10.9	10	11.4	11.1
Img6	23.2	20.9	22.8	10.2	10.6	10.2	10.4	10.7	10.4
Img7	9.84	8.89	9.66	10.1	14.3	14.0	10	14.6	14.1
Img8	38.23	34.4	37.8	10.5	8.48	8.07	11.1	8.45	8.45
Img9	28.67	25.8	28.2	10.5	9.73	9.35	10.7	10	9.54
Img10	37.40	33.7	36.9	10.8	8.58	8.18	10	9.03	8.45

Table 3 MSE, SNR and PSNR of capacity (92.5%)

	MSE			SNR			PSNR		
	Old	R1	R2	Old	R1	R2	Old	R1	R2
Img1	10.1	8.88	10.0	7.79	14.3	13.8	7.78	14.6	13.9
Img2	55.2	48.3	54.3	13.2	7.02	6.50	15.3	6.98	6.98
Img3	37.09	32.4	36.6	10.5	8.75	8.21	12.0	9.03	8.45
Img4	11.17	9.80	11.0	9.49	13.9	13.4	9.54	14.1	1.36
Img5	19.87	17.3	19.8	9.83	11.4	10.8	10	11.7	11.1
Img6	23.2	8.63	22.8	10.2	14.5	10.2	10.4	14.7	10.4
Img7	9.84	33.4	9.67	10.1	8.61	14.0	10	9.03	14.1
Img8	38.23	33.4	37.9	10.5	8.61	8.07	11.1	9.03	8.45
Img9	28.67	25.1	28.3	10.5	9.86	9.34	10.7	10	9.54
Img10	37.40	32.7	36.98	10.8	8.70	8.18	10	9.03	8.45

Table 4 MSE, SNR and PSNR of capacity (90.6%)

	MSE			SNR			PSNR		
	Old	R1	R2	Old	R1	R2	Old	R1	R2
Img1	10.1	9.78	9.99	7.79	13.9	13	7.78	14.1	14.1
Img2	55.2	53	54.3	13.2	6.6	6.51	15.3	6.98	6.98
Img3	37.09	35.6	36.71	10.5	8.3	8.21	12.0	8.45	8.45
Img4	11.17	10.7	10.98	9.49	13	13.4	9.54	13.8	13.6
Img5	19.87	19.1	19.7	9.83	11.0	10.9	10	11	11.1
Img6	23.2	22	22.8	10.2	10	10.2	10.4	10.4	10.4
Img7	9.84	9.48	9.66	10.1	14.0	14.0	10	14.3	14.1
Img8	38.23	36.7	37.8	10.5	8.20	8.07	11.1	8.45	8.45
Img9	28.67	27.6	28.2	10.5	9.45	9.35	10.7	9.542	9.54
Img10	37.40	35	36.9	10.8	8.30	8.18	10	8.45	8.45

Fig.2 Cover image used to test the proposed technique

Different capacity values are used by taking two different values in the code. Five capacities are applied. The first capacity value is found by taking the average of 50 and 25, and the result is 97%. The second capacity takes the values of 20 and 10; the third capacity takes 16 and 8; the fourth capacity takes 8 and 4; the last capacity takes 4 and 2. The main principle for using those values is that they must be known values. At the same time, they must show clear results of capacity, which is why the multiples of value are taken, for example, in 97% capacity hiding process going every time except in 25 or 50, 62% capacity, not hiding between 2 or 4.

Three different experiments are implemented for the ten images. The first one is the Savitha method, the second experiment is an enhanced method with little improvement over applying randomization, while the third experiment applies more randomization as it checks the value of each byte of an image.

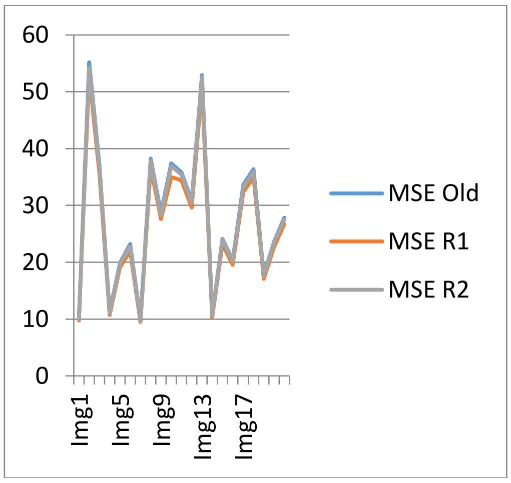
Tables 2 to 4 in the appendix show the results of MSE, SNR, and PSNR of three experiments. Each table has different capacity values, and the results in general clarify that the values of MSE are going to decrease when capacity is less.

Table 2 indicates the capacity of 97%. The first column refers to images used in three different experiments. The second column contains MSE values which indicate the values of errors. It is divided into three parts. The first part are the results of the Savitha algorithm (explained in section 3), the second part is related to the method of randomization, while the third part is the second enhanced method. The third column contains the results of SNR (quality of the image) of three different experiments and the last column includes the results of PSNR (peak value of SNR) of three different experiments.

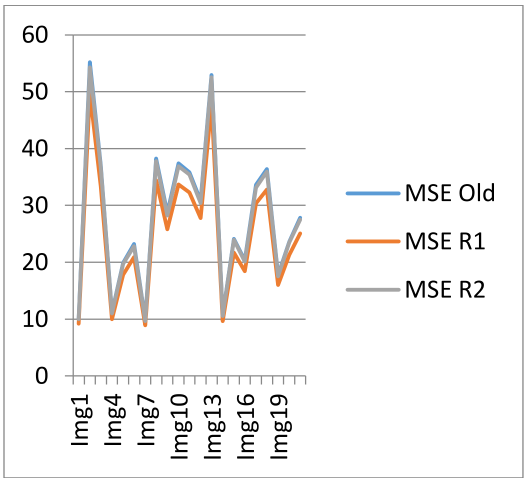
For MSE, which represents the statistical difference between cover and secret images, as we see from Table 2, the MSE values of the secret images range from 5.09 as the smallest value to 52.7 as the largest value among all the resultant stego images. Since the MSE value is still less than MSE of Savitha results, it indicates that the difference is marked even if it is negligible, and the MSE values of our results satisfy that criterion, then this is an indication that the statistical difference between the cover and the secret images in average is not too much, thus, the stego images are not perceptually detectable, which means high imperceptibility.

For PSNR, which represents the similarity between the cover and secret images, the values range from 6.98 as the smallest value to 17 as the largest value among all the resulting stego images.

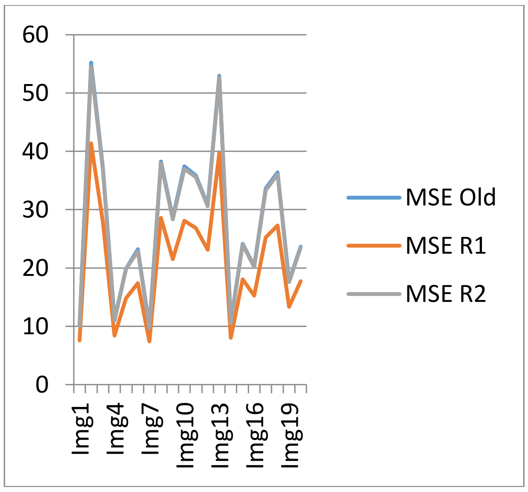
So, as we see from the results, the average of MSE ranges from 14 to 28 and PSNR ranges from 9 to 14 when the hidden images fill 25% to 45% of the cover image. This means both metrics MSE and PSNR indicate that the algorithm works with good imperceptibility. Figure 3 shows the chart of average MSE increase in Savitha and the second enhanced algorithms as the second technique which applied more randomness compared to the first enhanced algorithm which shows the better result of MSE and PSNR.



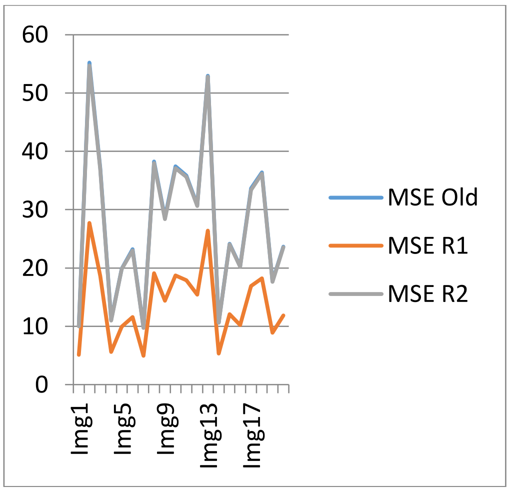
MSE of capacity 97%



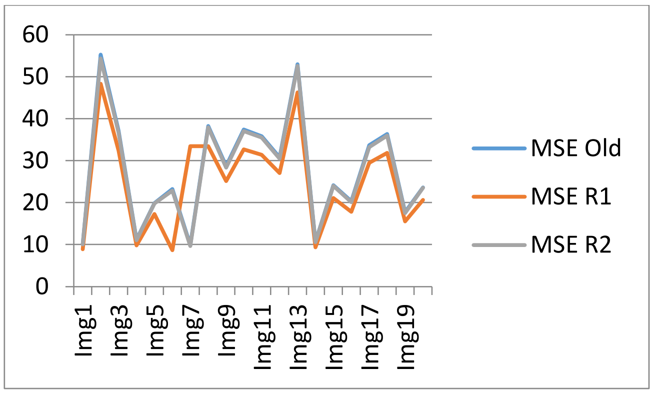
MSE of capacity 92.5%



MSE of capacity 90.6%



MSE of capacity 81.2%



MSE of capacity 62.5%
Fig.3 MSE averages

In our experiments, we chose ten images to test the results because we noticed that the change is too slight. First of all, ten images were chosen to be applied to the experiments, and the averages were taken. The average of MSE for the ten images with 97% capacity was:

Old method: 27.077, Method 1 (R1): 25.89, Method 2 (R2): 26.70

Then, five more images are added and the results are shown as follow:

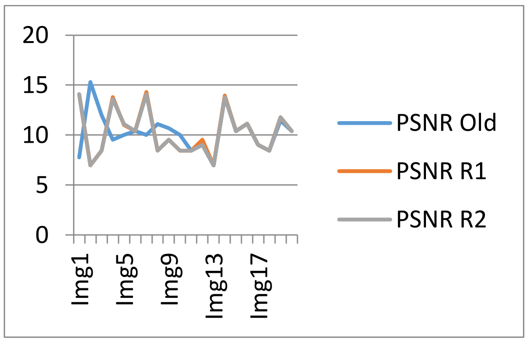
Old method: 28.34333, Method 1 (R1): 27.16067, Method 2 (R2): 28

It was noticed that there were slight changes in the experimental results. The total becomes ten images:

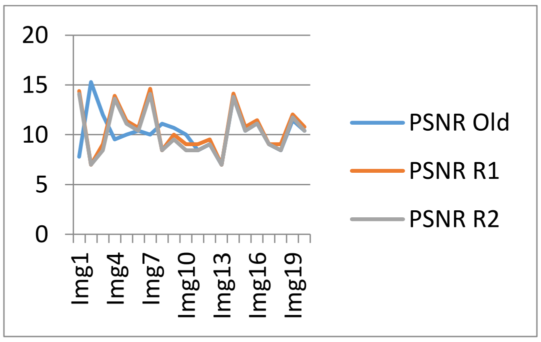
Old method: 27.842, Method 1 (R1): 26.7025, Method 2 (R2): 27.51

4 Result

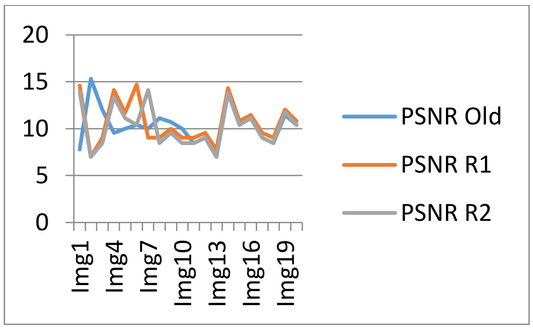
The results show that the change is too slight. From the results obtained, it is clear that the values still move in a narrow range, which leads to the effectiveness of the experiments even if images are going to increase. All the capacities have been taken to get the average of MSE and PSNR, as shown in Figure 4. The Savitha algorithm has been enhanced by adding the randomness feature to improve the level of security. The randomization pattern used in the proposed techniques is independent of specific bytes, leading to the difficulty in recognizing the secret data. Detecting the specific locations of bytes that contain the secret data is hard due to using a randomization pattern. This method can be used in other Steganography techniques based on a simple LSB idea to hide data in a sequential manner to enhance security.

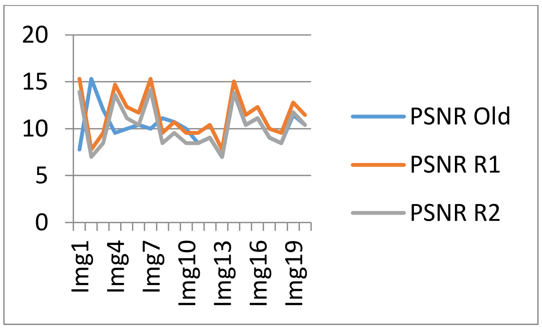


PSNR of capacity 97%

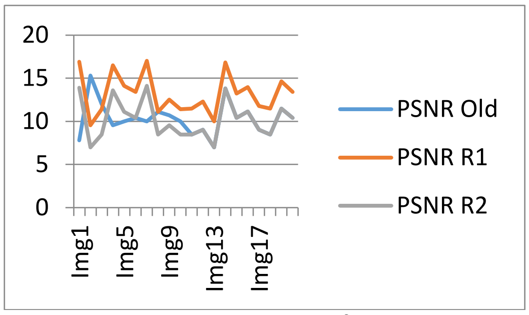


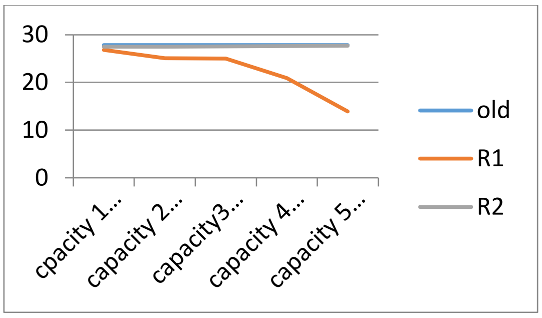
PSNR of capacity 92.5%



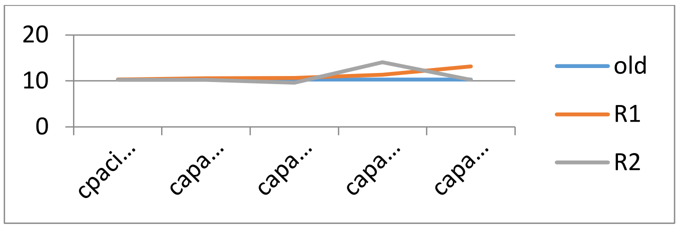


PSNR of capacity 90.6%　　PSNR of capacity 81.2%





PSNR of capacity 62.5%　　Average of MSE



Average of PSNR
Fig.4 The results of PSNR

5 Conclusion

The main aim of this research was to enhance the security and integrity of the message sent. The Savitha algorithm was enhanced, as it sequentially hides data by applying randomization. Thus, the algorithm was modified by manipulating the code to make the embedding look random. Both the algorithms were tested with 20 samples, and the results were evaluated. PSNR and MSE were evaluated and compared. Thus, the effectiveness of the proposed technique was proved. The achievements of this research are:

1. The Savitha algorithm has been enhanced by adding the randomness feature to improve the level of security.

2. The randomization pattern used in the proposed techniques is independent of specific bytes, leading to the difficulty in recognizing the secret data.

3. Detecting the specific locations of bytes that contain the secret data is hard due to using a randomization pattern.

4 This method can be used in other Steganography techniques based on a simple LSB idea to hide data in a sequential manner to enhance security.

Steganographic systems cannot be secure. Therefore, it is important to take care while hiding the secret data. First, the cover image should be of a suitable size to contain the secret data, such that the data does not fill, on average, more than 20% of a cover image. Second, it is preferred that the cover image is of medium dimensions, such as 500 × 500 pixels, or larger. Third, the data must be minimized as much as possible. For example, if the secret data is an image, effects such as sharpness must be removed. Fourth, the cover and secret images should be in grayscale to add confusion to the results and make the steganographic image not clear that it contains a secret message to increase the protection.

Some further research can be conducted in this area based on this research. One approach is to apply the proposed technique to various types of images, including compressed images. The proposed method can be applied to different image sizes and formats. Furthermore, the proposed method can be adapted to deal with text characters as message media.

References

[1] DAGAR E, DAGAR S. LSB based image steganography using X-Box mapping. [C]// Proceedings of the International Conference on Advances in Computing, Communications and Informatics. Piscataway NJ: Institute of Electrical and Electronics Engineers, 2014:351-355.

[2] MOLATO M R D, GERARDO B D, MEDINA R P. Secured data hiding and sharing using improved LSB-based image steganography technique. [C]// Proceedings of the 4th International Conference on Industrial and Business Engineering. New York NY: Association for Computing Machinery, 2018:238-243.

[3] SARRESHTEDARI S, GHOTBI M, GHAEMMAGHAMI S. On the effect of spatial to compressed domains transformation in LSB-based image steganography. [C]// Proceedings of theIEEE/ACS International Conference on Computer Systems and Applications. Piscataway NJ: Institute of Electrical and Electronics Engineers, 2009:260-264.

[4] AL FARSI G, JABBAR J, TAWAFAK R M. A Review on models of human face verification techniques. [C]// Proceedings of the International

Conference on Fourth Industrial Revolution. Piscataway NJ: Institute of Electrical and Electronics Engineers, 2019:1-5.

[5] CHIKOUCHE S L, CHIKOUCHE N. An improved approach for LSB-based image steganography using AES algorithm. [C]// Proceedings of the 5th International Conference on Electrical Engineering-Boumerdes. Piscataway NJ: Institute of Electrical and Electronics Engineers, 2017:1-6.