A Review on RGB Image Steganography based on Spatial Domain

Mohsen Hassan A. Hashem[1], Salah Eldin Deng Aljack[2] and Omaima Abdulaziz Bahaidara[3]

[1] Faculty of Mathematical Sciences and Informatics, Khartoum University. Khartoum. Sudan

[2] Faculty of Computer Science and Mathematics University of Bahri. Khartoum, Sudan

[3] Information Technology Department, Faculty of Engineering, Aden University. Aden, Yemen

DOI: https://doi.org/10.47372/uajnas.2022.n2.a10

Abstract

Steganography is not only the method to hide the message content but the message itself. [3]. So, "The importance of steganography lies as it hides the existence of the secret message which makes the job of attacker more difficult" [7]

This research paper presents PSNR results from different image steganography techniques such as Least Significant Bits (LSB), Most Significant Bits (MSB), Triple A algorithm, X-Box Mapping and Mod Factor method.

Execute these experimentally using VS 2017 (by C#) with 12 colored images and 6 different secret message lengths then conclude that the Mod Factor method gives good quality image with the highest PSNR and free error extracted secret message at receiver.

Key words: Steganography, image, secret message, spatial domain, method.

Introduction:

Steganography is the art and science of writing hidden messages in such the simplest way that nobody, except for sender and supposed recipient suspects the existence of message. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the Internet [8]. For hiding secret data in pictures, there exists many steganography techniques some are a lot complicated than others and everyone of them have various sturdy and weak points.

Steganography and Cryptography, are not similar. Steganography's intent is to hide the existence of the message, while cryptography is scrambling a message in such a way that it cannot be understood [8]. The use of steganography is the solution to the problem of information being looked by the intruders. There are two solutions to this problem: use steganography or use cryptography. The use of cryptography makes the secret message seen but un-understandable. So, the use of steganography is better because the secret message will be invisible, no one thinking that the secret message even exists.

Image Steganography is classified into two categories, these are Spatial-domain and Transform domain [8].

This research paper gives an overview of image steganography based on spatial domain, where the secret messages are embedded directly.

The methods presented in this research paper are Least Significant Bits (LSB), Most Significant Bits (MSB), Triple A algorithm, X-Box Mapping, Mod Factor.

Each Steganography method is mainly classified into two ways. The first is when no stego key imposed over the cover message, it is called pure steganography. A possible formula of this process may be represented as follows:

Cover medium + embedded message = stego medium

And to recover the message from stego medium the process is as follows:

Stego medium = cover medium + embedded message. [3]

The second way is secret key steganography where secret key is exchanged between receiver and the sender before the start of secret communication. As the secret is imposed to protect the embedded message. A possible formula of this process may be represented as follows:

Cover medium + embedded message + stego key = stego medium

To recover the message from stego medium the process is as follows:

Stego medium + stego key = cover medium + embedded message. [3]

Image Steganography Methods:

The 24-bit RGB image is chosen as a cover image which hides the text secret message inside red, green and blue color pixel values.

a-Least Significant Bits (LSB) insertion method:

Least Significant Bit is a spatial domain technique which is a very simple and straight forward method, takeing less time to embed message. The least significant bits of the cover image pixels are replaced by the message bits. The number of bits in each replaced pixel can be 1bit, 2 bits. 3 bits or 4 bits. At the receiver the least significant bits will be extracted and concatenated to get the secret message.[2,4,6,12,16,17,18,19,20]

b-Most Significant Bits (MSB) insertion method:

The message is embed into the most significant bit of the cover image pixel. Two methods used in this section are Most Significant Bits (3rd, 4th) and Most Significant Bits (5th, 6th).

One message bit is embed in a pixel by comparing the third and the fourth bits, or the fifth and sixth bits of the pixel. To embed 0, that two bits of the pixel must be equal, otherwise the fourth bit, the sixth bit value changed. Also, to embed 1, that two bits of the pixel must be not equal, otherwise the fourth bit, or the sixth bit value changed according to the method used. At the receiver, the secret message bits will be extracted by comparing the two bits in each pixel. If they are equal, it means that the message is 0 otherwise the message is 1. Then concatenated to get the secret message.[2,5,17]

c-Triple A algorithm:

This method can be applied to RGB images where each pixel is represented by three bytes to indicate the intensity of Red, Green, and Blue of that pixel. Variable number of bits were embeded in each channel (R, G or B) of pixel. Triple-A algorithm which needs to have a pseudo random number generator (PRNG). The assumption for PRNG is to give two new random numbers per iteration. The seeds of these PRNGs, namely Seed1 (S1) and Seed2 (S2), are formed as a function of the Key. This Key was used at receiver to generate the same S1 and S2 as PRNG at the sender, which help to extract the secret message. S1 is restricted to generate numbers in [0-6] while S2 is restricted to generate numbers in [1-3].[1,13]

Table 1: Seed1 Random Number Usage

Random No.	Meaning to the algorithm
0	use channel R.
1	use channel G.
2	use channel B.
3	use channel R and G.
4	use channel R and B.
5	Use channel G and B.
6	Use channel R, G and B.

Table 2: Seed2 Random Number Usage

Random No.	Meaning to the algorithm
1	embed 1 bit
2	embed 2 bits
3	embed 3 bits

Embedding Algorithm:

Input: A 24-bit RGB cover image of size m*n + Text Secret Message + Key
Output: Stego image of size m*n
Steps:

1. Take a RGB image and divide its pixels into equivalent Red, Green and Blue color.
2. Use Key to generate S1 and S2.
3. Embed number of secret message bits according to S2, into color channel according to S1.
4. Repeat steps 2 and 3 until all secret message bits get successfully embed in the cover image.

Extraction Algorithm:

To extract the secret message from the stego image, the following steps are performed:
Input: Stego image of size m*n + Key
Output: Secret message
Steps:

1. Take a RGB image and divide its pixels into equivalent Red, Green and Blue color.
2. Use Key to generate S1 and S2.
3. Extract number of secret message bits according to S2, from color channel according to S1.
4. Finally, concatenate the result to get the secret message.

d-X-Box Mapping:

At the first in this method, X-boxes must be generated. There are four X-boxes of 2*2 matrixes. and there are 16 values are stored in them from 0 to 15. To insert the values in the X-boxes, X-OR property is applied. For example, 13 is inserted in any of the X-boxes, 13= 1101, 11 XOR 01 = 10

Hence, the position of 13 in the x-box is 2^{end} row and 1st column. [7,10,14,15]

0	1		0	1		0	1		0	1	
0	4	0	15	11	0	10	14	0	5	1	0
13	9	1	2	6	1	8	3	1	7	12	1

Figure 1: X-Boxes

Now, we have different values of Red, Green and Blue pixel. So, to decide in which color to hide, there is a key taken which will decide the order in which to hide the data bits in the red, green or blue color pixel values. The key is a one dimensional array with values 0, 1 and 2 which will repeated again and again until all the secret bits get successfully embeded in the cover image. If the value of key is 0, then the pixel selected to hide the message in red is component, if it is 1, then green is component and, if it is 2, then blue component.[7].

Embedding Algorithm:

Input: A 24-bit RGB cover image of size m*n + Key
Output: Stego image of size m*n
Steps:

1. Take a RGB image and divide its pixels into equivalent Red, Green and Blue color.
2. Divide the secret message byte into four parts containing two bits each.
3. Map these four parts with the 4 X-boxes and get the new values for each combination.
4. Insert these new values (as 4 bits) into the LSB positions of the chosen color component values with the help of the key in the cover image.
5. Repeat steps 2, 3 and 4 until all secret message bits get successfully embed in the cover image.
6. The result after the insertion of secret message bits is the Stego image.

Extraction Algorithm:

To extract the secret message from the stego image, the following steps are performed:

Input: Stego image of size m*n + Key
Output: Secret message

Steps:
1. Select the pixel values according to the key from the stego image and extract 4 LSB's bit from the LSB position of each pixel.
2. Perform X-OR operation with those LSB bits.
3. Finally, concatenate the result to get the secret message.

e-Mod Factor:

It is also known as Modulus Method for Image Steganography. This method hide secret message in cover image based on calculating the modulus of RGB values with the mod-factor which is calculated as follows: Mod Factor = 2^n
Where, n is the number of bits to be hidden in one component of a pixel, so accordingly n can be 2,3,4 etc. By taking n=2, such that Mod Factor would be 4, two bits are inserted in each component, So, by this method, six bits of the secret message hidden in each pixel.[9,11].

Embedding Algorithm (Mod Factor = 4 = 2^2):

Input: A 24-bit RGB cover image of size m*n
Output: Stego image of size m*n
Steps:
1. Take a RGB image and divide its pixels into equivalent Red, Green and Blue component.
2. Calculate the modulus of each color component with the Mod Factor (4).
3. Now, compare the mod value as calculated in the previous step with the secret message bits (2 bits) which will be hidden in this iteration, by using Table 3.
4. If the mode value do not match with the previous table, then the pixel values are changed accordingly such that:
   a) If the message bits to be inserted are 00, then the mod value must be 0, if not, and then the pixel values are changed accordingly so that the mod value becomes 0.
   b) If the message bits to be inserted are 01, then mod value must be 1, if not, and then pixel values are changed accordingly so that mod value becomes 1.
   c) If the message bits to be inserted are 10 then mod value must be 2, if not, and then pixel values are changed accordingly so that mod value becomes 2.
   d) If the message bits to be inserted are 11 then mod value must be 3, if not, and then pixel values are changed accordingly so that mod value becomes 3.

Table 3: Secret message- mod value

Message Bits to be hidden	Mod Value
00	0
01	1
10	2
11	3

Extraction Algorithm(Mod Factor = 4 = 2^2):

For the retrieval of message, the same concept is used as for the insertion but in reverse.
Input: A 24-bit RGB Stego image of size m*n
Output: Secret message
Steps:
1. Take a RGB image and divide its pixels into equivalent Red, Green and Blue color.
2. Calculate the modulus of each color component with the Mod Factor (4).
3. Use Table 4 to extract the secret message bits from the mode values calculated in previous step.

Table 4: Mod value- secret message

Mod Value	Retrieved Message Bits
0	00
1	01
2	10
3	11

It can be explained as follows:
a) If the mod value is 0, then the retrieved message bits are 00.
b) If the mod value is 1, then the retrieved message bits are 01.
c) If the mod value is 2, then the retrieved message bits are 10.
d) If the mod value is 3, then the retrieved message bits are 11.
4. Finally, concatenate the result to get the secret message.

In Mod Factor method, 6 bits are inserted per pixel (2 bits in each component), therefore the proposed method provides a good capacity with minimum degradation in image quality. A Peak Signal-to-Noise Ratio is calculated which measure the quality of images used. Larger PSNR value indicates lower distortion and hence a better quality of image.[7]

Results and Discussion:

This section presents the experimental results obtained after implementing all methods, using visual studio 2017 (by C#). Then, calculate the PSNR and MSE to compare between the methods, as shown in the equation 1 and 2.

$$PSNR = 10\ log_{10}\left(\frac{(255)^2}{MSE}\right) \quad \ldots\ldots(1)$$

$$MSE = \frac{1}{m \times n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[C(i,j) - S(i,j)]^2 \quad \ldots\ldots(2)$$

Where,

m is the height of the cover image,
n is the width of the cover image,
C(i, j) pixel value before embedding data,
S(i, j) pixel value after embedding data.

MSE (Mean Square Error) is the average squared difference between a reference image and a distorted image. An Image steganography technique is efficient if it gives a low MSE. It is calculated or computed pixel-by-pixel by adding up the squared differences of all the pixels and dividing by the total pixel count.

PSNR (Peak Signal to Noise Ratio) is the most common metric used to evaluate the stego image quality. An Image steganography technique is efficient if it gives a high PSNR. The most common methods used for the evaluation of image quality by objective method is PSNR. PSNR and is of great importance as it is involved in the processing of images, and is used as standard model for the evaluation of different sorts of image quality evaluation methods. The value of PSNR is calculated by using the formula attached in equation 1.

Twelve color images each with size 512 × 512 , are used as cover-images in the experiments. The experiments are based on embedding 152, 208, 704, 864, 2992 and 3536 bits of secret message into the cover images by all methods discussed in the previous sections. The test images used are illustrated in Figure 2.

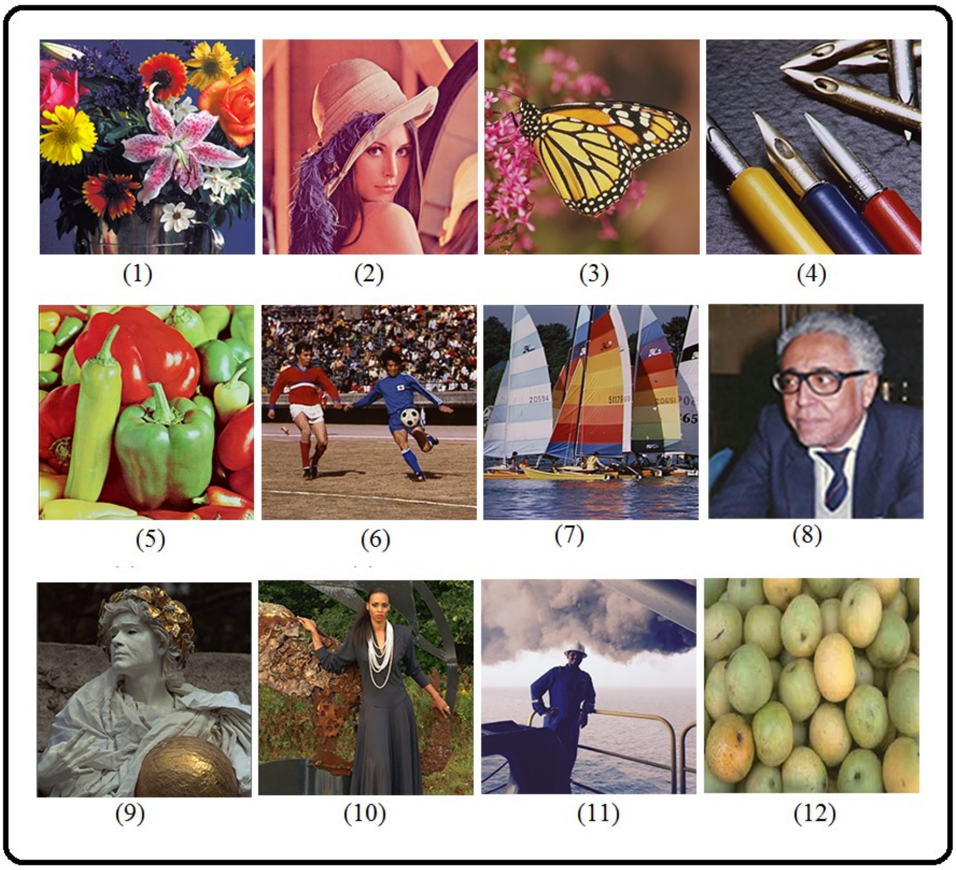

Figure 2: Cover images used in the experiments

Figure 3 illustrates screenshot of the embedding methods application.

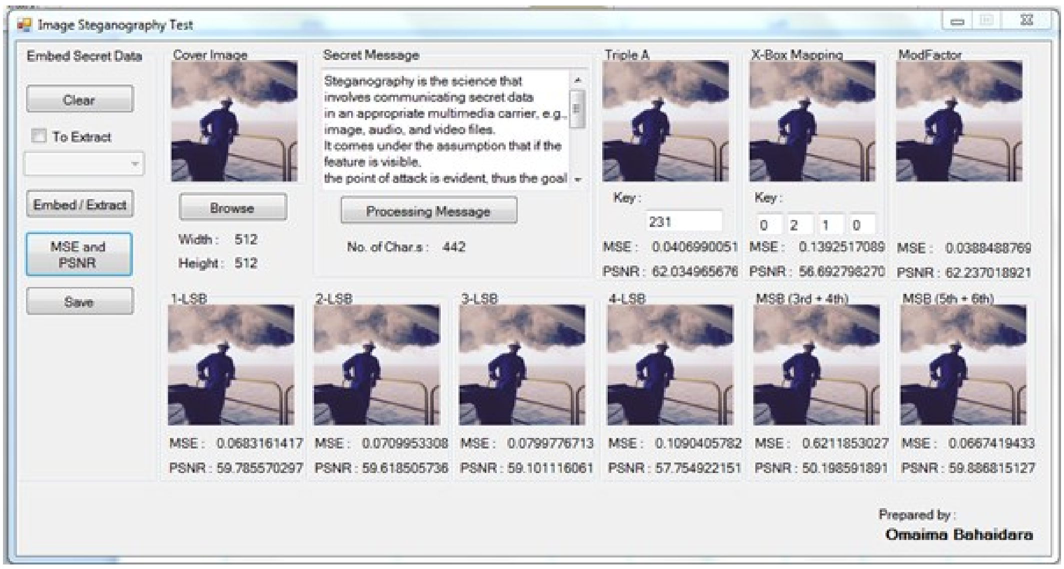

Figure 3: Screenshot of the embedding methods application

The results that are obtained from these experiments are recorded and summarized in Table 5 and Figure 4.

Table 5: The PSNR of cover image vs. stego image with different cover images, message length and embedding methods

		PSNR					
Test Image	Embedding Method	152 bits	208 bits	704 bits	864 bits	2992 bits	3536 bits
1. flowers.png	LSB 1	63.639	63.733	63.709	63.384	62.995	61.202
	LSB 2	63.673	63.761	63.675	63.328	62.713	60.975
	LSB 3	63.675	63.743	63.449	63.041	61.795	60.151
	LSB 4	63.505	63.505	62.740	62.256	59.593	58.309
	MSB 3&4	63.179	62.066	57.351	56.388	51.308	50.470
	MSB 5&6	68.195	68.323	66.712	65.549	62.020	60.197
	Triple A	64.634	64.751	64.675	64.243	63.569	61.526
	Mod. Factor	68.847	69.072	68.693	67.549	65.677	62.681
	X-Box Mapping	67.477	67.201	63.504	63.003	58.519	57.187
2. lenna.png	LSB 1	61.418	61.685	64.197	64.999	65.143	66.313
	LSB 2	61.435	61.703	64.200	64.988	65.047	66.156
	LSB 3	61.408	61.663	63.847	64.571	63.595	64.107
	LSB 4	61.308	61.463	62.991	63.442	61.195	60.917
	MSB 3&4	59.869	59.598	56.777	56.279	50.991	50.468
	MSB 5&6	61.989	62.234	64.323	64.759	61.938	61.895
	Triple A	59.659	59.835	61.294	61.678	61.696	62.177
	Mod. Factor	62.109	62.422	65.442	66.474	65.716	66.862
	X-Box Mapping	61.763	61.831	62.758	62.563	58.689	58.145
3. monarch.png	LSB 1	62.533	62.792	64.919	65.407	65.334	64.891
	LSB 2	62.541	62.798	64.847	65.278	64.836	64.381
	LSB 3	62.532	62.773	64.530	64.848	63.574	62.929
	LSB 4	62.472	62.566	63.766	63.554	60.758	59.913
	MSB 3&4	60.853	60.071	56.889	56.203	51.156	50.511
	MSB 5&6	63.107	63.345	64.789	64.911	61.827	61.357
	Triple A	60.079	60.219	61.247	61.434	61.279	61.068
	Mod. Factor	63.264	63.565	66.054	66.553	65.608	64.964
	X-Box Mapping	62.607	62.741	62.564	62.438	58.149	57.486
4. pens.png	LSB 1	60.680	60.909	62.941	63.542	63.665	64.529
	LSB 2	60.692	60.913	62.896	63.469	63.361	64.064
	LSB 3	60.675	60.898	62.659	63.141	62.242	62.470
	LSB 4	60.568	60.739	61.965	62.299	60.147	60.143
	MSB 3&4	59.799	59.361	56.878	55.985	51.080	50.452
	MSB 5&6	61.548	61.769	63.678	64.066	61.631	61.539
	Triple A	59.845	60.027	61.581	61.993	61.953	62.424
	Mod. Factor	61.658	61.941	64.569	65.410	64.878	65.852
	X-Box Mapping	61.270	61.372	61.969	62.105	58.247	57.878
5. pepper.png	LSB 1	57.043	57.149	57.984	58.198	58.265	58.611
	LSB 2	57.055	57.159	57.975	58.181	58.161	58.476
	LSB 3	57.051	57.153	57.901	58.065	57.753	57.954
	LSB 4	57.023	57.114	57.718	57.821	56.988	56.952
	MSB 3&4	57.572	57.396	55.725	55.224	50.891	50.312

	MSB 5&6	58.443	58.563	59.477	59.682	58.689	58.764
	Triple A	58.289	58.430	59.574	59.864	59.876	60.346
	Mod. Factor	58.499	58.643	59.828	60.143	60.043	60.516
	X-Box Mapping	58.214	58.268	58.566	58.637	56.261	56.239
6. soccer.png	LSB 1	62.040	62.264	63.982	64.330	64.268	63.792
	LSB 2	62.048	62.271	63.909	64.245	63.896	63.419
	LSB 3	62.033	62.230	63.622	63.886	62.769	62.129
	LSB 4	61.906	62.019	62.793	62.744	60.469	59.809
	MSB 3&4	60.582	59.818	56.695	56.347	51.209	50.468
	MSB 5&6	62.900	63.134	64.329	64.458	61.683	60.997
	Triple A	60.386	60.539	61.579	61.764	61.584	61.306
	Mod. Factor	63.057	63.336	65.492	65.927	65.098	64.322
	X-Box Mapping	62.486	62.476	62.454	62.183	58.091	57.545
7. yacht.png	LSB 1	64.684	64.788	64.635	64.197	63.717	61.590
	LSB 2	64.725	64.818	64.576	64.132	63.408	61.344
	LSB 3	64.685	64.741	64.219	63.726	62.331	60.501
	LSB 4	64.621	64.671	63.486	62.908	60.435	58.904
	MSB 3&4	63.702	62.419	57.504	56.361	51.182	50.284
	MSB 5&6	70.223	70.188	67.358	66.208	62.184	60.302
	Triple A	64.898	65.009	64.762	64.293	63.603	61.471
	Mod. Factor	71.192	71.592	70.248	68.678	66.348	62.898
	X-Box Mapping	68.276	68.365	64.323	63.045	58.316	57.106
8. G.png	LSB 1	63.985	64.136	64.559	64.322	63.933	62.0735
	LSB 2	64.013	64.159	64.512	64.239	63.610	61.826
	LSB 3	63.984	64.093	64.215	63.811	62.308	60.869
	LSB 4	63.935	63.966	63.514	63.033	60.308	59.196
	MSB 3&4	62.490	61.774	57.235	56.337	51.046	50.181
	MSB 5&6	66.949	67.024	66.316	65.565	62.057	60.375
	Triple A	63.164	63.295	63.593	63.382	62.879	61.342
	Mod. Factor	67.368	67.683	68.332	67.718	65.928	63.152
	X-Box Mapping	66.168	66.148	63.427	62.405	57.977	56.995
9. img1.png	LSB 1	62.957	63.135	64.103	64.099	63.841	62.506
	LSB 2	62.978	63.153	64.069	64.005	63.520	62.213
	LSB 3	62.964	63.101	63.789	63.599	62.347	61.180
	LSB 4	62.826	62.891	63.049	62.721	60.287	59.214
	MSB 3&4	61.693	60.880	57.199	56.214	51.124	50.367
	MSB 5&6	64.706	64.882	65.173	64.795	61.725	60.449
	Triple A	61.801	61.932	62.607	62.593	62.263	61.259
	Mod. Factor	64.931	65.189	66.581	66.521	65.297	63.372
	X-Box Mapping	64.073	64.061	62.939	62.486	58.167	57.272
10. img2.png	LSB 1	61.649	61.825	63.015	63.182	63.083	62.407
	LSB 2	61.659	61.828	62.972	63.109	62.798	62.104
	LSB 3	61.636	61.798	62.709	62.804	61.777	61.188
	LSB 4	61.598	61.689	62.211	62.096	59.989	59.278
	MSB 3&4	60.034	59.647	56.490	55.997	51.166	50.297
	MSB 5&6	62.269	62.423	63.196	63.128	60.866	60.058
	Triple A	59.884	59.997	60.709	60.807	60.637	60.231
	Mod. Factor	62.422	62.629	63.976	64.121	63.484	62.617

	X-Box Mapping	62.048	62.103	61.606	61.292	57.635	56.976
11. Moode.png	LSB 1	58.525	58.642	59.508	59.698	59.721	59.786
	LSB 2	58.538	58.654	59.499	59.674	59.594	59.619
	LSB 3	58.535	58.638	59.411	59.549	59.122	59.101
	LSB 4	58.511	58.544	59.121	59.138	57.959	57.755
	MSB 3&4	59.246	58.855	56.457	55.764	50.900	50.199
	MSB 5&6	60.512	60.666	61.666	61.822	60.164	59.887
	Triple A	60.235	60.405	61.757	62.062	61.951	62.035
	Mod. Factor	60.698	60.796	62.240	62.570	62.242	62.237
	X-Box Mapping	60.264	60.234	60.395	60.284	57.126	56.693
12. Pears.png	LSB 1	57.230	57.363	58.506	58.841	58.973	59.757
	LSB 2	57.235	57.366	58.489	58.818	58.865	59.594
	LSB 3	57.225	57.344	58.399	58.701	58.419	59.035
	LSB 4	57.204	57.302	58.182	58.369	57.561	57.877
	MSB 3&4	56.929	56.664	55.167	54.832	50.701	50.114
	MSB 5&6	57.728	57.859	58.916	59.192	58.324	58.736
	Triple A	57.086	57.213	58.296	58.604	58.666	59.381
	Mod. Factor	57.774	57.924	59.204	59.589	59.544	60.384
	X-Box Mapping	57.616	57.710	58.264	58.442	56.299	56.340

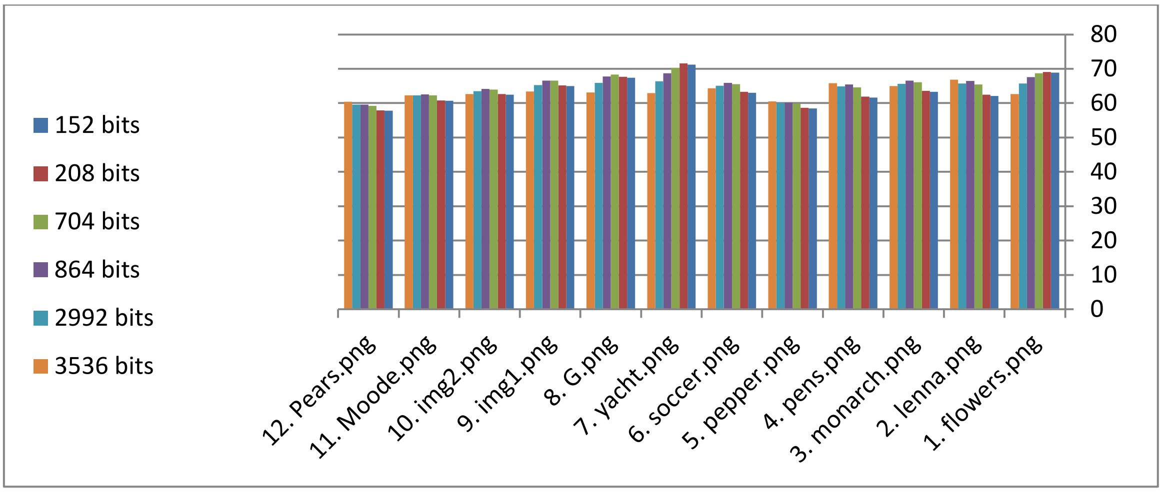

Figure 4: The PSNR with Mod-Factor method, different cover images and message length

In Table 5, secret messages with length 152, 208, 704, 864, 2992 and 3536 bits have been hidden in the cover images with sizes (512 x 512), using the nine embedding methods, it has been noticed that the Mod Factor embedding method has higher PSNR values.

The Mod Factor embedding method has higher PSNR values than other embedding methods which means that the stego image quality of the Mod Factor embedding method will be higher than the image quality of other embedding methods. This means the Mod Factor embedding method has succeeded to improve the security. while keeping the PSNR values higher than other techniques.

Conclusion:

Ensuring data security is a big challenge for computer users. The ultimate aim of steganography is to hide the existence of message in the cover medium.

The comparison study of different image steganography techniques, such as LSB, MSB, Triple A, Mod. Factor and X-Box Mapping, is performed in this paper.

All these techniques are based Spatial-domain, also the embedded secret message can be extracted from stego-images without the assistance of original images.

This paper provides a comparative table drawn from several experiments using twelve colored images and six different secret message lengths.

By studying the results, we can conclude that the Mod-Factor method is the best method with the highest PSNR. It can hide more data with less change in cover image. Also, don't need to send the original cover image to extract the secret message, the secret message has been extracted without mistakes and without needs the original cover image. About security enhancing, as a future work for implementing an encryption algorithm to provide more security for secret message can be done.

References:

1. Adnan Gutub, Ayed Al-Qahtani and Abdulaziz Tabakh (2009), "Triple-A: Secure RGB Image Steganography Based on Randomization", In 7th ACS/IEEE International Conference on Computer Systems and Applications, Rabat, Morocco, pp. 400–403.
2. Anil Khurana and B. Mohit Mehta (2012), "Comparison of LSB and MSB based Image Steganography", International Journal of Computer Science And Technology (IJCST), Vol. 3, Issue 3, July - Sept 2012, pp. 870- 871.
3. Beant Singh and Kul Bhusan Agnihotri (2015), "A Method to Hide Secret Information: Steganography", International Journal of Recent Advances in Engineering & Technology (IJRAET), ISSN (Online): 2347 - 2812, Volume-3, Issue -10, 2015, pp. 5-9.
4. Champakamala .B.S, Padmini.K and Radhika .D. K (2013), "Least Significant Bit algorithm for image steganography", International Journal of Advanced Computer Technology (IJACT), Volume 3, Number 4, 2013, pp. 34-38.
5. Chetan G. Tappe and Anil V. Deorankar (2017), " An Improved Image Steganography Technique based on LSB", International Research Journal of Engineering and Technology (IRJET), Volume 04, Issue 04, Apr- 2017, pp.1234 – 1237.
6. Ekta Dagar and Sunny Dagar (2014) , "Comparative Study of Various Steganography Techniques", International Journal of Emerging Engineering Research and Technology, Volume. 2, Issue 2, May 2014, PP 30-36.
7. Ekta Dagar and Sunny Dagar (2014), "LSB Based Image Steganography Using X-Box Mapping", International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2014, pp. 351-355.
8. Falesh M. Shelke, Ashwini A. Dongre and Pravin D. Soni (2014), "Comparison of different techniques for Steganography in images" International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 3, Issue 2, February 2014, pp. 171-176.
9. Gunjan Chugh, Rajkumar Yadav and Ravi Saini (2014), "A New Image Steganographic Approach Based on Mod Factor for RGB Images", International Journal of Signal Processing, Image Processing and Pattern Recognition (IJSIP), Volume 7, No. 3, 2014, pp. 27-44.
10. Madhugeeta Verma and Poonam Dhamal (2015), "High Security of Data Using Steganography with Hybrid Algorithm", International Journal of Science and Research (IJSR), Volume 4, Issue 11, November 2015, pp. 2469- 2473.

11. Mahmoud S. Abdelbar, Abdelmgeid A. Ali and M. Hasabala (2021), "A Technique for Increasing Payload Capacity of RGB Images Steganography based on Mod Factor and Segmentation", International Journal of Computer Applications, Volume 183, No. 13, July 2021, pp. 29- 35.
12. Mamta Jain and Pallavi Kumari (2017), " A Survey on Digital Image Steganography using RGB Color Channel ", Suresh Gyan Vihar University International Journal of Environment, Science and Technology, Volume 3, Issue 1, Jan 2017, pp.21-25.
13. Md. Mizanur Rahman, Pronab Kumar Mondal, Indrani Mandal and Habiba Sultana (2016), "Secure RGB Image Steganography Based on Triple-A Algorithm and Pixel Intensity", International Journal of Scientific & Engineering Research, Volume 7, Issue 3, March-2016,pp. 864-869.
14. Pooja and Santosh Kumar (2016), "Enhancing security in Image Steganography using X-Box", International Journal of Innovative Research in Technology (IJIRT), Volume 3, Issue 1, pp. 335- 338.
15. Prasannakumar Patil and Satish Shet.K (2015), "Implementation of an image steganography technique using X (X-OR) - Box mapping", International Journal of Advances in Electronics and Computer Science, Volume-2, Issue-8, Aug.-2015, pp. 32- 36.
16. Ressi Dwitias Sari and Andysah Putera Utama Siahaan (2018), "Least Significant Bit Comparison between 1-bit and 2-bit Insertion", International Journal for Innovative Research in Multidisciplinary Field (IJIRMF), Volume 4, Issue 10, Oct – 2018, pp. 110- 113.
17. Salam R. Mahdi, Yahia Najar and Mahmood Shaar (2012). "High Complexity with Steganography in LSB and MSB", Journal of Information Security Research, Volume 3, Number 2, June 2012, pp. 74- 80.
18. Satwinder Singh and Varinder Kaur Attri (2015), "State-of-the-art Review on Steganographic Techniques", International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol.8, No.7, pp.161-170.
19. Shashikant Singh, Seema Yadav, Ankur Raj and Priya Gupta (2018), "A Survey Paper on Different Steganography Techniques", International Conference on Emerging Trends in Expert Applications & Security (Kalpa Publications in Engineering), Volume 2, 2018, Pages 103- 108.
20. Vijaysih K. Jadeja, Mikin K. Dagli, Ajeetkumar S. Patel and Mayank P. Devani (2022), "More Secure Images Steganography Techniques Based on Encryption", International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), Volume 2, Issue 2, February 2022, pp. 394- 398.

مراجعة للإخفاء في الصورة الملونة بناءً على المجال المكاني

محسن حسن عبدالله هاشم[1]، صلاح الدين دينق الجاك[2] وأميمة عبدالعزيز باحيدرة[3]

[1] العلوم الرياضية والمعلوماتية، جامعة الخرطوم، الخرطوم، السودان

[2] كلية علوم الحاسوب والرياضيات جامعة بحري، الخرطوم، السودان

[3] قسم تقنية المعلومات، كلية الهندسة، جامعة عدن، عدن، اليمن

DOI: https://doi.org/10.47372/uajnas.2022.n2.a10

الملخص

إخفاء المعلومات ليست فقط طريقة لإخفاء محتوى الرسالة، ولكن الرسالة نفسها. لذا "تكمن أهمية إخفاء المعلومات؛ لأنه يخفي وجود رسالة سرية مما يجعل مهمة المهاجم أكثر صعوبة".

تقدم هذه الورقة البحثية نتائج (ذروة الإشارة إلى نسبة الضوضاء) PSNR من تقنيات إخفاء الصور المختلفة مثل وحدات البت الأقل أهمية، والبتات الأكثر أهمية، وخوارزمية Triple A، ورسم خرائط X-Box ، وطريقة عامل التعديل.

تم تنفيذ هذه التجارب باستخدام VS 2017 (بواسطة #C) مع 12 صور ملونة و 6 أطوال مختلفة للرسائل السرية.

أخيرًا تم استنتاج أن طريقة Mod Factor تنتج صورة جيدة الجودة مع PSNR عالية ورسالة سرية خالية من الأخطاء عند الاستلام.

الكلمات المفتاحية: إخفاء المعلومات، صورة، رسالة سرية، نطاق مكاني، طريقة.