Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

A Secure and High Capacity Image Steganography Approach Using Huffman Coding and RSA Encryption

Eman Lateef Kadhem[a], Salwa Shakir Baawi[b]

[a]*Department of Computer Science, University of Al-Qadisiyah, Al-Muthanna, Iraq. Email: eman.kadhem@qu.edu.iq*

[b]*Department of Computer of Information System, University of Al-Qadisiyah,Al-Diwaniya, Iraq. Email: salwa.baawi@qu.ediu.iq*

ARTICLE INFO

Article history:
Received: 10 /03/2023
Rrevised form: 21 /04/2023
Accepted : 25 /04/2023
Available online: 30 /06/2023

Keywords:
Capacity
Image Steganography
Huffman
RSA

ABSTRACT

Steganography is the practice of hiding data, such as images, videos, or text, within a cover image without it being detectable by the human eye. Several factors, such as the capacity, security, and robustness of the technique, are essential when transferring information using this method. In this study, we propose a new approach to image steganography that improves the Least Significant Bit (LSB) technique by utilizing images of 24 bits in each pixel. Improving the capacity and security of LSB-based steganography requires a combination of techniques, such as indirect embedding, embedding in multiple channels, and applying cryptographic and compression techniques. This approach conducts by encrypting each compressed secret message bit with the most significant bit of the red channel and then saving the output bit (hidden bit) in the least significant bit of the (green/blue) channel according to the row value (odd/ even). To further security, the suggested approach uses multi-level encryption; employs RSA to encrypt the secret message before applying the Huffman compression and encrypting the hidden bit by (XOR/XNOR) in the embedding method based on the row value of the pixel. Meanwhile, it is planned to use the Huffman coding technique to shorten the length of the encrypted message that will be inserted in the cover image. For the color photos in this work, the standard images were acquired from a standard dataset (USC-SIPI). The suggested approach performs better when measured in terms of mean square error (MSE), peak signal-to-noise ratio (PSNR), and comparison to findings from related prior efforts.

https://doi.org/10.29304/jqcm.2023.15.2.1231

1. Introduction

It is essential that communication be more secure as it develops and information technology is utilized more often [1]. Although the Internet is the most practical and pertinent medium for interpersonal communication, sensitive data is vulnerable to several risks. While communicating through the network, the Internet is not always assured [2], [3]. Security is mainly provided by two methods: data hiding and cryptography. Common and widely used methods of hiding data are steganography and watermarking. Encryption is defined as a process to secure messages that are transmitted via an unsecured communication channel. This process is based on using an encryption key before information transmission to another side. Only the legitimate key may be used to decode the secret data once it has been delivered to the designated recipient. Any outsider probing an unprotected internet connection in such a system receives a garbled text that he cannot comprehend and cannot decrypt since he lacks the encryption key[4].

∗Corresponding author

Email addresses:

Communicated by ‘sub etitor’

Steganography attempts on including private information that has to be sent in secret in an innocent-looking object. Many file formats, such as text, audio, video, and image files, may be utilized as containers. The process of selecting a particular steganographic medium is very important, though, since it greatly affects the steganographic design and security of the system [5].

But Due to their widespread use on the Internet, digital images are the most often utilized cover media. Moreover, pictures may have a lot of visual redundancy, which increases their power to conceal security information by taking advantage of the blind spots in the human visual system. Redundancy is defined as bits in a file that provides precision above what is required for the item. So, they may be changed without anybody noticing [5]. There have been several studies done in the field of image steganography [6] and [7] focused on improving the capacity, while [8] and [9] tried to increase transparency. Nonetheless, some researchers, such as those in [10] and [11], worked to increase security.

The basic objective of image steganography techniques, whether spatial or transform, is to enhance data concealing capacity while minimizing cover image distortion. Consequently, a steganography technique based on RSA, Huffman coding, and improved LSB is suggested in this research. The following benefits of the suggested plan: The proposed scheme is immune to hacking because it uses RSA, and the embedded secret message is hardly detectable by the human visual system (HVS) because Huffman coding is increasing the embedding bits in the cover image. High capacity is hidden as a result of the Huffman coding algorithm and improved LSB.

The rest of the study is organized as follows: The research work on image steganography techniques was described in the second part. The final section goes into further detail about the process. The fourth section then contains the findings of the experiment. Lastly, the fifth section's conclusion is described.

2. Related work

Several image steganography techniques used in data hiding have been published in the literature, For example,

In [12], the authors combined Huffman coding and Vigenere cipher techniques to create a novel image steganography technique. This approach improved the security of the message by preventing it from being recovered without the decrypting key and the Huffman Dictionary table. It also employed the technique of Exploiting Modification Direction (EMD) to enhance the robustness. Results showed that the proposed approach was more efficient than the previous image's steganography methods with respect to transparency by PSNR of (55.71) dB and capacity is (52.400) bytes, as well as, robustness achieved.

The image steganography technique, which is built on encryption and the LSB algorithm, was introduced by the researchers in [13]. The suggested technique was founded on the idea of extraction one of three chromatic RGB channel from each pixel, then identifying the channel in which a bit of the encrypted message was hidden. This technique employed four 512x512 images with three messages each in a different size. The findings of the experiments demonstrated that the suggested approach led to successful results.

An additional strategy in [14], The researchers proposed the Deflate algorithm, a multi-level compression method that makes use of both LZ77 and Huffman coding, for compressing the message content. LSB has been used to embed the compressed text into the cover picture. The suggested technique showed performance that was better than the state of the art using benchmark photographs. This can demonstrate how successful Deflate was as a data compression method before using the LSB.

A new approach for image steganography was developed by researchers in [11]. It makes use of two control random parameters and multi-level encryption. The stego image is undetectable when P Even/P Odd is used. To improve payload capacity, this approach used Huffman coding to compress secret data before embedding. It increased PSNR and image security. The suggested approach first compresses and encrypts secret data. To detect 0 (P Even) and 1 during embedding, it records and maps each bit in the stego picture to match the secret bits with the LSB (P Odd). The results have shown how to use P Even/P Odd with two control random parameters and multi-level encryption to combine the hidden message and enhance image steganography.

Researchers in [15] proposed using data compression and encryption for image steganography. The secret message is encrypted using RSA before compression. Before embedding, Huffman encoding is then completed, and each bit of a private message's Huffman code is embedded in the compressed cover by altering the least significant bit (LSB). Image cover size is decreased through discrete wavelet transform (DWT) loss compression. Savings, compression

time, capacity ratio, MSE, PSNR, SSIM, and compression speed of the system were evaluated. This system outperformed others using the same methodology.

Several approaches have been developed based on the literature to overcome the main security, capacity, and imperceptibility problems with conventional LSB. When it is difficult for an adversary to extract secret data from the cover medium, embedding techniques are considered secure. On the other hand, a high capacity means that there is more area for secret messages to be stored in the cover image. The improved quality of the stego-image generated by the algorithm, which is comparable to the original cover image, is the final element that leads to imperceptibility. The purpose of this study is to improve the quality and embedding capacity of a stego-image using the improve LSB technology by proposing a new strategy for selecting a better pixel for embedding.

3. Methodology

Before we start describing the proposed scheme, the main tools employed in it are as follows: -

3.1. RSA encryption

Two distinct keys are utilized for encryption and decryption in asymmetric key encryption. All senders receive the public key, while the owner of the private key is only that one person (the recipient). The fundamental drawback of this key coding, however, is that it is slower than symmetric algorithms [16]. Two distinct keys are utilized for decryption and encryption in asymmetric key encryption. All senders receive the public key, while the owner of the private key is only that one person (the recipient). The fundamental drawback of this key coding, however, it is slower than symmetric algorithms [17].

The two different types of keys that we have are the public key and the private key. The system is referred to as asymmetric if data encryption uses the public key and data decryption uses the private key alone. There is no requirement for two parties in a public key cryptosystem to exchange sensitive information. Data is therefore safer and less likely to be stolen or altered. It is the most popular strategy for implementing public keys and is named after MIT researchers who developed the idea in 1977. They were Ronald Rivest, Adi Shamir, and Leonard Adleman. When keeping secrecy is a major priority, the Advanced Encryption Technique (RSA), a symmetric key encryption standard, is routinely used to protect data [15].

To create the public and private keys in RSA, perform the following algorithm1 [18].

Algorithm 1: RSA encryption Input: plaintext P Output: cipher text C
4. Step1: Consider that P and Q are two large prime integers such that P = Q. 5. Step2: Compute N=P*Q. 6. Step3: Determine (P*Q) = (P-1) (Q-1). 7. Step4: Consider the public key k1 that has the properties GCD (ϕ (N), k1) =1 and 1<k1< ϕ (N). 8. Step5: Choosing the private key k 2 will ensure that k 2* okay mod ϕ (N) =1. 9. The process of encryption and decryption is as follows: 10. Step6: Compute cipher text C from plaintext P such that: 11. Encryption: $C=P^{k1}$ mod N Decryption: P= C^{k2} mod N

Algorithm 1 shows the RSA encryption

3.2. Huffman algorithm

Before embedding, the secret message is represented using the compression technique known as Huffman coding, which results in the secret message having the shortest binary length feasible. The benefit of this method is that it reduces the size of secret messages, increasing payload capacity and security[19].

Let's say a numerical illustration is given to demonstrate how Huffman will be used in the suggested model. The Huffman compression method is carried out in seven steps, as follows, using the five separate symbols S, E, C, R, and T with frequencies of 15, 56, 8, 12, and 19, respectively, in this example.

Step1: Construct a table using the provided symbols and their frequency values.

Symbol	S	E	C	R	T
Frequency	15	5	8	12	19

Step 2: Depending on the frequency of the symbols, reorder the input symbols in ascending order.

Symbol	C	R	S	T	E
Frequency	8	12	15	19	56

Step 3: Rearrange the final table and combine the two frequencies with the lowest values.

Symbol	CR	S	T	E
Frequency	20	15	19	56

Step 4: Repetition of Step 3 is necessary to get a single frequency number.

Step 4.1	symbol	CR	S	T
	Frequency	20	15	19
Step 4.2	Symbol	CR	ST	E
	Frequency	20	34	56
Step 4.3	Symbol	CRST	E	
	Frequency	54	56	
Step 4.4	Symbol	CRSTE		
	Frequency	100		

Step 5: Create a Huffman tree by assigning a value of 0 or 1 to each pair of branches in the tree.

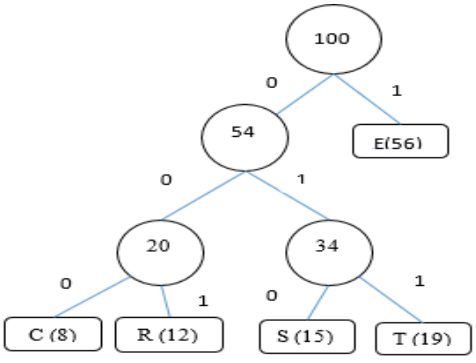


Fig. 1 - Shows Huffman tree

Step 6: Create the final table (Huffman coding), which includes the leaf nodes and their associated codes after the Huffman tree.

Table 1 - Huffman coding

Symbols	Code	Length	Frequency
E	1	1	56
T	011	3	19
S	010	3	15
R	001	3	12
C	000	3	8

Step 7: Rewrite the output codes using the Huffman coding table [1, 000, 001,010,011], as shown in Table 1 (in the column of Code), to produce the compressed secret message text [20].

According to the Huffman tree, the two parent nodes had frequencies of 20 and 34, respectively. The frequency of their respective children (8, 12) and (15, 19) was added to make Figure 1. The high-frequency letter "E" will be made with a great deal of effort. In the output code for Step 7, each leaf has a path that travels to the main node (100), thus the numbers and directions of the pathways correspond to these values. The letter "S" is coded as 010 if you move from "C" to "100," for example. "E" has code 1 since there is just one path that leads to the main node. When the Huffman tree for the text is complete, we get (218 bits), but the text's ASCII coding requires 800 bits (100 characters × 8 bits), as shown in Table 1.

ASCII: (56+19+15+12+8) * 8 = 800 bit

Huffman: (56*1+19*3+15*3+12*3+8*3) = 218 bits.

3.3. LSB algorithm

A bit of the secret message is exchanged for the least significant bit (8th bit) of some or all of the bytes in an image. In 24 bit images, we are able to add three bits of information to each pixel, one in each LSB position of the three eight bit values. The output stego image appears almost identical to the cover image since the LSB doesn't alter the appearance of the image with changes in value [21].

The LSB method [22] is the most well-known and traditional technique for concealing the message in a digital image. In the LSB technique, the message is concealed in the least significant bits (LSBs) of an image's pixel values. The LSBs of each pixel are used in this manner to distribute the binary version of the secret message.

a. Advantages of LSB

1. Less suspicious in the view of humans.
2. This strategy is widely used and is easy to implement.
3. Extreme perceptual transparency.
4. Completely guaranteed hidden data.

b. Disadvantages of LSB

1. Robustness, tampering, and resistance are three weaknesses.
2. Extremely susceptible to filtration of any kind.
3. Cropping, rotation, scaling, and adding further noise because the secret message is to be declared.

3.4. Suggested scheme

In this paper, suggests an image steganography scheme that has two main phases: embedding and extracting. The embedding phase involves three distinct stages namely: encryption RSA, Huffman encoding, and embedding by improving the LSB algorithm of steganography. On another side, extracting phase also includes three stages: extracting the hidden data, Huffman decoding, and decryption RSA. The scheme suggested in this study was inspired by related research that created an embedding technique that has been demonstrated to be successful. As a consequence, an improved steganography approach has been created by combining the benefits of the previously mentioned methods [14], [15]. Where maximizing payload capacity and security while maintaining transparency is the major objective. Figure 2 explains the workflow of the suggested image steganography scheme.

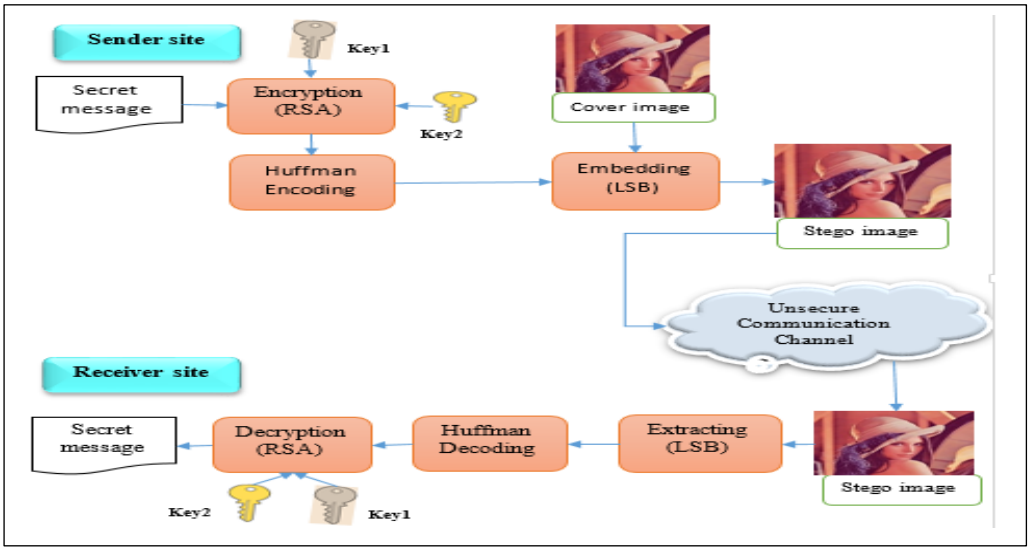


Fig. 2 - workflow of the suggested Image Steganography scheme

Following are the phases recommended during the embedding phase in more detail:

1. The cover picture is read as a 512*512 matrix.
2. After reading the covert text, change each character's value to its ASCII equivalent.
3. The RSA technique is then used to encrypt each ASCII value.
4. The encrypted secret message is then compressed using Huffman encoding after that.
5. To create the stego picture, include the secret message that has been compressed using Huffman technique, which embeds using the improved LSB technique.

While the following are the specific phases of the extraction phase:

1. The stego picture is read as a 512*512 matrix.
2. Retrieval of the embedded hidden bits by the improved LSB algorithm
3. Apply the Huffman decoding algorithm on all the bit messages retrieved from the stego image matrix.
4. After all the secret messages have been extracted, convert them into ASCII values.
5. Each ASCII value is then decrypted by the RSA algorithm to obtain the original message.

3.4.1 Improvement LSB algorithm

This part conducts embedding the compressed secret message using Huffman coding inside the cover image. The fundamental concept behind the improvement process is to suggest a strategy that works in a test of the even or odd row number in the image. According to Table 2, depending on the value of the row, one operation (either XOR or XNOR) must be performed.

Table 2 – truth table of XOR and XNOR functions

A	B	XOR	XNOR
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	0

Apply the XNOR operation between the first bit from the Red channel and the bit from the compressed binary text if the row number is odd. The output bit can then be integrated into the pixel's last bit of the blue channel in the cover picture. In every other case, combine the first bit from the red channel with the bit from the compressed binary text message using the XOR technique. The output bit from the proposed strategy may then be used to embed the final bit of the green channel of a pixel in the cover picture to create the Stego image, as explained in algorithm 2.

Algorithm 2: Embedding of Improvement LSB
Inputs: Cover image file, secret message compressed
Outputs: Stego image, Huffman code table

Step1: Split image into three channels (R, G, B), for each channel is a matrix of pixels (x, y) with size = 512*512
Step2: Read the compressed secret message
Step3 : For each coordinate pixel (x, y) in a range of (image size)
Step 3.1: If (x mod 2 = 0) then
a. Applying XOR Operation between bit from compressed binary text message and first bit from red channel.
b. Embedding the result bit from sub step 3.1.a in the last bit of green channel in the cover image to obtain Stego image
Step 3.2: Else
a. Applying XNOR Operation between another bit from Compressed binary text and first bit from red channel
b. Embedding the result bit from sub step 3.2.a in the last bit of blue channel in Cover image to obtain Stego image
Step 4: Return Stego image

Algorithm 2 shows the embedding improvement process of the LSB

Figure 3 explains the main steps for the embedding secret message into cover image to produce the stego image.

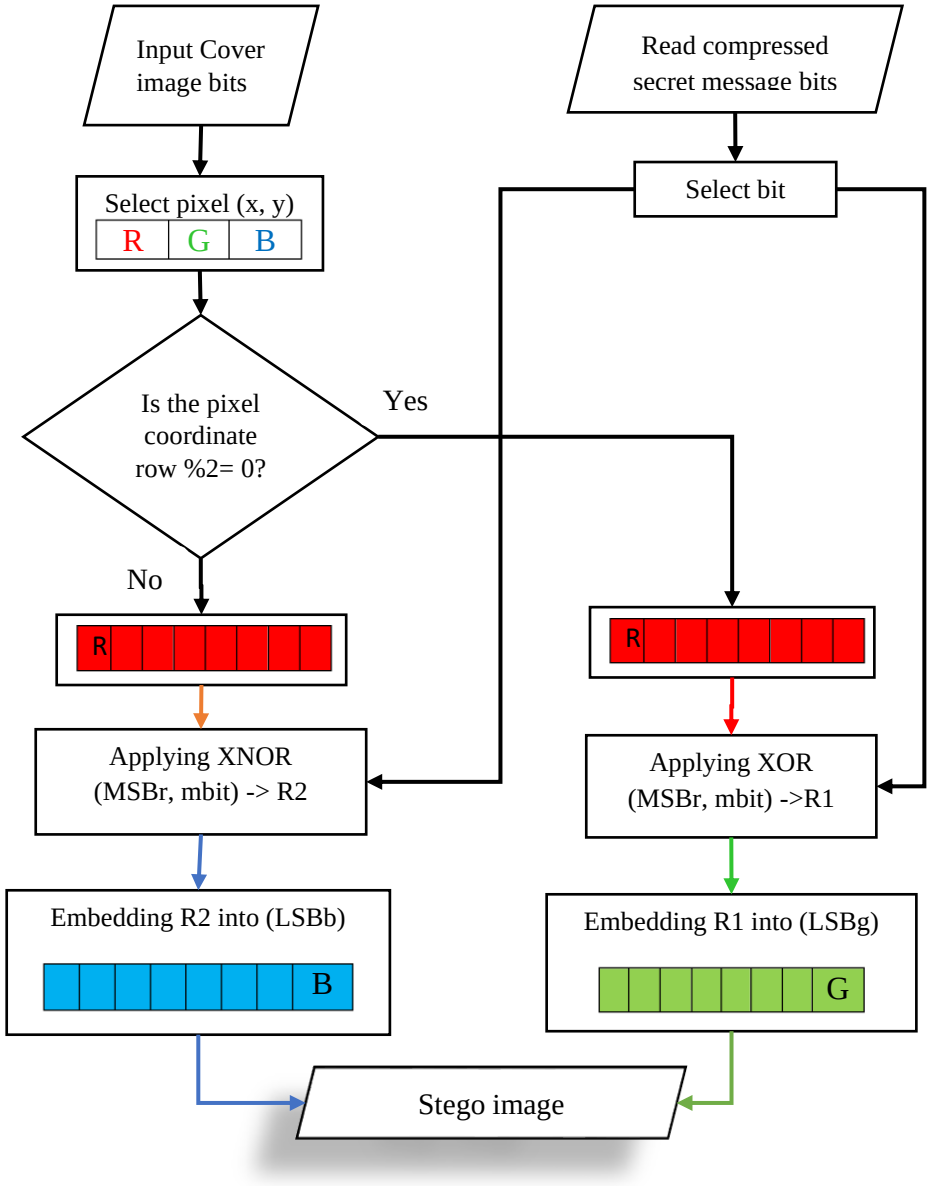


Fig. Error! No text of specified style in document. **- Embedding process**

3.4.2 Improvement Extracting algorithm

To There has been built a reverse operation to retrieve the embedded message from the stego picture. Following is a summary of Algorithm 3 used in the extraction step to extract the embedded secret message from the stego image: -

Algorithm 3: Improvement LSB extracting process Input: Stego image, Huffman code table Output: compressed Secret message
Step1: Split image into three channels (R, G, B), for each channel is a matrix of pixels (x, y) with size = 512*512 Step2: For each pixel coordinate (x, y) in a range of (image size)

Step2.1: If (x mod 2 = 0) then

a. Retrieve the least significant bit of green channel of stego image
b. Applying XOR operation between the result of 2.1.a and most significant bit from red channel (same pixel).

Step2.2: Else

a. Retrieve the least significant bit of blue channel from stego image
b. Applying XNOR operation between the result bit from 2.2.a and most significant bit of red channel (same pixel) to obtain bit of the compressed binary text bit

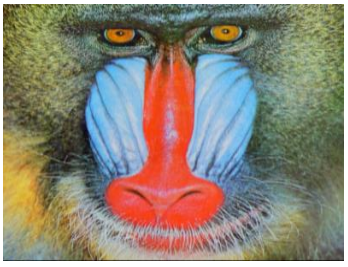
Step3: Return the compressed Secret message

Algorithm 3 shows the extraction improvement process of the LSB

4. Experiments and Discussion

The major goal of this study is to increase payload capacity while retaining image quality. The USC-SIPI Data Base, which has been widely utilized by researchers in the field of information concealing and delivers reliable results in terms of quality, was picked for five photographs with a size of 512*512 pixels. These photos were then used to test the suggested approach. Figure 4 displays the images that were used in the present study.


Lena.png


Papoon.png


Peppers.png


Man.png


Airplanef16.png

Fig. 4 – The cover images

Nevertheless, additional secret messages of various sizes written in English are required for the evaluation of the suggested image steganography model in addition to the images that were employed. Table 3 illustrates the effect of secret message lengths on applying the encoding methods.

Table 3 ***The effect of the secret messages length on the encoding methods***

Encoding Methods	Messages Size (in bits)								
	1 KB	2 KB	4 KB	6 KB	8 KB	16 KB	32 Kb	42.65 KB	60 KB
ASCII code	8192	16384	32768	49152	65536	131072	262144	-	-
Huffman coding	4388	8743	17422	26092	34985	69416	139253	185955	261945

Figure 5 illustrates graphically how utilizing various encoding methods impacts the length of secret messages of various sizes.

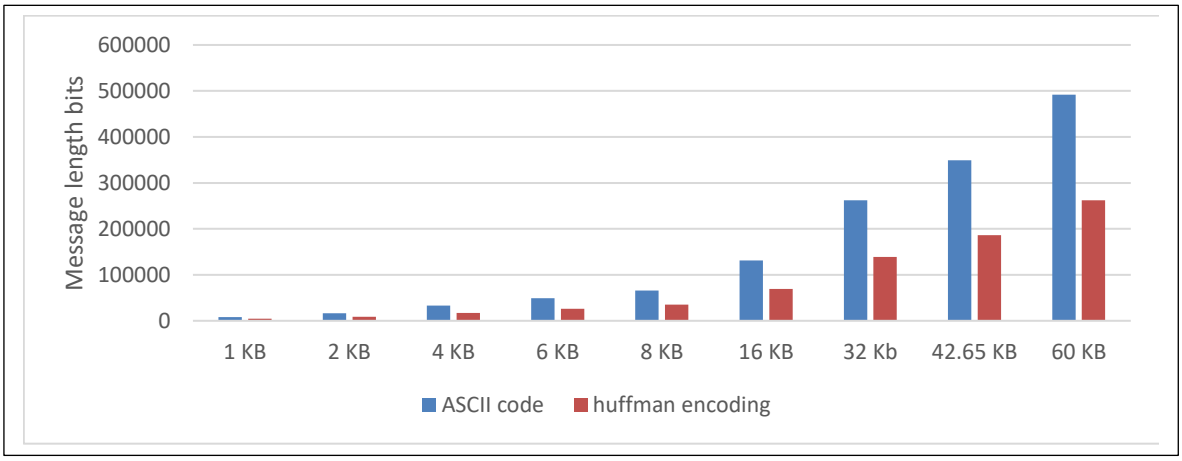


Fig. 5 – comparing secret message ASCII code and Huffman

These methods lower the amount of the final message length that needs to be hidden within the cover, as shown in Figure 4, which reduces the length of the message contents. Because of this, there are fewer places where data must be hidden. This will assist us to improve the payload capacity while maintaining the image quality.

Payload capacity, peak signal-to-noise ratio (PSNR), mean squared error (MSE), and structural similarity index measurement (SSIM) are four public metrics that will be used to assess the performance of the suggested model.

1- Payload capacity

This This subsection explains the evaluation of the proposed model in terms of the capacity which is the number of bits encoded in each pixel, as expressed in bits per pixel (bpp) using Equation (1) [23]. Table 4 illustrates the results of the proposed image steganography model in terms of the capacity ratio using images (512x512) when hiding secret messages of various sizes using ASCII and Huffman coding:

$$Capacity\ (bpp) = \frac{No.of\ embedded\ bits}{Total\ pixel\ in\ a\ cover\ image} \qquad (1)$$

Table 4 Results in terms of the exploited payload capacity in different encodings methods

Message size in (Kbyte)	ASCII	Huffman coding
M1= 1	3.13	1.67
M1= 2	6.25	3.34
M2= 4	12.50	6.65
M3= 6	18.75	9.95
M4= 8	25.00	13.35
M5= 16	50.00	26.48
M6= 32	100.00	53.12
M7= 42.65	-	70.94
M8= 60	-	99.92
Average	-	31.71

As shown in Table 4, the payload capacity varies depending on which encoding method is used. For instance, we observed that the amount of the payload capacity that is exploited in the ASCII encoding method was bigger than the payload capacity exploited in the suggested method. The suggested method proved that reduces the length of a secret message compared to ASCII, therefore, the suggested method exploited less amount number of bits per pixel to hide secret message bits. As a result, the suggested method achieved the task of improving the capacity of LSB. Figure 6 shows the relationship between the messages length and capacity ratio depending on which encoding method is employed.

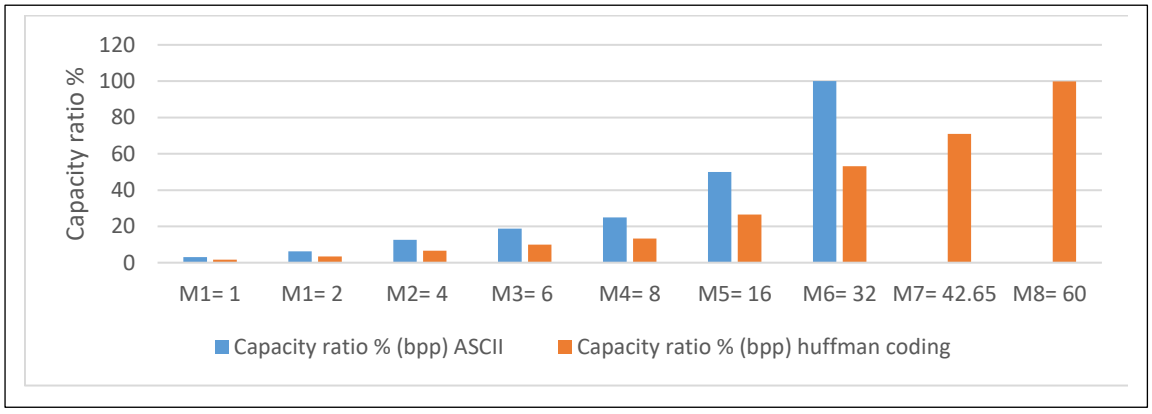


Fig. 6 - Relationship between the messages size and capacity ratio

The full payload capacity can be hidden in the image utilizing the ASCII encoding technique of approximately (32 kilobytes), as demonstrated in Figure 5. The suggested approach suggests expanding the payload capacity data inside the picture to reach about (60 Kilobytes). The primary goal of this study, an increase in payload capacity using the suggested approach, has thus been achieved.

2- The peak signal-to-noise ratio (PSNR)

PSNR aims to statistically measure the quality of the image, it can be expressed as in Equation (2)[24]:-

$$PSNR = 10log_{10}\frac{255^2}{MSE} \qquad (2)$$

Where 255 represents the Max value of intensity in the image. The bigger value of PSNR refers to better steganography.

3- mean squared error (MSE)

By computing the total quadratic error between (cover and stego) pictures, the MSE is computed [13]. It may be written out as Equation (3).

$$MSE = \frac{1}{M\times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(S-C)^2 \qquad (3)$$

N and M refer to the width and height of the (cover and stego) images. Whereas, C (i,j) and S(i,j) refer to the pixel values of two images correspondingly. When MSE has a small value that is mean better steganography is obtained.

4- Structural Similarity Index Measurement (SSIM)

This statistic determines how comparable the stego image and original image of an object are. When the SSIM value is 1, it denotes that the similarity between them reflects the top. The SSIM value picks values between -1 and 1[25]. The SSIM is shown as follows in Equation (4):

$$SSIM(C,S) = \frac{(2\mu_C\mu_S+C_1)\times(2\sigma_{CS}+C_2)}{(\mu_C^2+\mu_S^2+C_1)\times(\sigma_C^2\sigma_S^2+C_2)} \qquad (4)$$

Table 5 shows the Results of our proposed method in terms of PSNR, MSE, and SSIM for hiding 1KB, 2KB, 4KB, 6KB, 8KB, 16KB, 32KB, 42.65KB, and 60KB secret message data hiding in 512*512 image size.

Table 5 Results of PSNR, MSE and SSIM

Message length	measure	Color images 512*512				
		Lena	Papoon	Peppers	Airplane f16	Man
1KB	MSE	0.0028	0.0027	0.0027	0.0028	0.0027
	PSNR	73.606	73.690	73.742	73.606	73.680
	SSIM	0.9999	0.9999	0.9999	0.9999	0.9999
2KB	MSE	0.0056	0.0055	0.005	0.0055	0.0054
	PSNR	70.621	70.699	70.661	70.712	70.756
	SSIM	0.9999	0.9999	0.999	0.9999	0.9999
4KB	MSE	0.0109	0.0111	0.0111	0.0110	0.0112
	PSNR	67.717	67.673	67.649	67.692	67.622
	SSIM	0.9998	0.9999	0.999	0.9999	0.9998
6KB	MSE	0.0165	0.016	0.016	0.0165	0.0165
	PSNR	65.933	65.968	65.946	65.954	65.943
	SSIM	0.9998	0.9999	0.9998	0.9998	0.9997
8KB	MSE	0.0221	0.0220	0.0223	0.0223	0.0224
	PSNR	64.679	64.698	64.641	64.631	64.623
	SSIM	0.9997	0.9999	0.9998	0.9997	0.9996
16KB	MSE	0.0440	0.0442	0.0441	0.0441	0.0443
	PSNR	61.690	61.667	61.683	61.677	61.664
	SSIM	0.9995	0.9999	0.9996	0.9995	0.9993
32 KB	MSE	0.0887	0.0886	0.0887	0.0881	0.0888
	PSNR	58.647	58.652	58.650	58.678	58.644
	SSIM	0.9992	0.9997	0.9994	0.9991	0.9990
42.65 KB	MSE	0.1180	0.1179	0.1178	0.1177	0.1173
	PSNR	57.409	57.415	57.419	57.419	57.434
	SSIM	0.9990	0.9994	0.9992	0.9989	0.9989
60 KB	MSE	0.1666	0.1670	0.1661	0.1658	0.1672
	PSNR	55.911	55.902	55.924	55.933	55.896
	SSIM	0.9986	0.9992	0.9990	0.9987	0.9988

On the other hand, Table 6 compares of the proposed scheme with recent research [13] and [14] for three different payload sizes: 4700 bits, 14500 bits, and 24250 bits. The suggested scheme performs better. Table 6 shows that the proposed scheme is superior in performance to methods in [13], [14].

Table 6 Comparison results of suggested model with other methods

Color image with size 512*512	methods	Secret message 4700 bits		Secret message 14500 bits		Secret message 24250 bits		Average	
		MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
Airplane	Neamah et al (2020)	0.0457	60.44	0.0448	55.54	0.0445	52.84	0.045	56.27
	Tayyeh et al.(2022)	0.0016	76.03	0.0046	71.49	0.0067	69.82	0.0043	72.44
	Our Proposed	**0.0016**	**76.08**	**0.0049**	**71.20**	**0.0082**	**68.95**	0.0049	72.07
Lina	Neamah et al (2020)	0.0336	61.74	0.0347	56.75	0.0339	53.65	0.034	57.38

	Tayyeh et al.(2022)	0.0017	75.81	0.0046	71.47	0.0067	69.84	0.0043	72.37
	Our Proposed	**0.0015**	**76.11**	**0.0048**	**71.25**	**0.0081**	**69.03**	0.0048	72.13
Peppers	Neamah et al (2020)	0.0879	57.65	0.0865	42.84	0.0862	39.99	0.0868	46.82
	Tayyeh et al.(2022)	0.0017	75.77	0.0046	71.49	0.0066	69.89	0.0043	72.38
	Our Proposed	**0.0015**	**76.14**	**0.0049**	**71.20**	**0.0082**	**68.98**	0.0048	72.10
Baboon	Neamah et al (2020)	0.0740	58.39	0.0049	42.84	0.0729	40.89	0.0506	47.37
	Tayyeh et al.(2022)	0.0017	75.80	0.0044	71.60	0.0066	69.90	0.0042	72.43
	Our Proposed	**0.0016**	**76.02**	**0.0049**	**71.19**	**0.0081**	**69.03**	0.0048	72.08
Average		**0.00155**	**76.0875**	**0.004875**	**71.21**	**0.00815**	**68.9975**	0.0048	72.098

As seen in Table6, it is evident that the suggested model outperformed the results of Neamah et al (2020) for all images in terms of MSE and PSNR. This has been shown for three message sizes where the suggested model provided a better MSE and PSNR values. Also, it achieved an improved steganography performance compared to the Neamah et al (2020) research. This can be demonstrated by the effectiveness of data compression, particularly when employing the LSB technique and Huffman coding. The text is compressed, resulting in a considerable reduction in text size, which raises PSNR to (72.09833333) and lowers MSE to (0.0048583333).

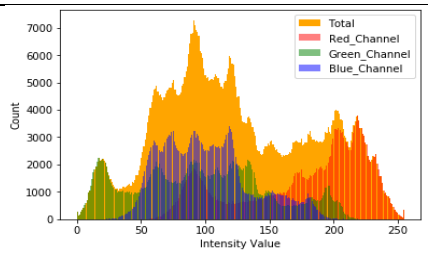
5. Histogram analysis

A histogram analysis between the cover and stego images of the suggested approach is shown for evaluating the resilience against typical statistical attacks [26]. Figures 7 and 8 shows the histograms produced by the embedding procedure using the images of the Lena and man. The frequency of pixel values is typically modified by embedding the secret bits in the cover image and can be seen in the histogram. The frequency histogram for the cover images is shown in Figure 7 (c) and Figure 8 (c). The frequency histogram of the stego pictures is displayed in Figure 7 (d) and Figure 8 (d) at the same time to hide 2 killobytes into image.

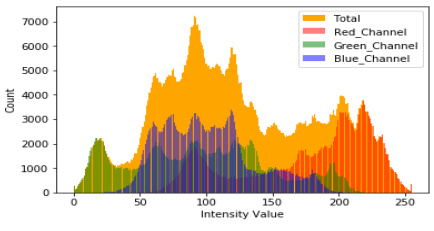
The findings show that distortions caused by the embedding process are invisible to human vision. The analysis's findings indicate that there is no discernible difference between the cover and stego images' histograms. The different histogram approach has difficulty detecting embedded information. Therefore, the proposed method has the advantage of reduced visual distortion while yet maintaining the security of secret messages.

(a) Cover- Lena

(b) Stego- Lena



(c) Cover- Lena Histogram



(d) Stego - Lena Histogram

Fig. 7 - Histogram of the cover and stego Lena image

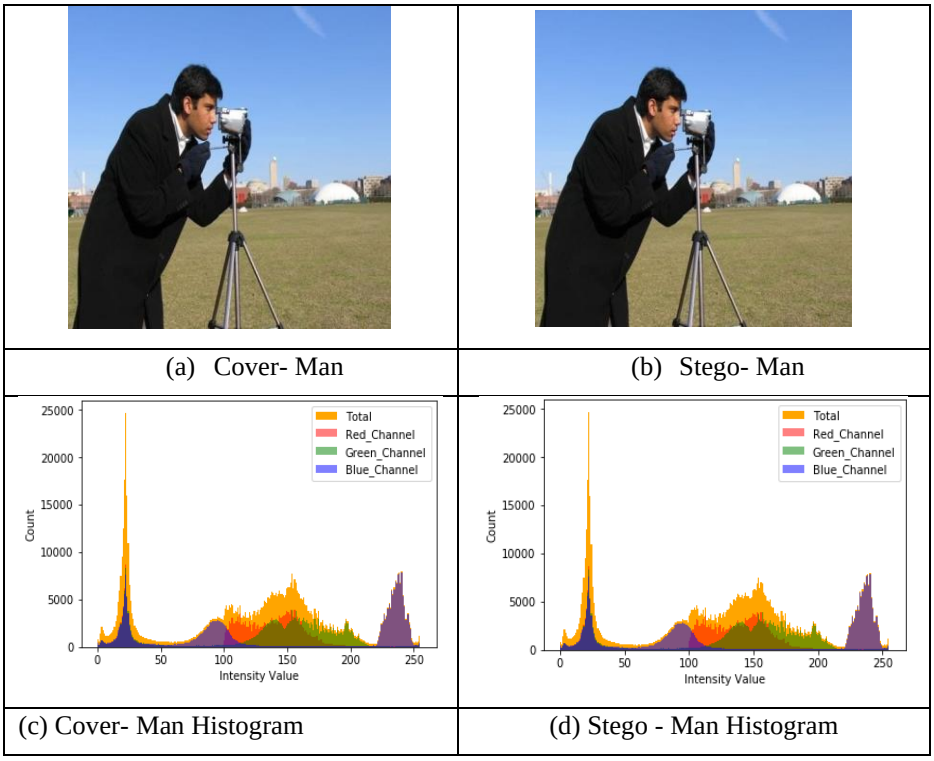


Fig. 8 - Histogram of the cover and stego man image

6. Conclusion

This research offers a model for secure high-capacity image steganography that uses cryptography and data compression techniques. RSA cipher is a technique of cryptography that is widely used for providing secure data during transmission via communication channels. RSA is an asymmetric processor; which uses two different but linked keys. Huffman coding is a frequently used technique to compress text to a smaller size without information loss. Information is encoded as a binary string (bits of 1's and 0's), and the goal is to transmit the information unambiguously with the fewest possible. Then encrypted and compressed information is inserted into a cover image file using LSB steganography. PSNR, MSE, and histogram have been used to find the operation of the suggested algorithm rule. Experimental results demonstrated that the values area unit of MSE is low and the PNSR of the suggested model is high, which ensures the transparency of the cover information through the stego image. Also, the histogram diagram of the cover image and stego image are extremely near to one another, which confirms the robustness of the suggested model against the attacks.

References

[1] A. Taha, A. S. Hammad, and M. M. Selim, "A high capacity algorithm for information hiding in Arabic text," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 32, no. 6, pp. 658–665, 2020.

[2] R. H. Ali and J. M. Kadhim, "Text-based Steganography using Huffman Compression and AES Encryption Algorithm," *Iraqi J. Sci.*, vol. 62, no. 11, pp. 4110–4120, 2021, doi: 10.24996/ijs.2021.62.11.31.

[3] G. Maji and S. Mandal, "Secure and robust image steganography using a reference image as key," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 7, pp. 2828–2837, 2019.

[4] R. Atta and M. Ghanbari, "A high payload steganography mechanism based on wavelet packet transformation and neutrosophic set," *J. Vis. Commun. Image Represent.*, vol. 53, pp. 42–54, 2018, doi: 10.1016/j.jvcir.2018.03.009.

[5] K. Shabnam, S. & Hemachandran, "LSB based Steganography using Bit masking method on RGB planes," *Int. J. Comput. Sci. Inf. Technol.*, vol. 7, no. 3, pp. 1169–1173, 2016.

[6] D. R. I. M. Setiadi, "Improved payload capacity in LSB image steganography uses dilated hybrid edge detection," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 2, pp. 104–114, 2019, doi: 10.1016/j.jksuci.2019.12.007.

[7] Y. Wang, M. Tang, and Z. Wang, "High-capacity adaptive steganography based on LSB and Hamming code," *Optik (Stuttg).*, vol. 213, pp. 1–9, 2020, doi: 10.1016/j.ijleo.2020.164685.

[8] M. Cem kasapbaşi and W. Elmasry, "New LSB-based colour image steganography method to enhance the efficiency in payload capacity, security and integrity check," *Sadhana - Acad. Proc. Eng. Sci.*, vol. 43, no. 5, pp. 1–14, 2018, doi: 10.1007/s12046-018-0848-4.

[9] Ahmed, A. and A. Ahmed, "A Secure Image Steganography using LSB and Double XOR Operations," 2020.

[10] A. Y. Hindi, M. O. Dwairi, and Z. A. AlQadi, "A Novel Technique for Data Steganography," *Eng. Technol. Appl. Sci. Res.*, vol. 9, no. 6, pp. 4942–4945, 2019, doi: 10.48084/etasr.2955.

[11] M. H. Mahdi, A. A. Abdulrazzaq, M. S. Mohd Rahim, M. S. Taha, H. N. Khalid, and S. A. Lafta, "Improvement of Image Steganography Scheme Based on LSB Value with Two Control Random Parameters and Multi-level Encryption," in *IOP Conference Series: Materials Science and Engineering*, 2019, vol. 518, no. 5, pp. 1–14, doi: 10.1088/1757-899X/518/5/052002.

[12] Z. S. Younus and M. K. Hussain, "Image steganography using exploiting modification direction for compressed encrypted data," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2951–2963, 2022, doi: 10.1016/j.jksuci.2019.04.008.

[13] R. M. Neamah, J. A. Abed, and E. A. Abbood, "Hide text depending on the three channels of pixels in color images using the modified LSB algorithm," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 1, pp. 809–815, 2020, doi: 10.11591/ijece.v10i1.pp809-815.

[14] H. K. Tayyeh and A. S. A. Al-Jumaili, "A combination of least significant bit and deflate compression for image steganography," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 1, pp. 358–364, 2022, doi: 10.11591/ijece.v12i1.pp358-364.

[15] O. F. A. Wahab, A. A. M. Khalaf, A. I. Hussein, and H. F. A. Hamed, "Hiding data using efficient combination of RSA cryptography, and compression steganography techniques," *IEEE Access*, vol. 9. pp. 31805–31815, 2021, doi: 10.1109/ACCESS.2021.3060317.

[16] A. Jeromel and B. Žalik, "An efficient lossy cartoon image compression method," *Multimed. Tools Appl.*, vol. 79, no. 1–2, pp. 433–451, 2020, doi: 10.1007/s11042-019-08126-7.

[17] F. Adhanadi, L. Novamizanti, and G. Budiman, "DWT-SMM-based audio steganography with RSA encryption and compressive sampling," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 2, pp. 1095–1104, 2020, doi: 10.12928/TELKOMNIKA.v18i2.14833.

[18] S. Bhargava and M. Mukhija, "Hide Image and Text Using Lsb, Dwt and Rsa Based on Image Steganography," *ICTACT Journal on Image and Video Processing*, vol. 9, no. 3. pp. 1940–1946, 2019, doi: 10.21917/ijivp.2019.0275.

[19] K. Sailunaz, M. Rokibul Alam Kotwal, and M. Nurul Huda, "Data Compression Considering Text Files," *Int. J. Comput. Appl.*, vol. 90, no. 11, pp. 27–32, 2014, doi: 10.5120/15765-4456.

[20] A. Gupta, A. Bansal, and V. Khanduja, "Modern lossless compression techniques: Review, comparison and analysis," 2017, doi: 10.1109/ICECCT.2017.8117850.

[21] Stuti Patel, "Enhancement in PSNR using Inverted LSB Mechanism," *Int. J. Eng. Res.*, vol. V4, no. 10, 2015, doi: 10.17577/ijertv4is100572.

[22] M. Yadav and A. Dhankhar, "Image Steganography Techniques: A Review," *IJIRST*, vol. 2, no. 2, pp. 243–248, 2015.

[23] I. J. Kadhim, P. Premaratne, and P. J. Vial, "Improved image steganography based on super-pixel and coefficient-plane-selection," *ScienceDirect*, vol. 171, no. 107481, pp. 1–20, 2020.

[24] N. Subramanian, O. Elharrouss, S. Al-Maadeed, and A. Bouridane, "Image Steganography: A Review of the Recent Advances," *IEEE Access*, vol. 9. pp. 23409–23423, 2021, doi: 10.1109/ACCESS.2021.3053998.

[25] R. Wazirali, R. Ahmad, A. Al-Amayreh, M. Al-Madi, and A. Khalifeh, "Secure watermarking schemes and their approaches in the iot technology: An overview," *Electron.*, vol. 10, no. 14, 2021, doi: 10.3390/electronics10141744.

[26] A. A. Zakaria, M. Hussain, A. W. A. Wahab, M. Y. I. Idris, N. A. Abdullah, and K. H. Jung, "High-capacity image steganography with minimum modified bits based on data mapping and LSB substitution," *Appl. Sci.*, vol. 10, no. 11, pp. 1–19, 2018, doi: 10.3390/app8112199.