Hybrid Approach For Multimedia Data Security With Embedded Image Compression

Ruchita Ingole[1] & Shrikant J. Honade[2]

[1]P.G. Student, Electronics & Telecommunication Engineering, G H Raisoni College of Engineering & Management, Amravati, Maharashtra, India.
[2]Assistant Professor, Department of Electronics & Telecommunication Engineering, G H Raisoni University, Amravati, Maharashtra, India.

***Abstract*— In today's online world, security of data and information is very important. Intruder can hack the information if it is not secured properly. They can use this information for any unethical purpose. The present work deals with new steganography algorithm based on hiding text, image and audio in cover image. LSB technique is used for encoding data. The encoded data is compressed in PNG format to enhance security, this is a lossless compression technique. Hence after decompression there is no loss in data. Performance parameter such as PSNR and compression ratio showed better results as compared to the existing techniques. The compression ratio found to be least and PSNR value found to be maximum as compared to the existing techniques.**

Keywords— compression ratio, steganography, PSNR, secured & LSB algorithm

INTRODUCTION

THE today's world, Internet data transmission must be fast as well as provided. There are many ways to do communication quickly, but at the same time protecting data when transferring is very important. This way, hiding data has become the most important thing for network security. Flexible software with low-priced digital devices, such as music players and other handheld devices, has made it possible for consumers from all over the world to create and share multimedia data. Consumers exchange sensitive messages and files over the Internet, and anyone can have some information to keep secret. Steganography is a powerful tool that enhances security in data transmission[4].

In steganography, data that will be hidden first hidden in a file called a cover and forms another new object called Stego, and then this new file can be transmitted or saved. Various formats that can be used for steganography are text, images, audio, protocol. Steganography image is one of the important and known methods of data hiding, which embeds secret data into image signals using encryption and decryption[6].
Based on the sound steganography has a greater potential to hide information because the sound files are larger than the image and small amplitude changes can store a huge amount of information.

The aim of the work is to design and develop a novel method for data hiding by combining multimedia data using LSB steganography and embedded image compression. To provide security from intruder and data hackers is most important objective of this project. Number of methods based on encryption and decryption are available. In case of encryption hacker come to know that data is encrypted, so he try to decrypt it. In steganography as image looks very natural nobody come to know that data is hided in image, so most of the intruder even not try to check for data. Hence security gets provided. Even some of intruder come to know that there is data, it is very difficult to get that data without key. The following objectives are proposed in this project:

- To design and develop an algorithm for Text, Image and Audio data encoding and decoding.
- To design and develop an algorithm for compression and decompression.
- To evaluate the developed algorithms using parameters such as PSNR and compression ratio.

LITERATURE SURVEY

The invisible Water-marking scheme, in which the watermark is a little plane of grey scale instead of the sequence of a Gaussian type, is offered. Embedding is done by applying the wavelet transform and converting the cosine, and then the signal will be restored. The reliability of this method is carried out by means of different tampontypes using an attacker. In addition, the proposed scheme also acts with the ambiguity of the attack [15].

A new algorithm for marking HDR images, resistant to TMOs implemented in the study. Two important concepts of spatial activity, i.e. activity of reliability and activity of perception, respectively, are determined to characterize spatial diversity of reliability and unacceptiveness of the tone-mapped images. In addition, the hierarchical intensity of embedding and hybrid perceptual masks are designed to enhance the non-acceptance and reliability of the HDR image watermarks. Experiments with numerous HDR and TMOs images, and the results show that the proposed algorithm surpasses modern water labeling algorithms in terms of the unadulterity, reliability, and embedding capabilities [18].

A new water labeling scheme of images using a entropy host and texture image watermark was investigated. The entropy is evaluated for discrete wavelet conversion factors to the image of the host. The highest entropy subgroup is chosen to embed a texture of a watermark. The convert Arnold is used to textutization a watermark and the texture is chosen randomly. The security level is increased by embedding the texture instead of the watermark. Various experiments are conducted to analyze proposed entropy-based selection, linear

weight estimation, embedding watermarks and watermark-host extraction. Experimental results prove that this method efficiently extracts watermark and host image with better unadulterity [23].

One of the steps to maintain information security is to use steganography techniques. Steganography image – The development of science from steganography. The first part of this article explains what steganography means in general. Then proceed to the explanation of the study of the use of steganography on MP3 (MPEG Layer3) Audio. Methods can be used in Steganography methods on MP3 files, although the least significant bit (LSB), the spread of the spectrum, the direct spectrum of the sequence (SSSS), PARITY encoding, phase encoding, echo concealment, DCT and DWT [28].

Social communication and interactions between users of online social networking (OSNs) has certain security properties, but the real end-to-end security can not be achieved. In this article, we are developing a secure channel between two UN friends using social messaging. Confidentiality is achieved through the detection of communication based on distributed Nizkoentropìï steganography, thus, the implementation of security controls for the content of the VIS-à-VIS untrusted un provider. Implement a prototype solution in the form of Android mobile app [29].

From the above study literature survey it is found that there is still a need of proper steganography techniques. The values for PSNR and compression ratios needs to be improved and the technique need to simple and without much flaws as seen in the previous research work. The researchers were mostly worked in programming tools that are not according to the recent developments.

SYSTEM MODEL DESCRIPTION

The proposed work will be carried out using the implementation of the steganography technique for encoding and decoding multimedia data. This multimedia data will be hidden in cover image for generating embedded image. This embedded image will be compressed using the compression technique. The original multimedia data and the cover image used for generating embedded image will be recovered using decoding and lossless data compression technique. The evaluation of the proposed approach will be carried out using parameters such as PSNR and compression ratio.

Tools used

In the present work the programming is carried out in the python programming language which is one of the advanced tool to carry out research work. This is also a open source tool and there is no any kind of legal version required so that the work is absolutely a proper research work. This may required for hardware as well as software requirements.

Hardware Requirement:
Computer: Processor: Intel I3
Operating System: Windows 8
Memory, Graphics & Storage: 4GB DDR4 RAM (1x4), Intel HD Graphics 620, Storage: 256GB M.2 SSD
Software Requirements:
Operating System: windows 8
Coding Language : PYTHON 3.7.4

METHODOLOGY:

The proposed work is carried out using the implementation of the steganographic technique for encoding multimedia data. Firstly it will select which multimedia data to be hidden either it will text, image or audio and this will be referred as secret data. Then size of data to be calculated for hiding in cover image. Then LSB technique will be used for encoding this multimedia data, the resultant image will be embedded image. After that lossless compression technique will be applied. This compressed image will then send to open channel. Secret data will obtain by applying decompression technique and decoding. After decompression, embedded image will obtained. By decoding this embedded image, secret data and cover image will obtained. The following figure 1 shows the encoding in steganography while figure 2 and 3 decoding in steganography.

In proposed methodology secrete data in terms of text / image / audio will hide in image which is the cover medium. Firstly it will calculate the size of secret data which is to be hidden in the cover medium. The cover medium is image then the resultant image becomes resultant image which is to be compressed for transmission. The compressed image will be in jpg format, png format or bmp format. This is generalized encoding process. While in the decoding process of proposed method the compressed image is to be selected and it can be in format (bmp / png) after that it will be decompressed by using decompression technique. After decompression decoding procedure starts. After decoding secrete information and cover image get separated. If jpg image selected as compressed image, after decompression decoding process starts which gives cover image and the secret information will be received where data will be losses because jpg compression technique is lossy compression technique.

Tool : Pycharm 2019.

DESIGN AND IMPLEMENTATION

A. Problem Definition:

It is observed that problem in providing security from intruder and the security from the data hackers. There are many methods used by the researchers which are based upon encryption as well as decryption. In case of encryption method used by the other researchers it is found that hacker come to know that data is encrypted, so he try to decrypt it. Therefore to remove such flaws the proposed work is carried out for steganography in text, image as well as audio. The flaws in the technique observed by the researchers are then tried to solved in the present work through the programming in Python

language.

B. Steganography in text:

First of all we need to enter value among 1 or 2 or 3. enter 1 to enter text, 2 for image and 3 for audio, after that we have to check if code one is there if yes then encode text otherwise

If the image size (cover image) is greater than (no. Of letters X 8/6) then initially select i=0 ('i' represents number of rows), j=0 ('i' represents number of rows), rgb=0 ('rgb' is color coordinate) & l=0 ('l' is number of bits in text message). In the next step hide two bits of text message in R of rgb coordinate of ith and jth pixel. Then it shall check the length of message which is l=lmax (lmax is last bit of text message). If it is true then stop the procedure of encoding otherwise hide another two bits of text message in G of rgb coordinate of ith and jth pixel.

Then we need to check if the value of rgb is equal to 3, if not found then hide another 2 bits of text message in B of rgb coordinate of ith and jth pixel. If the value is true then go for next pixel of first row i.e. j=j+1 and the value of rgb is set to zero. Again it is to be checked that the value of j has become maximum or not i.e. the last column of first row. If not found then hide 2 bits of text message in rgb coordinate of ith and jth pixel. Otherwise go for next row which is i=i+1 and j=0. The iteration is continued till the end of length of message. If the total bits of text message are finished then the encryption is completed. When all the bits of text message are hidden in image or cover image then the embedded image is known as stego image.

The following steps to be followed for the decoding

C. Steganography in image:

The following flowchart shows the encoding of image by LSB technique:As stated earlier in the steps of steganography in text it is needed to enter the number '2' for the image encoding. Now, calculate the size of cover image and message image. Then check 1 pixel of message image is greater than / equal to 4 pixel of cover image, if found 'no' then select the cover image of higher size otherwise select i=0 (i- number of rows in the cover image), j=0 (j - number of columns of cover image), rgb_cov=0 (rgb_cov - color coordinate of cover image) rgb_mess=0 (rgb_mess -color coordinate of message image), im,jm=0 (number of rows and columns of message image) and bit_count=0 (1bit_count=2 bits, there are 8 bits then there are total 4 bit_count), this is for first pixel.

Else go for next pixel of message image and bit_count=0. then it will check if jm=jmmax. If found no then again hide two bits of message image to rgb of cover image. Else go for next row of message image and jm=0 and then it will check whether im=immax,immax is the last row of message image. If found no then again hide two bits of message image to rgb of cover image. Else terminate the encoding process and it will show the embedded image / stego image.

check the condition of code is entered two or not. Now for text encoding convert the text into bit stream, then calculate the size of text stream and image. Then we have to check the image size which is cover image should be greater than (no. Of letters X 8/6). if the image size is less than (no. Of letters X 8/6) then select higher image size.

steganography in text:

For the decoding of text message from cover medium select stego image then initialize i=0, j=0, rgb=0, st=0 ('I,j,rgb is as explained in the above paragraph','st' is the variable which receives text bits from image by joining with temporary variable). Now, separate last two bits of R coordinate of rgb from stego image and assign it to temporary variable. Then proceed for next coordinate. Then we have to check if rgb is equal to 3, if found no then again separate last 2 bits of G coordinate of rgb and assign to temp variable. Then proceed for next coordinate otherswise check for terminating character (i.e. the length of message) if the value is true. Again check the terminating character, if found true then stop encoding otherwise go for next pixel or next column and rgb is set to zero. Then separate last two bits of R coordinate of rgb and assign to temp variable. Then again check the value of rgb is 3 or not, if found no then separate last two bits of g coordinate of rgb and assign to temp variable. Otherwise go for next pixel and rgb set to zero. Check the terminating character, if found yes then display the text otherwise check whether j=jmax. If found 'no' then again separate last 2 bits of rgb coordinate otherwise go for next row. Again separate last two bits of rgb and assign to temp variable.

Now, hide two bits message image to r coordinate of rgb of cover image. Then it will check rgb_cov =3, if found yes then go for next pixel of cover image j=j+1 and rgb of cover image is set to zero. Again check whether j=jmax(jmax - the last column of first row). Otherwise again hide two bits of message image to rgb of cover image. Otherwise go for next row and j is set to zero and again hide two bits of message image to rgb of cover image. Then it will check rgb_cov=3, if found no then check the bit count whether it is increased by two or not. Then check if bit_count=5, if found no then again hide two bits of message image to rgb of cover image. Otherwise it will go for next coordinate of rgb of message image and bit_count =0. It will check rgb_mess=3, if found no then again hide two bits of message image to rgb of cover image.

The following steps to be followed for the decoding of image: Now, select the stego image as input, initially i=0, j=0, rgb=0, i_mess=0, bit_count=0, i_mess=0, rgb_mess=0 (the parameters are already explained earlier). This is for first pixel, Then separate two bits of r coordinate of rgb from cover image and assigned to message image. It will check if rgb=3, if found yes then go for next pixel of cover image and rgb set to zero. Then it will check if j=jmax, if found no Then separate two bits of next coordinate of rgb from cover image and assigned to

message image. Else go for next row and j=0 and again then separate two bits of next coordinate of rgb from cover image and assigned to message image.

Then it will check rgb=3, if found no then bit_count increased by two and it will check bit_count=5. If found no then separate two bits of next coordinate of rgb from cover image and assigned to message image. Else go for next coordinate of rgb of message image, Is rgb_mess=3? If found no then separate two bits of next coordinate of rgb from cover image and assigned to message image. Else go for next column and rgb_mess=0 and it will check j_mess=j_messmax. If found no separate two bits of next coordinate of rgb from cover image and assigned to message image. Else go for next row and j_mess=0 and it will check i_mess=i_messmax,and j_mess=j_messmax. If found no then separate two bits of next coordinate of rgb from cover image and assigned to message image and if found yes then terminate the process. This will show the cover image and message image.

D. Steganography in audio:

For the encoding sound file select the input image as cover image and audio file as secret data. After choosing audio file it will play the sound so that it is come to know which one is the secret file. Then enter the key, now convert key and sound data into bit stream. For first pixel initialized I=0, j=0 & i_aud=0, j_aud=0 which is i & j coordinate of audio bit stream, then sound file is stored in 'y' and bit_count set to zero. Now mask two bits of pixel in image thne separate two bits of y and combine it with r coordinate of rgb in pixel of image. Next it will check rgb=3, if found yes it will go for 2nd column, j=j+1 & value of rgb is set to 0. Then it will check the value of i which is maximum or not, if found no then it will again separate two bits of y and combine it with next coordinate of rgb. Otherwise it will go for next row and value of j=0. now it will check the value of i which is maximum or not if found no then it will again separate two bits of y and combine it with next coordinate of rgb else stop and display image. Now if the value of rgb is not equal to 3 then bit_count is increased by 1. then it will check if bit_count=32 if found no then it will again separate two bits of y and combine it with next coordinate of rgb, else it will go for 2nd column and bit_count=0. Then it will check if i_aud=2, if found no then again it will again separate two bits of y and combine it with next coordinate of rgb else it will go for first column of next row. Now it will check if i_aud=i_audmax i.e it will reach to the last row or not if found no then again it will again separate two bits of y and combine it with next coordinate of rgb otherwise display the image & terminate process.

Select the stego image as input, initialize i=0 (i- number of rows in the cover image), j=0 (j - number of columns of cover image), rgb=0 (rgb - color coordinate of cover image), i_aud=0 (i_aud-represent rows of audio bit stream), j_aud=0 (j_aud-represent columns of audio bit stream), y represents to play the sound file, bit_count=0. Now separate last two bits from r coordinate of rgb of cover image and combine it with y, then it will check rgb=rgb+3. It will check if rgb=3, if found yes then go for next pixel of cover image and rgb set to zero. Then it will check if j=jmax, if found no Then separate last two bits from next coordinate of rgb from cover image and combine it with y. Else go for next row and j=0, now check i=imax, if found no then separate last two bits from next coordinate of rgb of cover image and combine it with y. Else play 'y' and terminate the process.

When rgb is not equal to 3 then bit_count is increased by one and it will check bit_count=5, if found no then then separate last two bits from next coordinate of rgb of cover image and combine it with y. Else go for next column of audio, Is j_aud=2? If found no then separate last two bits from next coordinate of rgb of cover image and combine it with y. Else it will go for next row and it will check i_aud = i_audmax? If found no then separate last two bits from next coordinate of rgb of cover image and combine it with y. Else play 'y' and terminate the process and show the cover image.

EXPERIMENTAL RESULTS AND DISCUSSION

As steganography is becoming more widely used in digital image processing. The proposed method involves LSB technique of steganography for encoding of text, image or audio in cover image. The experiments are performed in Python programming language which is open source software for programming and gives better results than other programming tools. Below are the results obtained after the experiment i.e. hybrid approach for multimedia data security with embedded image compression in Python.

A. Case I: Encoding & decoding of text

Once the embedded image is displayed then it shows that the encoding is completed then the next number is to be entered for further process.

B. case II: Encoding & decoding of image

The image steganography started, after display of hidden message then it shall ask to select the next number for further process. Here we will use two images, one is for cover image and other for secret image. Select number '3' for encoding of the image. After entering '3' then it will ask to enter the cover image name.

For the decoding process it will enter '4', after entering 4 then it will show the decoded image in which it show the display of cover image along with recovered secret message image. The image obtained is the compressed image in the png format. The decoded image obtained from jpg format is blurred (loss of information) due to lossy compression technique. This is the complete process of image steganography.

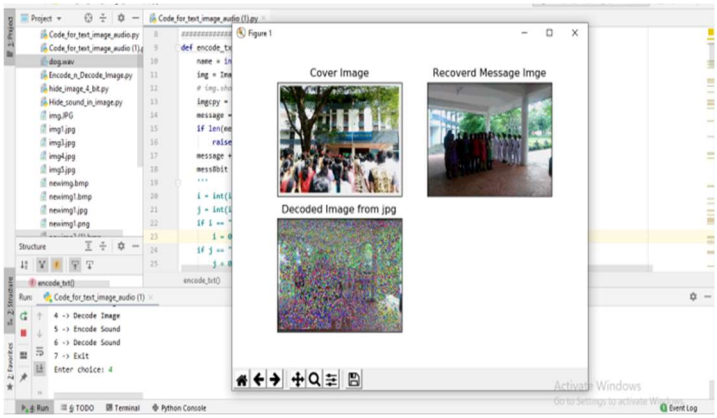

Fig. 1: Output of decoding of image

C. Case III: Encoding & decoding of audio

Now, the process of audio steganography started, here the cover medium is image and the secrete data is sound file. After closing the image then it will ask to select the number for further process.

Once the process of encoding is finished then it will show the embedded image in which the first image is cover image and second one is message image. In this message image masking of lsb is carried out and then the sound file is added in the cover image. The output of audio steganography is shown in the figure 13.

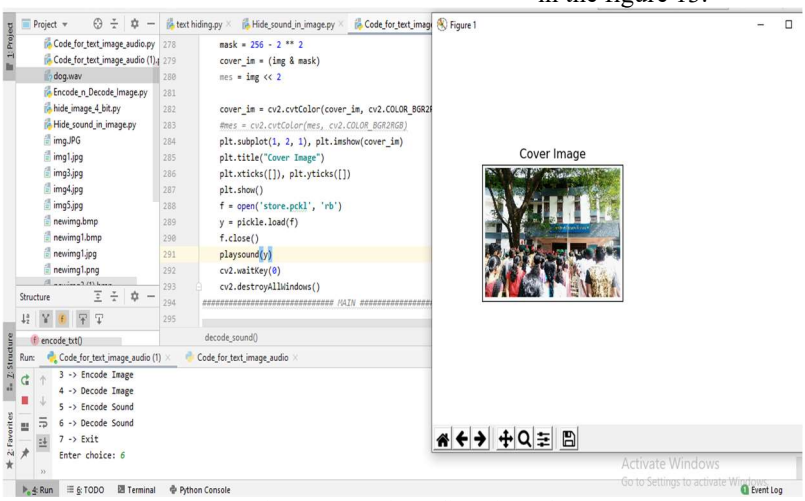

Fig. 2: output of encoding of sound

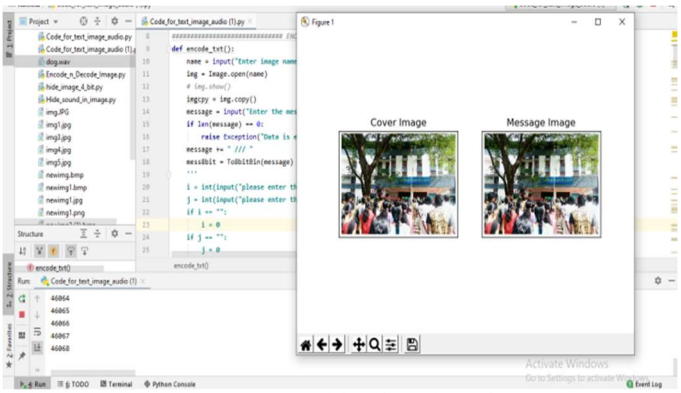

Fig. 3: Output of decoding of sound

Once the sound of dog barking is hear then it will ask to select the number for next process. If the number '7' is entered then the process is finished with exit code '0' as shown in the figure 14.

D. performance parameters:

The results of the programme are obtained and the

parameters to check the performance are studied and mentioned below.

1) compression ratio:

The formula to calculate the compression ratio is mentioned below:

$$\text{Compression ratio} = \frac{\text{Compressed Image}}{\text{Original Image}}$$

Compression ratio for text

$$\text{Compression ratio} = \frac{\text{Embedded Image}}{\text{Cover Image}} = \frac{1100kb}{6751kb} = 0.16$$

Compression ratio for Image

$$\text{Compression ratio} = \frac{\text{Embedded Image}}{\text{Cover Image}} = \frac{606kb}{901kb} = 0.67$$

Compression ratio for Audio

$$\text{Compression ratio} = \frac{\text{Embedded Image}}{\text{Cover Image}} = \frac{438kb}{901kb} = 0.48$$

2) Peak signal to noise ratio (psnr value):

This PSNR value is directly calculated with the help of Pycharm progamming tool and the values are mentioned below

TABLE 1
PSNR VALUE FOR TEXT

Original Image	Embedded Image	PSNR value
img1.jpg	newimg1.jpg	37.72
	newimg1.png	93.38
	newimg1.bmp	93.38

TABLE 2
PSNR VALUE FOR IMAGE

Original Image	Embedded Image	PSNR value
Img3.jpg	stego.jpg	32.39
	stego.png	32.13
	stego.bmp	32.13

TABLE 3
PSNR VALUE FOR AUDIO

Original Image	Embedded Image	PSNR value
Img4.jpg	stego.jpg	39.3
	stego.png	42.59
	stego.bmp	42.54

The comparison of compression ratio and PSNR value with other researchers are also mentioned in the below table 4.

TABLE 4
COMPARATIVE ANALYSIS OF PARAMETERS

Sr. No.	Authors	Parameters					
		Compression Ratio			PSNR value		
		text	image	audio	text	image	audio
1	Ahmed Kadem Hamed Al-Saedi [7]	0.21	-	0.51	66.98	-	41.15
2	Arshiya Sajid Ansari [1]	0.26	-	-	71.25	-	-
3	M. Parthasarathi A [9]	-	0.56	0.55		31.1	17.27
4	M.J.Thenmozhi [8]	0.35	-	-	85.52	29.34	-
5	Markus Mainberger [33]	0.19	-	-	64.52	-	-
6	Khan Muhammad [6]	-	0.65	-	-	31.5	-
7	Proposed Method	0.16	0.67	0.48	93.38	32.13	42.54

From the above table it can be observed that the proposed method gives good results as compared to the existing work in terms of parameters of compression ratio and PSNR values. Compression ratio found to be lowest in case of proposed work while PSNR value found to be highest in case of proposed work in comparison with the existing work. Some of the data is not available in the existing work and therefore no data is entered.

It is found that the proposed work has shown the better

results as compared to the existing work in terms of parameter. The charts showed some of the data which is not available for text as and that means the steganography technique is used in proposed work is used for text, image and audio, such type of work not carried out in the existing work.The table no. 5 gives the sizes in kb for compression ratio is mentioned as below:

TABLE 5
SIZES IN KB FOR COMPRESSION RATIO

Sr. No.	Authors	Size (kb)					
		text		image		audio	
		embedded image	Cover Image	embedded image	Cover Image	embedded image	Cover Image
1	Ahmed Kadem Hamed Al-Saedi [7]	1264	6021	-	-	522	1024
2	Arshiya Sajid Ansari [1]	1800	6951	-	-	-	-
3	M. Parthasarathi A [9]	-	-	339	606	468	851
4	M.J.Thenmozhi [8]	2702	7721	-	-	-	-
5	Markus Mainberger [33]	1282	6751	-	-	-	-
6	Khan Muhammad [6]	-	-	665	1024	-	-
7	Proposed Method	1100	6751	606	901	438	901

The table shows various image size as per taken in the different existing work to calculate compression ratio.

CONCLUSION:

In this work the powerful programming tool i.e. Python is used deliberately in order to obtain better results in the existing work by using LSB technique. The steganography on text, image and Audio files were successfully implemented. Also in this work a new approach is proposed of text, image and audio steganography. From the experiential analysis depicted in table it is found that the values of compression ratio and PSNR significantly improved as compared to the existing work. Therefore it be observed that the proposed work entitled 'hybrid approach for multimedia data security with embedded image compression' is a better approach to send the files in terms of data security in various places like banks, military, etc.

REFERENCES

[1] Ansari, A.S., Mohammadi, M.S. and Parvez, M.T., 2019. A comparative study of recent steganography techniques for multiple image formats. International Journal of Computer Network and Information Security, 11(1), p.11.

[2] Ambika, Biradar, R.L. and Burkpalli, V., 2019. Encryption-based steganography of images by multiobjective whale optimal pixel selection. International Journal of Computers and Applications, pp.1-10.

[3] Danny Adiyan, Z., Purboyo, T.W. and Nugrahaeni, R.A., 2018. Implementation of Secure Steganography on Jpeg Image Using LSB Method. International Journal of Applied Engineering Research, 13(1), pp.442-448.

[4] Ramacharla, n., priya, s. and logashanmugam, d., 2018. a secure image steganography for speech data hiding in digital images. international journal of pure and applied mathematics, 118(17), pp.509-522.

[5] Aagarsana, B.G., Anjali, T.K. and Kirthika, M.S.S., 2018. Image Steganography using secured force algorithm for hiding audio signal into colour image. IRJET access, 5.

[6] Hussain, M. et al., 2018. Image steganography in spatial domain: A survey. Signal Processing: Image Communication, Volume 65, pp. 46-66.

[7] Al-Saedi, A.K.H., 2016. A method to hide text in image. Journal of Misan Researches, 12(24), pp.11-23.

[8] Thenmozhi, M.J. and Menakadevi, T., 2016. A New Secure Image Steganography Using Lsb And Spiht Based Compression Method. International Journal of Engineering Research & Science (IJOER), 2(3), pp.81-85.

[9] Parthasarathy, M.B. and Srinivasan, B., 2015. Increased security in image cryptography using wavelet transforms. Indian Journal of Science and Technology, 8(12), pp.1-8.

[10] Singh, P., 2016. A comparative study of audio steganography techniques. J Int Res J Eng Technol, 3(4), pp.581-585.

[11] Shrivastava, A. and Singh, L., 2016. A new hybrid encryption and steganography technique: a survey. International Journal of Advanced Technology and Engineering Exploration, 3(14), pp.9-14.

[12] Tim Bruylants, Adrian Munteanu, and Peter Schelkens, "Wavelet Based Volumetric Medical Image Compression," Elsevier Signal Processing: Image Communication pp. 112–133, 2015.

[13] Fahima Tabassum, Md. Imdadul Islam, and Mohamed Ruhul Amin, "A Simplified Image Compression Technique Based on Haar Wavelet Transform," 2nd Int'l Conf. on Electrical Engineering and Information & Communication Technology (lCEEICT) 2015, Jahangirnagar University, Dhaka-I 342, Bangladesh, 21-23 May 2015.

[14] Senthilkumar, M. & Mathivanan, V., 2016. Performance Analysis of Data Compression Techniques for Multimedia Data Hiding. International Journal of Emerging Research in Management &Technology, 5(7), pp. 42-49.

[15] Li CT, Lee CW, Shen JJ. A secure three-party authenticated key exchange protocol based on extended chaotic maps in cloud storage service. In international conference on information networking (ICOIN) 2015 (pp. 31-6).IEEE.