International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Pixel Masking Based Data Hiding Technique (PMDHT)

Vivek Jaladi[1], Avinash[2]

[1]H.O.D of E&CE, Lingaraj Appa Engineering College, Bidar, Karnataka, India

[2]Pursuing M.Tech (VLSI & ESD), Lingaraj Appa Engineering College, Bidar, Karnataka, India

Abstract: ***In this paper a pixel masking based data hiding technique (PMDHT) has been presented. The main purpose of this paper work is to hide secret message/image/video into source image/video. Here we are embedding secret message/image/video into image/video by choosing standard 3 x 3 mask in a row major order in which each bits of the secret image/video are embedded in the single bit position of the source image/video. A message digest MD5 is generated from the secret image/video and to be inserted in the embedded image/video for additional security purpose. We then perform embedding all this (Secret and MD5) in the source image/video using our proposed technique. Result is analyzed in terms of SNR.***

Keywords: PMDHT, Pixel masking, Message Digest, MATLAB, and GUI.

1. Introduction

At present digital media and internet are getting more popular. Internet is one of the rapidly growing technologies in the present era. Since it is a public network, hence the requirement of secure transmission is also increased. Nowadays it's really a big challenge task to hide the data across the network. Data hiding is the process of embedding information into digital content without changing its original perception. By embedding the information within the source data we can justify the authentication, identification and ownership of the data. Data hiding is the art and science of hiding data in such a way that, no one can realizes there is a hidden secret data except sender and the intended recipient. There are several tools & techniques to protect the originality of the image/video. In this paper PMDHT technique has been proposed to perform the task.

J.K Mandal et al. [1] proposed paper by embedding an authenticated message/image in to video/color image. The bits from authenticating message/image are embedded under each byte of the source image by choosing a standard 3 x 3 masking in a row order. In this proposed technique the authentication process is performed by mathematical operations on 3 x 3 masks to insert bits into the source image byte. The insertion is done through mask selections in row order for the entire image bytes so that it provides extra level of security. A message digest MD5 also been generated from authenticating message/image and it has to be inserted into the source image for additional security purpose. They compared there performance with S-Tools technique and concluded that there proposed paper performs better.

Shikha et al. [2] proposed paper Steganography: The art of hiding text into image using matlab. They proposed a method that hides the secret messages into the image using matlab. Matlab is not only programming software but also a programming environment. They encrypt the text using play fair method. First the text is encrypted using fair play method. Then they encode the encrypted text in to the image file using matlab. At the recipient end they decode the image file and extraction of text using matlab code. Finally they decrypt the text using play fair method. In their proposed technique capacity of data hiding to hide the secret messages are high.

Nammer N. EL-Eman et al. [3] proposed a steganography algorithm to hide the large amount of data with high security. They implemented new algorithm for hiding a large amount of data file into color bitmap image. The data file may be image file, audio file or text file. They used adaptive image segmentation and filtering with bits replacement on the suitable pixels. These pixels are not selected sequentially but selected randomly. The proposed algorithm allows hiding data inside other data with hopes that except intended sender and recipient no one can think to there is a hidden content in that. The algorithm was described by pseudo code so that it was possible to design a steganography algorithm for hiding a large amount of data into bitmap image. They compared their result with S-Tools algorithm and shown that there proposed work embeds a large amount of data with high quality output.

P. Moulin et al. [4] proposed model for image watermarking and data hiding. They used recent theoretical results to characterize the fundamental capacity limits of data hiding and image watermarking systems. They considered wavelet statistical, auto aggressive, and block-DCT models for computing data hiding capacity for compressed as well as for uncompressed host image sources. The fundamental capacity is determined by the statistical model that is used for the host image.

C. Y. Lin et al. [5] they proposed an effective technique for image authentication which prevents from malicious manipulations but meantime it allows JPEG lossy compression. Their proposed method distinguishes the malicious manipulation from JPEG lossy compression regardless of the number of compression iterations or compression ratio. They describe adaptive methods to handle distortions introduced by manipulations such as integer rounding, image filtering or image enhancements. They also

presented excremental results and theoretical results to demonstrate the efficient of the technique.

2. The Technique

2.1 Basic Concept

In our presented work we are going to embed the secret message or secret image or secret video into source image or source video. A standard mask of size 3 x 3 is selected from the source image matrix in row order and single bit is to be inserted into each byte from the secret message or image. The position where secret bit has to be inserted in calculated through mathematical function, which depends on the position of the pixel within the mask. A 128-bits MD5 key is generated from the secret message or image or video and it has to be inserted with the embedded image or video. Figure 2.1 shows the generic block diagram of embedding secret message or image or video into source image or video. Here source data may be image or video, while secret data might be message or image or video format.

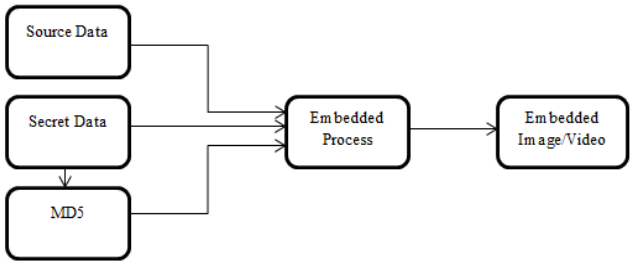

Figure 2.1: Generic block diagram for embedding secret data into source data.

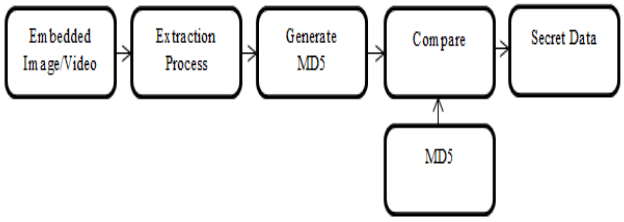

Figure 2.2: Generic block diagram for extraction process.

Figure 2.2 shows the generic block diagram for extraction process. Here an embedded image or video which was generated at the sender side is applied as input to the extraction process. In the extraction process secret data is extracted and MD5 is generated from that. The MD5 key generated by extraction process and that was generated at the sender side are compared. If both matches, then we can trust that the recipient is an authenticated user. Then user can extract the secret data easily. Suppose if MD5 generated at the receiver side does not match with the MD5 that was generated at the sender side, then we can say that the user is unauthenticated. The following are the different task which we presented in this work using PMDHT technique:

- Hiding secret message into source image.
- Hiding secret image into source image.
- Hiding secret image into source video.
- Hiding secret video into source video.

All these are explained briefly in below sections with their corresponding block diagrams.

2.2 Hiding Secret Message into Source Image

In this section we are going to discuss about hiding secret message into source image and extraction of secret image from embedded image. Figure 2.3 shows the block diagram for embedding secret message into source image. A standard mask of size 3 x 3 is selected from the source image matrix in row order and single bit is to be inserted into each byte from the secret message. The insertion position of the secret bit is calculated through a mathematical function, which depends on the position of the pixel within the mask. At the embedding process, the secret message gets embed within the source image. In addition a MD5 is generated from the secret message and it has to be inserted into the embedded image. And this embedded image can be transmitted across the network.

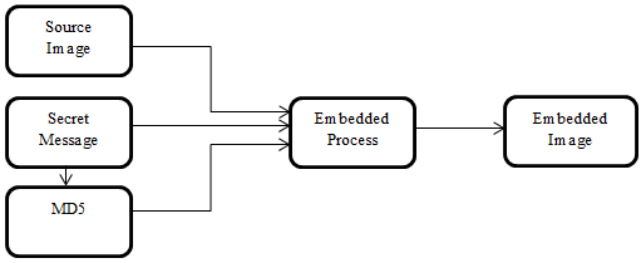

Figure 2.3: Embedding of secret message into source image.

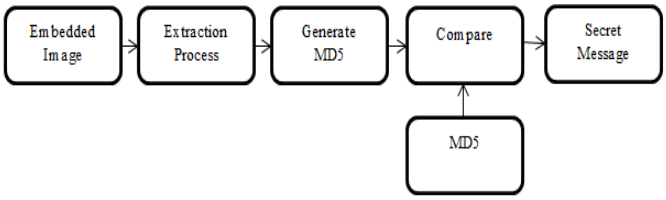

Figure 2.4: Extraction of secret message from embedded image.

While figure 2.4 shows the block diagram for extraction of secret message from the embedded image. Here at the extraction process, the embedded image which was generated at the sender side is used as input at the receiver side. After completion of extraction process, MD5 is generated for the output of the extraction process. Then this MD5 is going to compare with the MD5 that was generated at the sender side. If both matches, then we can say the user is an authenticated user. Otherwise, he is an unauthenticated user. Finally, an authenticated user can easily extract the secret message.

2.3 Hiding Secret Image into Source Image

In this section we are going to discuss about hiding secret image into source image and extraction of secret image from embedded image. Figure 2.5 shows the block diagram for embedding secret image into source image. A standard mask of size 3 x 3 is selected from the source image matrix in row order and single bit is to be inserted into each byte from the secret image. The insertion position of the secret bit is calculated through a mathematical function, which depends

on the positions of the pixel within the mask. At the embedding process, the secret image gets embed within the source image. In addition a MD5 is generated from the secret image and it has to be inserted into the embedded image. And this embedded image can be transmitted across the network.

While figure 2.6 shows the block diagram for extraction of secret image from the embedded image. Here at the extraction process, the embedded image which was generated at the sender side is used as input at the receiver side. After completion of extraction process, MD5 is generated for the output of the extraction process.

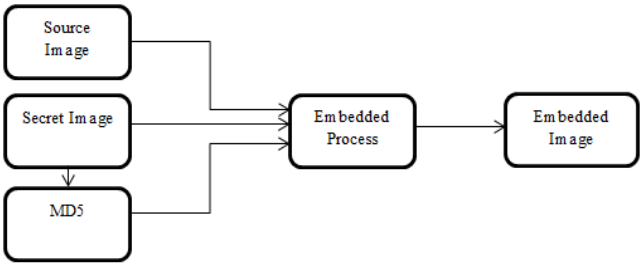

Figure 2.5: Embedding of secret image into source image.

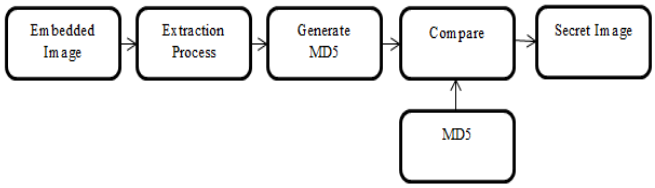

Figure 2.6: Extraction of secret image from embedded image.

Then this MD5 is going to compare with the MD5 that was generated at the sender side. If both matches, then we can say the user is an authenticated user. Otherwise, he is an unauthenticated user. Finally, an authenticated user can easily extract the secret image.

2.4 Hiding Secret Image into Source Video

In this section we are going to discuss about hiding secret image into source video and extraction of secret image from embedded video. Figure 2.7 shows the block diagram for embedding secret image into source video. For source video initially we have read out the number of frames the video file has, and then we have to convert it to images. A standard mask of size 3 x 3 is selected from those converted images matrix in row order and single bit is to be inserted into each byte from the secret image. The insertion position of the secret bit is calculated through a mathematical function, which depends on the positions of the pixel within the mask. At the embedding process, the secret image gets embed within the converted images. Again these images are converted back to frames. In addition a MD5 is generated from the secret image and it has to be inserted into the embedded video. And this embedded video can be transmitted across the network.

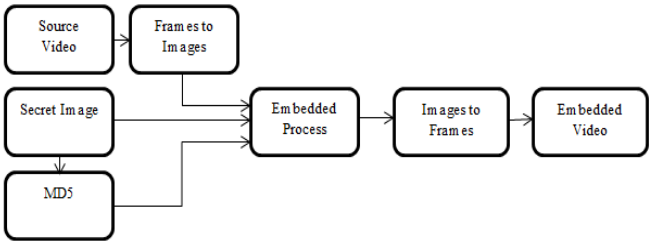

Figure 2.7: Embedding of secret image into source video.

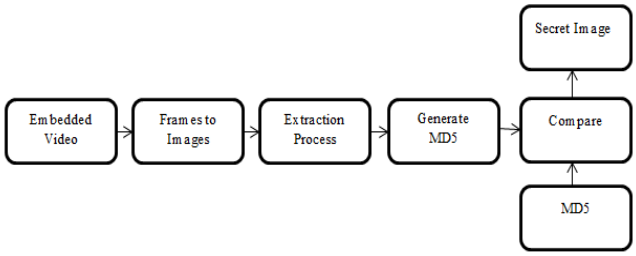

Figure 2.8: Extraction of secret image from embedded video.

While figure 2.8 shows the block diagram for extraction of secret image from the embedded video. Here at the extraction process, the embedded video which was generated at the sender side is used as input at the receiver side. Before beginning of the extraction process, we have to read out the frames of the embedded video and then those frames to be converted to the images. After completion of extraction process, MD5 is generated for the output of the extraction process. Then this MD5 is going to compare with the MD5 that was generated at the sender side. If both matches, then we can say the user is an authenticated user. Otherwise, he is an unauthenticated user. Finally, an authenticated user can easily extract the secret image.

2.5 Hiding Secret Video into Source Video

In this section we are going to discuss about hiding secret video into source videos and extraction of secret video from the embedded videos. Figure 2.9 shows the block diagram for embedding secret video into source video. Before beginning of the embedding process both secret video and source video has to convert to images depending on their respective frames. A standard mask of size 3 x 3 is selected from those converted images matrix in row order and single bit is to be inserted into each byte from the secret converted image. The insertion position of the secret bit is calculated through a mathematical function, which depends on the positions of the pixel within the mask. At the embedding process, the secret converted image gets embed within the converted images. Again these images are converted back to frames. In addition a MD5 is generated from the secret image and it has to be inserted into the embedded video. And this embedded video can be transmitted across the network.

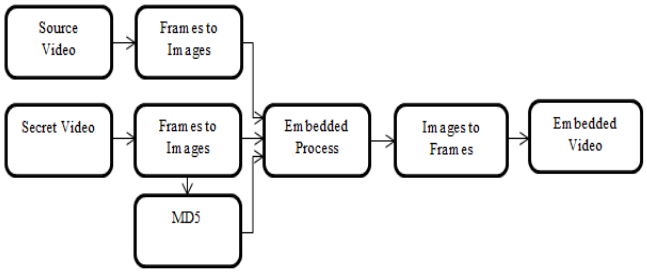

Figure 2.9: Embedding of secret video into source video.

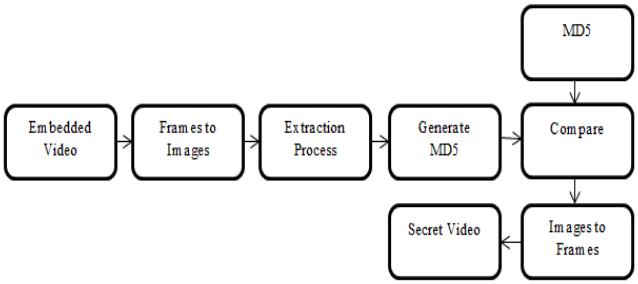

Figure 2.10: Extraction of secret video from embedded video.

While figure 2.10 shows the block diagram for extraction of secret video from the embedded video. Here at the extraction process, the embedded video which was generated at the sender side is used as input at the receiver side. Before beginning of the extraction process, we have to read out the frames of the embedded video and then those frames to be converted to the images. After completion of extraction process, MD5 is generated for the output of the extraction process. Then this MD5 is going to compare with the MD5 that was generated at the sender side. If both matches, then we can say the user is an authenticated user. Otherwise, he is an unauthenticated user. On successful matching of the MD5, those images are converted back to framed which reconstructs the secret video.

3. Results & Discussions

In this section we are discussed about our presented work result. As we are using MATLAB R2013a software for implementing out presented work. Our results are shown under GUI window.

In section 3.1 we are hiding secret message into the source image. We selected 'world cup.bmp' as source image and 'we will take it back' as our secret message. This secret message is embedded in source image. This is shown in figure 3.1. We extracted this message by selecting the embedded image as input at the extraction side. This is shown in figure 3.2.

In section 3.2 we are hiding secret image into source image. We selected 'Dhoni.bmp' as source image and 'world cup.bmp' as secret image. The 'world cup.bmp' image is embedded in the 'Dhoni.bmp' image. This we can see in figure 3.3. We extracted the secret image by selecting embedded image as input at the extraction side. This is shown in figure 3.4.

In section 3.3 we are hiding an image into source video file. We selected 'dhoni.avi' as source video and 'world cup.bmp' as secret image. Here the secret image 'world cup.bmp' gets embedded in the source video file. The embedded video can be seen in figure 3.5. We extracted this secret image by selecting the embedded video as input to the extraction side. This is shown in figure 3.6.

In section 3.4 we are hiding video file into another video file. i.e., we are hiding secret video into source video file. We selected 'msd.avi' as source video and 'modi.avi' as secret video. The secret video gets embedded in the source video file and that can be seen in figure 3.7. We can extract this secret video from the embedded video by selecting embedded video as input at the extraction side. This is shown in figure 3.8.

3.1 Embedding a secret message into source image and extraction of secret message from the embedded image:

Figure 3.1: Embedding of message into an Image.

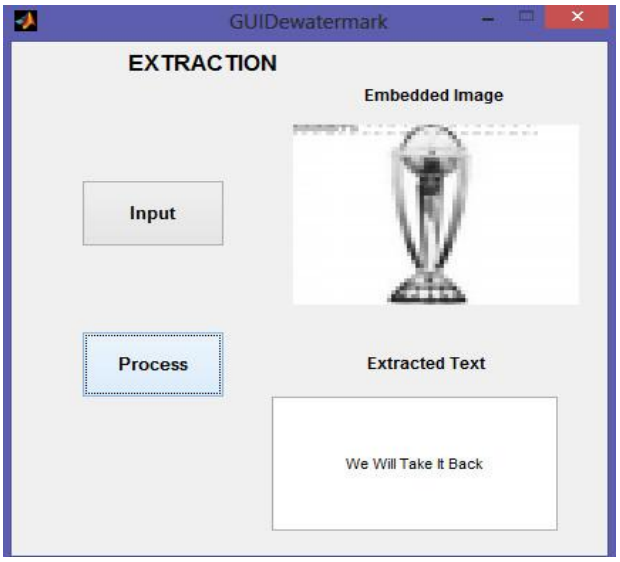

Figure 3.2: Extraction of message from Image

3.2 Embedding a secret image into source image and then extraction of secret image from the embedded image:

Figure 3.3: Embedding of an Image into an Image.

Figure 3.4: Extraction of an Image from Image.

3.3 Embedding a secret image into source video and then extraction of secret image from the embedded video:

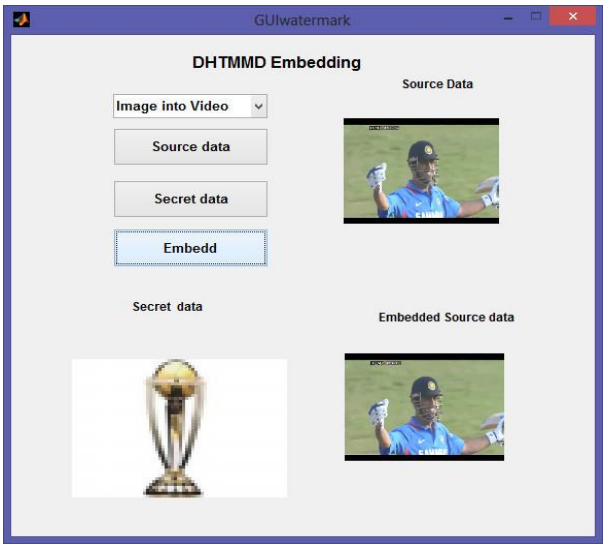

Figure 3.5: Embedding of an Image into Video.

Figure 3.6: Extraction of Image from Video.

3.4 Embedding a secret video into source video and then extraction of secret video from the embedded video:

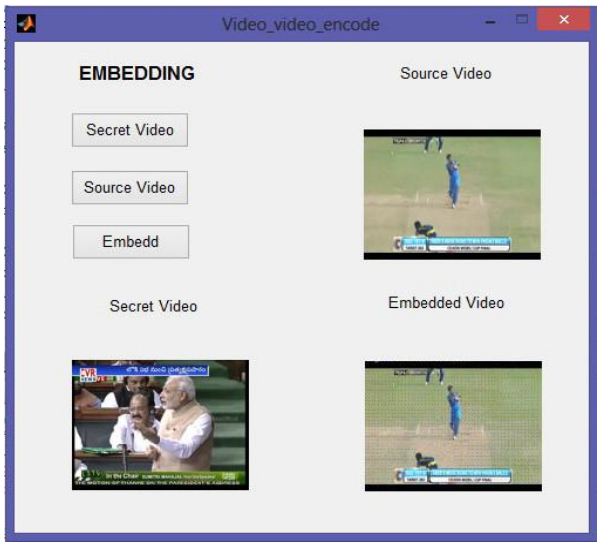

Figure 3.7: Embedding Video into Video.

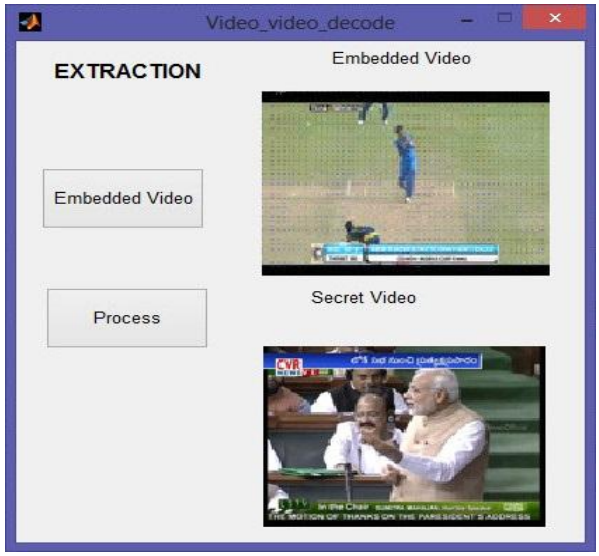

Figure 3.8: Extraction of Video from Video.

3.5 SNR Analysis

Figure 3.9 shows the SNR of secret image, embedded image and that of extracted image using PMDHT. For the purpose of analyzing the SNR, we are embedding the secret image 'world cup.bmp' into the source image 'Dhoni.bmp'. The SNR value of the secret image before its get embedded is 13.9243. Once it gets embedded with the source image file then the SNR value of the embedded image is 12.7900. After extraction of the secret image the SNR value of the extracted secret image is 13.9243. The result shows clearly that the secret image has been recovered without any degradation in the SNR value. The result of this is shown in figure 3.9.

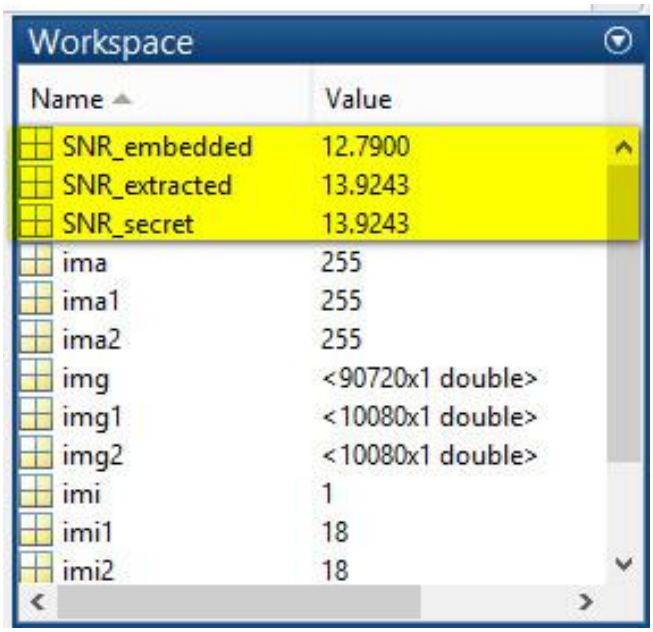

Figure 3.9: SNR of secret image, embedded image and an extracted image using PMDHT.

4. Conclusion

The presented work is powerful for secure communication. From the analysis of the results it is clear that the image quality (like brightness, sharpness) distortion is negligible. As all are embedded within the source image/video like size of secret image/video, content of it and MD5 key within the source image/video hence no other information is needed for decoding at receiver end. Finally, it can be easily extended at the recipient. Results are analyzed in terms of SNR of source image, an embedded image and an extracted image using PMDHT.

References

[1] N. Ghoshal, J. K. Mandal, A. Sarkar, and D. Chakraborty, "*Masking based data hiding and image authentication technique (MHDIAT),*" Advanced Computing and Communications, ADCOM 2008, pp. 119-122, 2008.

[2] Shika and Vidhu Kiran Dutt, "*Steganography: The art of hiding text in image using MATLAB,*" International of Advanced Research in Computer Science and Software Engineering ISSN 2277-128X vol. 4, Issue 9, pp. 822-828, Sept. 2014.

[3] Nameer EL-Emam, "*Hiding a large Amount of data with High Security Using Steganography Algorithm*," in Journal of Computer Science ISSN 1549-3636, vol. 3, no. 4, pp. 223-232, 2007.

[4] M. K. Michcak and P.Moulin, "*A framework for evaluating data hiding capacity of image sources*," IEEE Transaction on image processing, vol. 11, pp. 1029-1042, Urbana, Illinois, Sept. 2002.

[5] S.F. Chang and C. Y. Lin, "*A robust image authentication method surviving JPEG lossy compression*," Proceedings SPIE, vol. 3312, San Jose, pp. 296-307, Jan 1998.

Author Profile

Vivek Jaladi has completed his B.E from Basaveshwar Engineering College, Bagalkot in the year 2008 and completed M.Tech in Digital Electronics & Communication from Dayanand Sagar College of Engineering, Bengaluru in 2011. Presently working as Head of the Department of Electronics & Communication Engineering in Lingaraj Appa Engineering College, Bidar. His area of research includes Image Processing, Signal Processing and Networking.

Avinash completed his bachelor of engineering in Electronics & Communication Engineering from Government Engineering College, Raichur in 2013. Currently pursuing M.Tech in VLSI Design & Embedded Systems in Lingaraj Appa Engineering College, Bidar. He is interested in embedded systems and has capability of writing own programs in assembly language 8051 microcontroller.