Volume 4, Issue 7, July 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Medical Image Steganography with Digital Water Marking

Muralikrishna Nangedda[1]
[#1]Pursuing M.Tech , Department of Computer Science, Vasi Reddy Venkatadri Institute of Technology Nambur, Guntur (DT), Andhra Pradesh, India

A Sudharsan Reddy[2]
[#2]Assoc prof., Head of Department Information Technology, Vasi Reddy Venkatadri Institute of Technology Nambur, Guntur (DT), Andhra Pradesh, India

Abstract— This paper presents securing the transmission of medical images. The presented algorithms will be applied to images. This work presents a new method that combines image data hiding encryption and Steganography technique for denoised and secure image transmission purpose. In this method we embed the original image with patient information by using lossless LSB data hiding method then apply two shares encryption algorithm for encryption of embedded image so that both image and patient information is completely encrypted after that for more security and cover completely to embedded encrypted image .We apply steganography by medical image of any other as cover image and embedded encrypted image as secrete image with the Secret key. In receiver side when the message is arrived then we apply the inverse methods in reverse order to get the denoised original image and patient information. We have applied and showed the results of our method to medical images

Keywords— Data Hiding, Data Embedding, Data Extraction, Decryption, Encryption, Steganography, Data Hiding.

I. INTRODUCTION

The need of fast and secure transmission is vital in the medical world. Nowadays, the transmission of images is a daily routine and it is necessary to find an efficient way to transmit them over the net. In this paper we propose a new technique to cipher an image for secure and denoised transmission. Our research deals with image encryption, data hiding and steganography. There are several methods to encrypt binary or grey level images [1,2,3]. Watermarking can be an answer to merge image with patient information. For applications dealing with images, the watermarking objective is to embed visibly or invisibly message inside the image. [1] To embed the image in the patient information we have used a lossless LSB watermarking technique. A secret sharing scheme shares a secret into a number of shares so that the cooperation of a predetermined group of shareholders reveals the secret whereas the secret reconstruction is impossible for any unauthorized set of shareholders. Visual cryptography is a kind of secret sharing in which the secret reconstruction can be done only by the human visual system .[4] In previous methods owner encrypts the original uncompressed image using an encryption key to produce an encrypted image and then a data hider embeds additional data into the encrypted image using. a data-hiding But there was a problem To decrease the transmission time and recover the image and patient information.

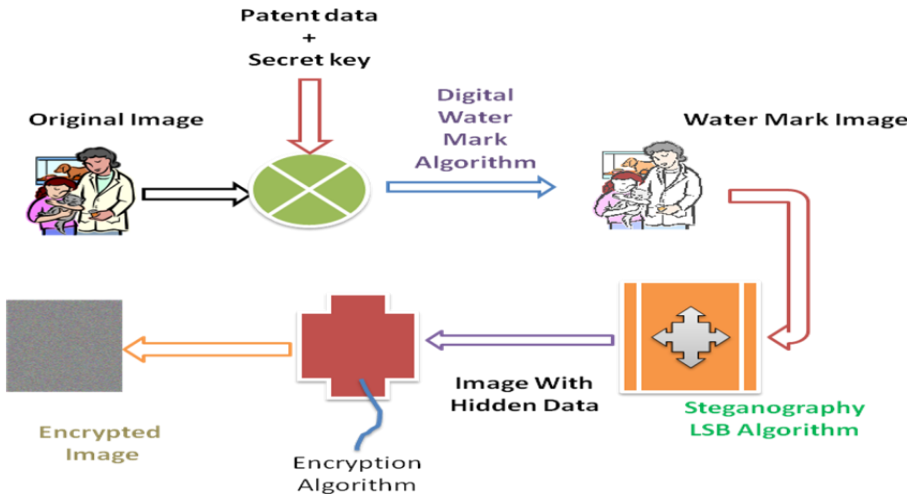


Fig 1: Overall System Architecture

To reduce transmission time the data compression is necessary. Since few years, a new problem is trying to combine in a single step, compression, and encryption and data hiding .So far, few solutions have been proposed to combine image encryption and compression for example. Nowadays, a new challenge consists to embed data in encrypted images. Since the entropy of encrypted image is maximal, the embedding step, considered like noise, it is not possible by using standard

data hiding algorithms. A new idea is to apply reversible lossless data hiding algorithms on image before encryption is done. there was another problem that the image and patient information is not completely hidden so with the help of image analysis intruder can decrypt the image or if either of data hiding key or encryption key is leaked then the intruder can extract or decrypt the message and can see the patient information through data hiding key or decrypt the message through encryption key. To resolve this problem we use steganography by using other medical image and if any hacker get the image then He will assume that the other medical image is real one. [3][5] So in order to achieve this idea we have divided into three modules i.e. Digital Water Marking , Data Hiding (LSB algorithm Steganography), Encryption / Decryption

II. DIGITAL WATER MARKING

The mark is a Gaussian sequence of pseudo-random real numbers and will be denoted X = x1, x2,….xn were n is length of watermark. The choice of the watermark length n determines to which degree the watermark is spread out. In most cases the larger the watermark the lesser the embedding strength needs to be. There is no one watermark length n that is suitable for all images therefore it is image specific.

Embedding Watermark

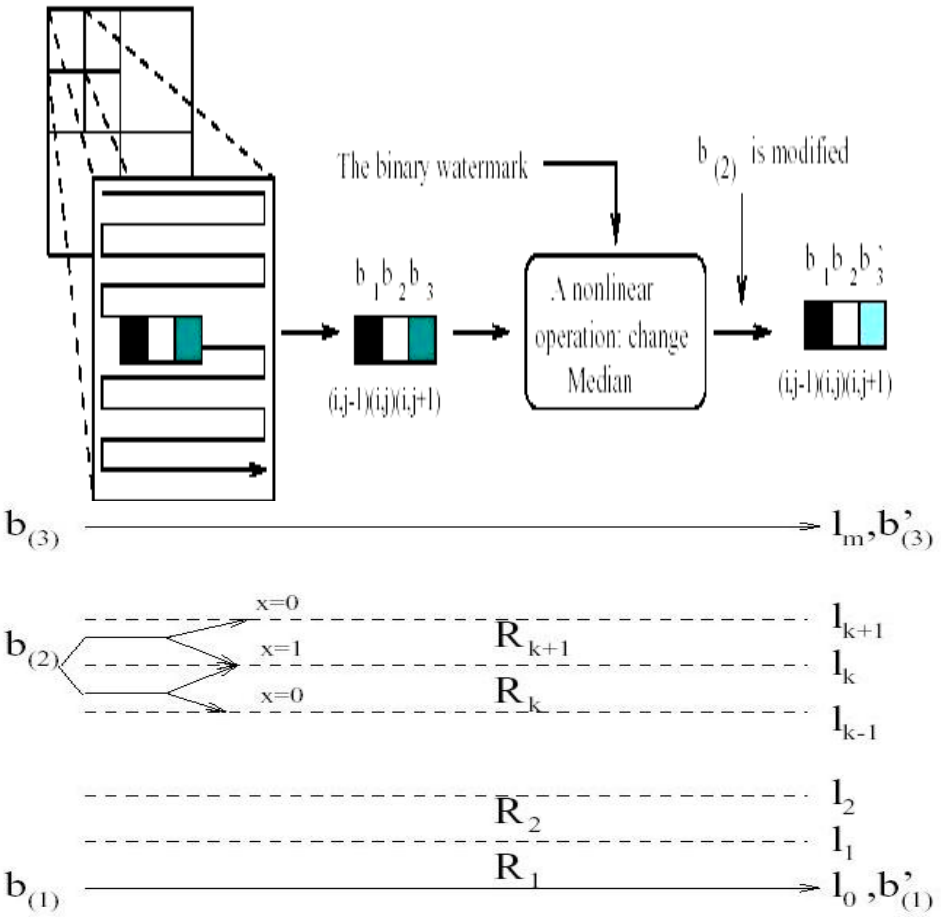
When embedding the watermark a series of coefficients are extracted from the images DCT domain. Let the image be denoted by P and the series of coefficients be C = c1, c2,….cn were n is length of watermark. The series of coefficients C are the largest (most significant) in the DCT domain. Before watermarking the image P it has to be subjected to a DCT transformational change. Extraction of significant coefficients can then take place. The watermark can then be embedded by altering the coefficients by the following formula:

(1) c_i' = c_i + a x_i

Where a can vary from 0 to 1 and is the embedding strength of the watermark. As noted in Cox's paper it may not be applicable to use a single scaling factor for all values of ci. It is possible to change the above formula to:

(2) c_i' = c_i + a_i x_i

Rules can then be set so that a_i is chosen so that watermark will not be perceptually noticeable. The first of the equations is implemented for Cox's algorithm. Because we are using a single scaling factor it is best to leave this at its default value of 0.5 for most pictures so that watermark will most likely not be noticeable. Figure 1 below illustrates the embedding algorithm just described.



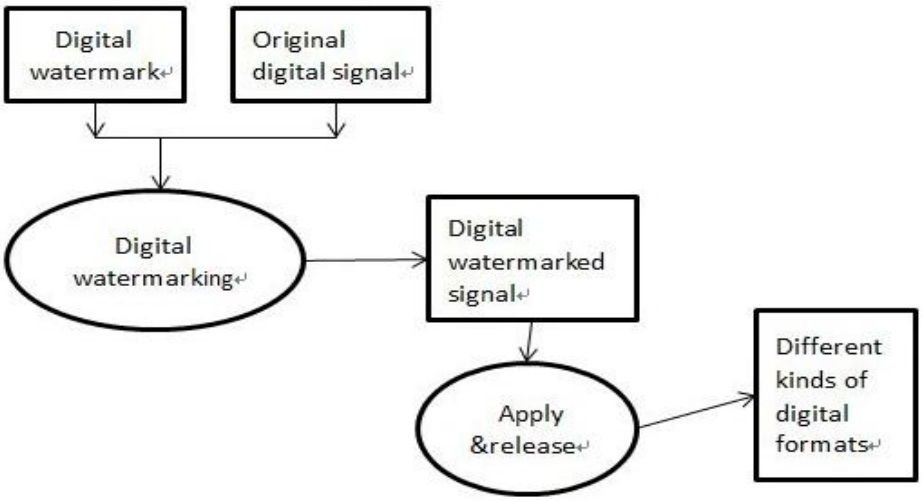


Fig2. workflow of digital watermarking application

Algorithm for Watermark Generation

Step 1 : Read the source image file into a BufferedImage object using the ImageIO.read() method.
Step 2 : Obtain graphics context of the BufferedImage object.
Step 3 : Use the Graphics2D object to paint the watermark which can be a String, an image or whatever can be drawn with the Graphics2D's API. But basically, the watermark is usually translucent so an alpha channel is needed.
Step 4 : Write the output image using the ImageIO.write() method.

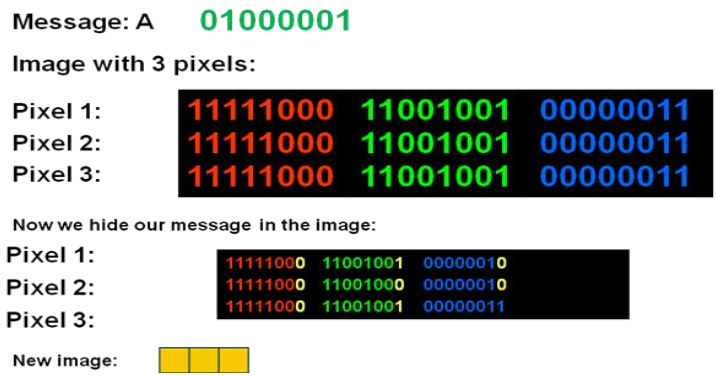
```
BufferedImage sourceImage = ImageIO.read(sourceImageFile);
Graphics2D g2d = (Graphics2D) sourceImage.getGraphics();

// initializes necessary graphic properties
AlphaComposite alphaChannel = AlphaComposite.getInstance(
AlphaComposite.SRC_OVER, 0.3f);
g2d.setComposite(alphaChannel);
g2d.setColor(Color.RED);
g2d.setFont(new Font("Arial", Font.BOLD, 64));
FontMetrics fontMetrics = g2d.getFontMetrics();
Rectangle2D rect = fontMetrics.getStringBounds(text, g2d);

// calculates the coordinate where the String is painted
int centerX = (sourceImage.getWidth() - (int) rect.getWidth()) / 2;
int centerY = sourceImage.getHeight() / 2;

// paints the textual watermark
g2d.drawString(text, centerX, centerY);

ImageIO.write(sourceImage, "png", destImageFile);
g2d.dispose();

System.out.println("The tex watermark is added to the image.");
```

Code 1: Digital Watermarking

III. DATA HIDING IN IMAGE (STEGANOGRAPHY)

A digital image is represented by a one-dimensional array of numbers that represent the different light intensities of each pixel. The dimensions of a 640 x 480 pixel image can literally be multiplied out to find the total amount of pixels in the image, in this case 307,200 pixels. In digital photography this is known as the resolution of an image; a digital camera that takes pictures of 640 x 480 is known as a 0.3 Mega Pixel resolution camera for this reason. Digital images usually use either 24-bits (standard bitmap) or 8-bits (standard GIF image, colour or grey scale) for the storage of intensity information per pixel. This means that in a bitmap image (BMP) there are a potential 16.8 Million colours per pixel and in a GIF, 256 different colour combinations. In a typical 24-bit image, each pixel has three colour components, red, green and blue, each component using 8-bits to represent a value from 0 to 255. An 8-bit image on the other hand can either have a colour palette of 256 different grey levels or colour values. Naturally this results in the 8-bit colour image having to perform a "best fit" in order to match a real world colour to its limited palette. The number of pixels in an uncompressed BMP image contributes directly to file size, for instance, a 640 x 480 image has 307,200 pixels in total, and each of these is represented by 24-bits which equal a total of 900 Kilobytes.

STEPS OF LSB INSERTION ALGORITHM



Step 1:

- The carrier image called the **cover object** is converted to array of bits
- This uses the java classes java.awt.image.BufferedImage, javax.imageio.ImageIO,
- java.awt.Graphics2D,java.awt.image. WritableRaster and java.awt.image.DataBufferByte.
- **BufferedImage:** A bufferedImage is something to be comfortable with when dealing with images. They are easily used with the newly introduced ImageIO class of Java 1.5.0 as well as containing methods for accessing the raster and buffer of the image, which makes image editing much easier.
- **ImageIO:** A useful class to handle IO operations on images. This class has much to offer, but as far as this program is concerned, the read() and write() methods will be sufficient.
- **Graphics2D:** A class which has been around for a long time as far as Java is concerned, and allows access to some of the more in depth aspects of graphics/images. Allows for creating editable areas in a new image or an image which already exists. As well as allowing a way to reach the renderable area of the image. This class also allows for an easy switch from image space to user space, which is necessary when modifying or reading certain bytes of an image.
- **WritableRaster:** This by definition is the process of rendering an image pixel by pixel, which comes in handy when you need to access the bytes of an image, that are representing pixels. Writable Raster is a sub-class of Raster itself, which has methods to access the buffer of an image more directly.
- **DataBufferByte:** The form of a byte array buffer for an image.

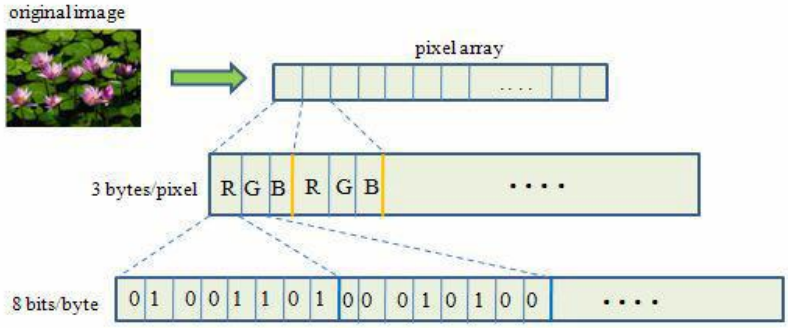


Fig3: Accessing Of Bits

Step 2:

- The secret message text file called the stego object is read and its characters/ bytes are converted to ascii values and then to array of bits.
- For reading the file it uses StringBuffer, BufferedReader and FileReader classes.
- After reading the file, the file is stored in a String.
- The String is converted array of bits by converting all characters to ascii value and doing some bit operations.

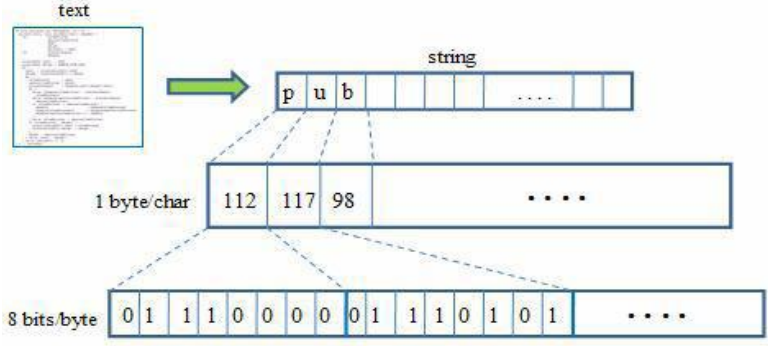


Fig4: for accessing the bits of a message

Step 3:

Now the text bytes are embedded in the carrier image bytes.

Here is a carrier image byte:

0	1	1	0	1	0	1	0

The decimal value of this byte is 106

When we change one LSB from 0 to 1

0	1	1	0	1	0	1	1

The decimal value changes to 107. This change will affect a little bit in some of the colour of a pixel which cannot be marked with human eye.

- But if we change some bits other than LSB, there will be a significant change in value and can be marked with human eye. That's why we are replacing the LSB of the image byte with the secret data bytes.
- One bit of the secret data bit is inserted to the LSB of the image byte. So one byte of the secret data requires 8 bytes of the image.

  The length of the text in binary form is calculated beforehand, and hidden in the image before the text. In other words, the steganographic information (the *stego*) has two parts: the size of the binary message, followed by the message itself.

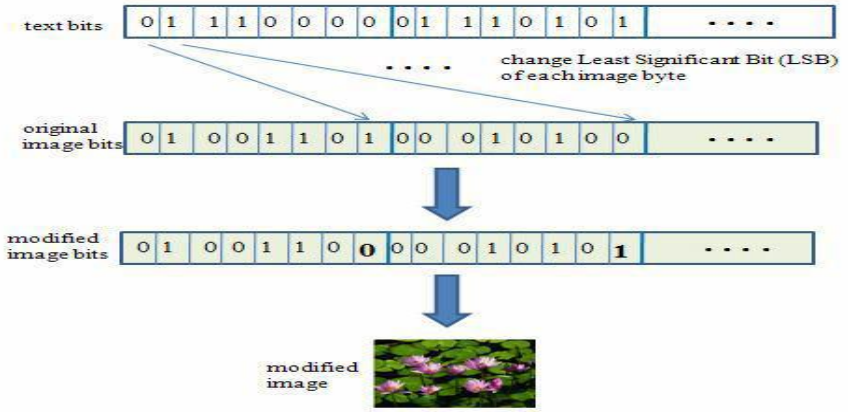


Fig5: Inserting Text In To Image

First 32 bytes of the image consists of the size of the secret data. Because size of data is an integer and integer takes 4 bytes or 32 bits in java. So to accommodate 32 bits of the size of data(integer), it require 32 bytes as each bit will be inserted to LSB first 32 bytes of data.

Size of message	Message

32 bytes

Step 4:

- The message de-embedded/ extracted from the image.
- Extracting the text from the modified image involves copying the LSB of the modified image's bytes and recombining them into bytes in a text file as shown in the figure.
- After that hidden bytes are constructed by shift left operation and inserting hidden bytes.
- All those hidden bytes are collected and finally written to a new text file and saved.

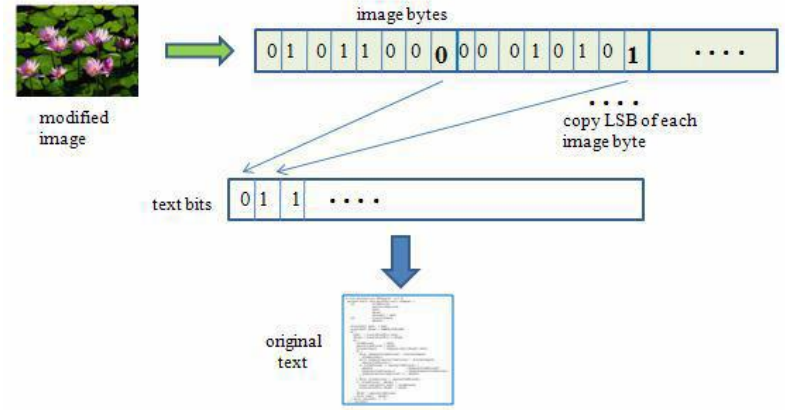


Fig6: Extracting text from modified image

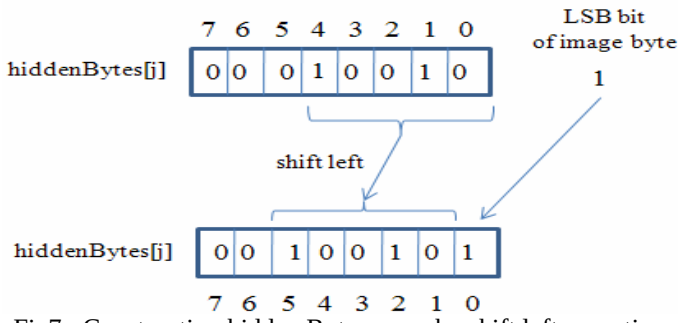


Fig7 : Constructing hidden Bytes array by shift left operation

ADVANTAGES OF LSB ALGORITHM

The advantages of LSB are its simplicity to embed the bits of the message directly into the LSB plane of *cover-image* and many techniques use these methods. Modulating the LSB does not result in a human-perceptible difference because the amplitude of the change is small. Therefore, to the human eye, the resulting *stego-image* will look identical to the *cover-image*. This allows high perceptual transparency of LSB.

The advantages of LSB techniques are:

- Popularity
- Easy to understand and comprehend
- High perceptual transparency.
  Low degradation in the image quality

IV. ENCRYPTION /DECRYPTION

In This Module Image Is Converted In To Another Form To Prevent The Unauthorized View Of The Image Data I am explaining how to encrypt image in a simple way. This mechanism uses very simple logic, i.e., it encrypts the pixels than encrypting the whole image file. So first I am giving the theoretical explanation. Basically an image contains lot of pixels, so as i told earlier our encryption mechanism encrypts only the pixels. For encryption and decryption it uses the same key. Based on the key, everypixel value of the image will be XORed and that XORed value will be replaced in the encrypted image. The Decryption and encryption process is the same

CODE FOR ENCRYPT DECRYPT

Pseudo Random Numbers are generated using SHA1PRNG algorithm. Secure Random is used to generate random numbers.

```
SecureRandom sr= SecureRandom.getInstance("SHA1PRNG");
sr.setSeed(sb.toString().getBytes());

BufferedImage FSImg=ImageIO.read(new File(sPath));
for (int w = 0; w < FSImg.getWidth(); w++) {
for (int h = 0; h < FSImg.getHeight(); h++) {
Color color=new Color(FSImg.getRGB(w, h));
Color newColor=new

Color(color.getRed()^sr.nextInt(255),
color.getGreen()^sr.nextInt(255), color.getBlue()^sr.nextInt(255));
FSImg.setRGB(w, h, newColor.getRGB());
}
}
System.out.println("Process Completed!!..");
```

The password is converted to ASCII code that value is given as seed for the random number generation. Every pixel in the image is retrieved, and the RGB values are XORed with the random value. The new image is created using the newly obtained pixel values.

V. CONCLUSION

Steganography is a really interesting subject and outside of the mainstream cryptography and system administration that most of us deal with day after day. “You never know if a message is hidden”, this is the dilemma that empowers steganography. As more emphasis is placed on the areas of copyright protection, privacy protection, and surveillance, we believe that steganography will continue to grow in importance as a protection mechanism.

REFERENCES

[1]. A Joint Encryption/Watermarking System for Verifying the Reliability of Medical Images Dalel Bouslimi, *Member, IEEE*, Gouenou Coatrieux, *Member, IEEE*, Michel Cozic,and Christian Roux, *Fellow, IEEE*

[2] . Digital Watermarking Technique for protecting Digital images" by Munesh Chandra,shikha pande, rama chaudhri,computer science and information technology (iccsit),2010 3rd IEEE international conference

[3]. R Anderson, C. Manifvas, "Chameleon- A new kinf of stream cipher," FSE '97, vol.1267,pp.107-113,1997.

[4]. K. Chen, T.V Ramabadran, "Near lossless compression of Medical Images through Entropy Coded DPCM" IEEE Trans. On Medical Imaging, vol.13, pp. 538-548, 1994.

[5]. U.Ragendra Acharya , U. C Niranjan, S. S Iyangar "Simultaneous storage of patient information with medical images in frequency domain" Computer Methods Programs biomed., vol. 76, pp.13-19, 2004.

[6] W. Puech" Image Encryption and Compression for Medical Image Security" PROCEEDING OF IEEE Image Processing Theory, Tools & Applications.

[7] Ming YANG,Lei SONG,Monica TRIFAS,Dorothy BUENOS-AIRES,Lei CHEN, Jaleesa ELSTON," Secure Patient Information and Privacy in Medical Imaging IEEE "

[9] Xinpeng Zhang]IEEE SIGNAL PROCESSING LETTERS, VOL. 18, NO. 4, APRIL 2011 255 Reversible Data Hiding in Encrypted Image

[10] Zhicheng Ni et al. "Reversible Data Hiding", IEEE Transactions on Circuits and Systems for Video Technology, Vol.16, No.3, March 2006.

[11] U. Rajendra Acharya, U. C. Niranjan, S. S. Iyengar, N. Kannathal, andL. C. Min, "Simultaneous storage of patient information with medicalimages in the frequency domain," Comput. Methods Programs Biomed.,vol. 76, pp. 13– 19, 2004.

[12] U. Rajendra Acharya, D. Acharya, P. Subbanna Bhat, and U. C. Niranjan,"Compact storage of medical images with patient information," IEEETrans. Inf. Technol. Biomed., vol. 5, no. 4, pp. 320–323, Dec. 2001.

AUTHORS

MURALIKRISHNA NANGEDDA ,Pursuing M.Tech , Department of Computer Science , VasiReddy Venkatadri Institute of Technology Nambur, Guntur (DT), Andhrapradesh , India.

A SUDARSAN REDDY Assoc prof., Head of Department Information Technology, VasiReddy Venkatadri Institute of Technology Nambur, Guntur (DT), Andhrapradesh , India

