ISSN 2319-2720

Volume 1, No.2, September – October 2012
International Journal of Computing, Communications and Networking
Available Online at http://warse.org/pdfs/ijccn07122012.pdf

Novel Steganography Technique for Information Hiding

Dilip Vishwakarma[1], Dept of Computer Application, S.A.T.I. Vidisha, (M.P), India, dilipvishwakarma.85@gmail.com
Satyam Maheshwari[2], Dept of Computer Application, S.A.T.I. Vidisha, (M.P), India, satyam.vds@gmail.com
Deepak Chopra[3] Dept of Computer Application, S.A.T.I. Vidisha, (M.P), India

ABSTRACT

Steganography is the art of hiding information in ways that prevent the detection of hidden messages. It includes a vast array of secret communications methods that conceal the message's very existence. These methods include invisible inks, microdots, character arrangement, digital signatures, covert channels, and spread spectrum communications. Steganography and cryptography are cousins in the spy-craft family. This paper information hiding and its applications, and image compression using proposed efficient encoding technique with the main focus being on hiding in the spatial domain are developed. Three information hiding methods are proposed, which are based on the encoding technique, are tested and the results are analyzed. Increase in tolerance level would allow using all range blocks so that more data can be stored. However low tolerance is desirable in order to give an image that is visually close to the original. Further research should go towards improving the watermarking program and adding extra functionality. One of these is looking at having multiple watermarks for a single image, so that different parts of the image have a different watermark. There is also the need to further develop the robustness of existing watermarking techniques to combat the ever-increasing attacks on watermarks. We used fixed partitioning scheme. Instead of this, adaptive partitioning scheme can be used, which would yield better results if used as the basis for the data hiding method.

Keywords: Steganography, LSB, Hiding Encoding Technique

1. INTRODUCTION

Steganography hides the message so it cannot be seen. A message in cipher text, for instance, might arouse suspicion on the part of the recipient while an "invisible" message created with steganographic methods will not. The word 'steganography' comes from the Greek "steganos" (covered or secret) and "graphy" (writing or drawing) and thus means, literally, covered writing. It is about exploiting the limited powers of the human visual system. Within reason, any plain text, cipher-text, other images, or anything that can be embedded in a bit stream can be hidden in an image. It does not have to be robust, data should be just invisible. In contrast to cryptography, where the "enemy" is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem, the goal of steganography is to hide messages inside other "harmless" messages in a way that does not allow any "enemy" to even detect that there is a second secret message present. Steganography is in the (especially military) literature also referred to as transmission security.
Steganographic technique finds its main application in the field of secret communication. It can be used by intelligence agencies across the world to exchange highly confidential data in a covert manner e.g. a secret agent can hide a map of a terrorist camp in a photograph using image steganographic software and post it on a public discussion board or forum. An officer from the head office could download the photograph from the forum and easily recover the hidden map. Steganographic techniques can also prevent a legitimate entity against coercion e.g. if trade secrets are encrypted and stored on hard disks

they can be easily visible and a malicious user may coerce the legitimate user to disclose the same [1-5]. Digital representation of signals brings many advantages when compared to analog representations, such as lossless recording and copying, convenient distribution over networks, easy editing and modification, and durable, cheaper, easily reachable archival. Unfortunately, these advantages also present serious problems including wide spread copyright violation, illegal copying and distribution, problematic authentication, and easy forging. Piracy of digital photographs is already a common phenomenon on the Internet. Today, digital photographs or videos cannot be used in the chain of custody as evidence in the court because of nonexistence of a reliable mechanism for authenticating digital images or tamper detection. Information hiding in digital documents provides a means for overcoming those problems.

The aim is to develop a new fractal encoding technique, which can find out the possibility to hide maximum amount of data in an image without degrading its quality. Second issue is to make the hidden data robust enough to withstand image processing which do not change the appearance of image. So this technique can also be used a sophisticated algorithm provided security. And also, this technique should be computationally less intensive.

2.BACKGROUND TECHNIQUES

Steganography is derived from the Greek for covered writing and essentially means “to hide in plain sight”. Steganography is the art and science of communicating in such a way that the presence of a message cannot be detected. Simple steganographic techniques have been in use for hundreds of years, but with the increasing use of files in an electronic format new techniques for information hiding have become possible.

Figure 1 shows how information hiding can be broken down into different areas. Steganography can be used to hide a message intended for later retrieval by a specific individual or group. In this case the aim is to prevent the message being detected by any other party. The other major area of steganography is copyright marking, where the message to be inserted is used to assert copyright over a document.

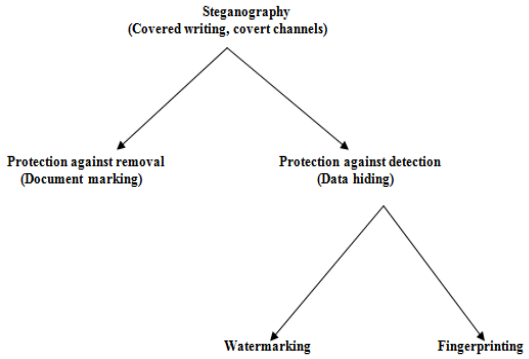


Figure 1: Types of steganography

Steganography and encryption are both used to ensure data confidentiality. However the main difference between them is that with encryption anybody can see that both parties are communicating in secret. Steganography hides the existence of a secret message and in the best case nobody can see that both parties are communicating in secret. This makes steganography suitable for some a task for which encryption isn’t, such as copyright marking. Adding encrypted copyright information to a file could be easy to remove but embedding it within the contents of the file itself can prevent it being easily identified and removed.

The objective of steganography is to hide a secret message within a cover-media in such a way that others cannot discern the presence of the hidden message. Technically in simple words “steganography means hiding one piece of data within another”. Modern steganography uses the opportunity of hiding information into digital multimedia files and also at the network packet level. Hiding information into a media requires following elements:

➔The cover media(*C*) that will hold the hidden data
➔The secret message (*M*), may be plain text, cipher text or any type of data
➔The stego function (*Fe)* and its inverse (Fe^{-1})
➔An optional stego-key (*K*) or password may be used to hide and unhide the message.

The stego function operates over cover media and the message (to be hidden) along with a stego-Key (optionally) to produce a stego media (*S*). The

schematic of steganographic operation is shown below (Figure 2).

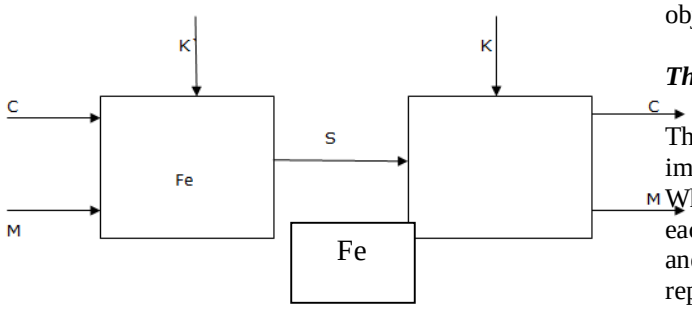


Figure 2: The Steganographic operation

Steganography and Cryptography are great partners in spite of functional difference. It is common practice to use cryptography with steganography [3, 6].

2.1 Requirements of Hiding Information Digitally

There are many different protocols and embedding techniques that enable us to hide data in a given object. However, all of the protocols and techniques must satisfy a number of requirements so that steganography can be applied correctly. The following is a list of main requirements that steganography techniques must satisfy:

➔The integrity of the hidden information after it has been embedded inside the stego object must be correct. The secret message must not change in any way, such as additional information being added, loss of information or changes to the secret information after it has been hidden. If secret information is changed during steganography, it would defeat the whole point of the process.

➔The stego object must remain unchanged or almost unchanged to the naked eye. If the stego object changes significantly and can be noticed, a third party may see that information is being hidden and therefore could attempt to extract or to destroy it.

➔In watermarking, changes in the stego object must have no effect on the watermark. Imagine if you had an illegal copy of an image that you would like to manipulate in various ways. These manipulations can be simple processes such as resizing, trimming or rotating the image. The watermark inside the image must survive these manipulations, otherwise the attackers can very easily remove the watermark and the point of steganography will be broken.

➔Finally, we always assume that the attacker knows that there is hidden information inside the stego object [1, 7].

The LSB Technique:

The least significant bit i.e. the eighth bit inside an image is changed to a bit of the secret message. When using a 24- bit image, one can store 3 bits in each pixel [illegible] a bit of each of the red, green and blue [illegible]onents, since they are each represente[illegible] An 800×600 pixel image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data. As an example, suppose that we have three adjacent pixels (9 bytes) with the RGB encoding.

10010101 00001101 11001001
10010110 00001111 11001011
10011111 00010000 11001011

When the number 300, can be which binary representation is 100101100 embedded into the least significant bits of this part of the image. If we overlay these 9 bits over the LSB of the 9 bytes above, we get the following (where bits in **bold** have been changed)

10010101 0000110**0** 1100100**0**
1001011**1** 0000111**0** 11001011
10011111 00010000 1100101**0**

Here the number 300 was embedded into the grid, only the 5 bits needed to be changed according to the embedded message. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size. Since there are 256 possible intensities of each primary color, changing the LSB of a pixel results in small changes in the intensity of the colors. The human eye cannot perceive these changes - thus the message is successfully hidden. With a well-chosen image, one can even hide the message in the LSB without noticing the difference.

2.2 Modern Techniques of Steganography

The common modern technique of steganography exploits the property of the media itself to convey a message. The following media are the candidate for digitally embedding message [3, 5]: -

- ***Plaintext***
- ***Still imagery***
- ***Audio and Video***

- ***IP datagram.***
- **Binary File Techniques**
- **Image Techniques**
- **Sound Techniques**

There is a general attack on mark readers which explores an image on the boundary between no mark having been found and one being detected. An acceptable copy of the image can be iteratively generated which does not include the mark. Clearly the software used to implement steganographic techniques needs to be secure and ideas from other areas of computer security can be used to ensure this. Steganographic attacks consist of detecting, extracting and destroying hidden object of the stego media. Steganography attack is followed by steganalysis. There are several types of attacks based on the information available for analysis. Some of them are as follows: -

- ➔**Known carrier attack:** The original cover media and stego media both are available for analysis.
- ➔**Steganography only attack:** In this type of attacks, only stego media is available for analysis.
- ➔ **Known message attack:** The hidden message is known in this case.
- ➔ **Known steganography attack:** The cover media, stego media as well as the steganography tool or algorithm, are known.

3. RELATED WORKS

The current Steganography tools based on the LSB algorithms include S-Tools, Hide and Seek, Hide4PGP and Secure Engine Professional. These tools support BMP, GIF, PNG images and WAV audio files as the carriers. Each of these tools has unique features. S-Tools reduce the number of colors in the image to only 32 colors. Hide and Seek makes all the palette entries divisible by four. In addition, it forces the images sizes to be 320x200, 320x400, 320x480, 640x400 or 1024x768 pixels.Hide4PGP embeds the message in every LSB of an 8-bit BMP images, and in every fourth LSB of a 24- bit BMP image. These applications are flawed because they do not analyze the image file after it has been embedded with data to see how vulnerable it is to steganalysis. The transform domain based steganography tools embed the message in the transform coefficients of the image. The main transform domain algorithm is described. These applications can only work with JPGs because most other image formats do not perform transforms on their data. The document based steganography tools embed the secret message in document files by adding tabs or spaces to .txt or .doc files [3] and [8].

These applications are limited because they only work with document files. They also cannot hide much data because there are a very limited number of tabs or spaces they can reasonably be added to a document. In addition, they are vulnerable to steganalysis because it is easy for an attacker to notice a document file that has been embedded with additional tabs or spaces.

The file structure based steganography tools embed the secret message in the redundant bits of a cover file such as the reserved bits in the file header or the marker segments in the file format. These applications cannot hide very large data files because there are a very limited number of header or marker segments available for embedding hidden data [5] and [6].

"An introduction to steganography methods" written by Masoud Nosrati et al. World Applied Programming, Vol (1), No (3), August 2011. 191-195:

Steganography and cryptography are closely related. Cryptography scrambles messages so they cannot be understood. Steganography on the other hand, will hide the message so there is no knowledge of the existence of the message in the first place. In some situations, sending an encrypted message will arouse suspicion while an"invisible" message will not do so. Both sciences can be combined to produce better protection of the message. In this case, when the steganography fails and the message can be detected, it is still of no use as it is encrypted using cryptography techniques. There exist two types of materials in steganography: message and carrier. Message is the secret data that should be hidden and carrier is the material that takes the message in it. In this paper introduce different types of steganography considering the cover data. As the first step about text steganography and investigate its details. Then, image steganography and its techniques will be investigated. Some techniques including Least Significant Bits, Masking and filtering and Transformations will be

subjected during image steganography. Finally, audio steganography which contains LSB Coding, Phase Coding, Spread Spectrum and Echo Hiding techniques will be described [1, 4].

"A Steganography Method Based on Hiding secrete data in MPEG/Audio Layer III" written by Mohammed Salem Atoum et al. IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.5, May 2011, 184-188:

Steganography is a method for hidden information in such a way that can only be detected by its intended recipient. Steganography in audio becomes a challenging discipline, since the Human Auditory System (HAS) is highly sensitive. One of the main obstacles of the data hiding in audio is to develop a system which has the quality to include a big amount of data and without affectively the quality of sound. This paper proposes a novel information-hiding method to hide more information into audios media file (MP3). The bits of information will be hidden between frames (BF) in MP3 file. A In the experimental results, they hide more characters into audios and extract them correctly. The audios with secret information are indiscernible to human ears.

To overcome the shortcomings of the proposed method, suggested :Before All Frame (BAF). The merits of Before All Frames (BAF): extracting the text is faster than other methods for limited size, because the whole text lies in the beginning. The song can be played on all multi-media audio players. High audio quality. A relatively good text file size. It can be played whether the MP3 file encodes VBR or CBR . Taking in consideration that the major issue is the numberof frames in the cover file not the file size (as number offrames will not actually depends on audio file size only but also on frame size) [2, 8].

"A Spatial Domain Image Steganography Technique Based on Matrix Embedding and Huffman Encoding" written by P.Nithyanandam, T.Ravichandran, N.M.Santron & E.Priyadarshini et al. International Journal of Computer Science and Security (IJCSS), Volume (5) : Issue (5) : 2011, 456-468:

This paper presents an algorithm in spatial domain which gives less distortion to the cover image during embedding process. Minimizing embedding impact and maximizing embedding capacity are the key factors of any steganography algorithm. Peak Signal to Noise Ratio (PSNR) is the familiar metric used in discriminating the distorted image (stego image) and cover image. Here matrix embedding technique is chosen to embed the secret image which is initially Huffman encoded. The Huffman encoded image is overlaid on the selected bits of all the channels of pixels of cover image through matrix embedding. As a result, the stego image is constructed with very less distortion when compared to the cover image ends up with higher PSNR value. A secret image which cannot be embedded in a normal LSB embedding technique can be overlaid in this proposed technique since the secret image is Huffman encoded. Experimental results for standard cover images, which obtained higher PSNR value during the operation is shown in this paper.

The proposed technique is not robust against any geometrical distortion such as rotation, translation, scaling, cropping etc., induced on the stego image. Improving this parameter is still under research and not matured yet. The proposed algorithm should be customized to support embedding in the frequency domain. It should be enhanced to withstand geometrical distortion induced on the image [9-10].

We can find out the possibility to hide maximum amount of data in an image without degrading its quality. Second issue is to make the hidden data robust enough to withstand image processing which do not change the appearance of image. So this technique can also be used a sophisticated algorithm provided security.

4.PROPOSED TECHNIQUES

This method takes an image file and steganographic data and produces a new image file that contains the steganographic data. The output image is called steganographic image file and it is similar to the input image file. The encoding technique identifies parts of the image that are most suited for data hiding.

4.1 Retrieving Steganographic Data

Retrieving steganographic method is reverse of hiding. Range and domain regions remain same as in hiding process. To find a bit of steganographic data in the image file, we select a range block from the range

region in the same order as in the hiding process and find corresponding domain block that matches it by a linear relationship. The linear relationship is that each pixel in range block is formed by multiplying corresponding pixel in the domain block by a scale factor and adding an offset factor. We cannot use fractal image compression techniques for the match as relationship between the two blocks is linear. If we use fractal image compression techniques, they would also yield some blocks that are similar but have not been used in data hiding. So, for matching we compute scaling and offset factors from any two pixels in the range and domain blocks, then they are applied on rest of the pixels in the domain block and determined if the corresponding pixel exists in the range block. The match is successful if we can find this linear relationship between the two blocks. If match occurs in D0, the first half of domain region, we set current bit of steganographic data to 0, otherwise if match occurs in D1, we set the bit to 1. Finally, this sequence of bits is XOR with key to get the original data back.

Pseudo-code for hiding steganographic data

• Append data with the label END OF DATA.
• XOR data with key.
• Subsample domain blocks D so that they have same number of pixels as range blocks.
• Classify all domain blocks.
• While there are bits to be stored {
• Take a range block Ri from R.
• Classify Ri.
• If current bit bi is 0
• Consider domain blocks from D0 that are of same class as of Ri.
• else
• Consider domain blocks from D1 that are of same class as of Ri.
• Calculate scale si, offset oi and rms distance drms between Ri and all Djs in the selected quadrant.
• Select domain block Dmin with least drms and corresponding si and oi.
• Multiply each pixel of Dmin by si and add oi and overwrite the corresponding range pixel by the result.
} //

Pseudo-code for retrieving steganographic data

//
• While not END OF DATA {
• Take a range block Ri from R.
• Calculate scale si, offset oi and rms distance drms between Ri and all Djs in D.
• Select domain block Dmin with least drms and corresponding si and oi.
• If Dj is in D0
• Data bit bi is 0.
• else
• Data bit bi is 1.
}
• XOR this bit stream with key to get original data.
//

→It is assumed that the input image is of equal length and breadth.
→Image is grey scale and in the sun-raster format.
→ Size of image header is 800 bytes.
→Each byte in the image data represents a pixel whose level of grey is from 0 to 255.

The hiding algorithm takes image I, steganographic data and a key as input. It outputs a visually identical image I knew that has steganographic data hidden in it. The retrieving algorithm takes image I and key as input and outputs steganographic data. Image I is partitioned into four quadrants. First and third quadrants constitute the range region R and second and fourth quadrants domain region D. Range region is divided into square blocks {r0, r1, . . . , rl} of equal size (4×4 or 8×8). These blocks are non-overlapping. A domain library is built from a set of blocks in the domain region. The length of both sides of domain block is twice that of range blocks therefore there are 4 times as many pixels in a domain block than there are in a range block. Domain blocks are overlapping. Domain library D is represented by the blocks {D0,D1, . . . ,Dm}, so m+1 is the number of domain blocks in D. Domain library is split into two halves {D0,D1, . . . ,Dm/2} which are from first quadrant of image and {D(m/2)+1,D(m/2)+2, . . . ,Dm} from third quadrant of image. First half of D is D0 and other is D1.

We used fixed block partition. Size of all range blocks and domain blocks is fixed. The search strategy we used is block classification.

Domain and range blocks are compared using rms metric. Given two squares containing n pixel intensities, a1, . . . , an (from Di) and b1, . . . , bn (from Ri), we can seek s and o to minimize the quantity

$$R' = \sum (S.ai + o - bi)^2 \quad \text{……………………..(1)}$$

Where i=1……………n.

This will give us contrast and brightness settings that make the finely transformed ai values have the least squared distance from the bi values. The minimum of R0 occurs when the partial derivatives with respect to s and o are zero.

$$A = \pi r^2 \quad \text{……………………..(2)}$$

Method 1: Input Image Partitioning Process

Input image is partitioned into range and domain regions and sub-blocks as described earlier. Range blocks are selected in a pre-defined order. Then the matching domain block is searched in the I or III quadrant of the image depending upon the bit value which to be stored. Before comparing a domain block Dj with the range block Ri, Dj is contracted by a factor of two on each side by averaging neighboring pixels, followed by the application of one of the eight rotations and reflections making up the isomerties of a square. This increases the size of domain library by eight times, so there are greater chances of getting a good match. For each Ri, we calculate si, oi and rms value corresponding to all Djs in the selected quadrant.

$$s = \frac{[n\sum aibi - \sum ai \sum bi]}{[n\sum a^2 i - (\sum ai)^2]} \quad \text{…………………..…..(3)}$$

$$o = 1/n \; [\sum bi - s\sum ai] \quad \text{……………………..(4)}$$

Where all i = 1……..n, n>=1.

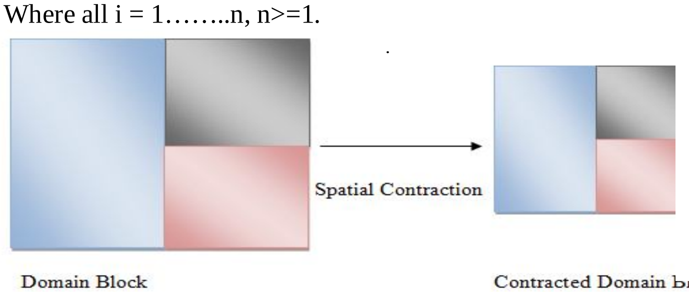


Figure 3: Spatial contraction of a domain block.

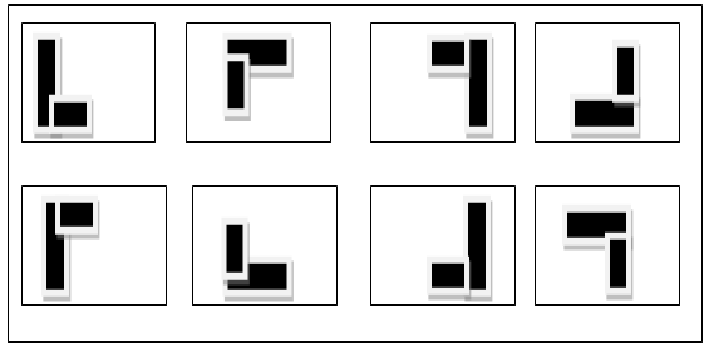


Figure 4: The square isometrics.

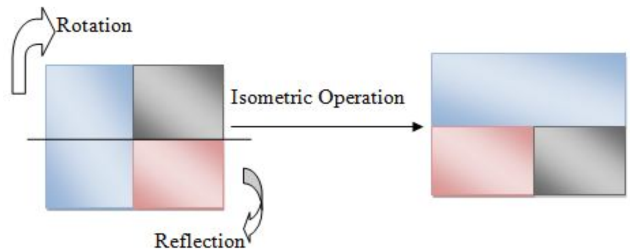


Figure 5: An isometric applied to a domain block.

These parameters a^2 are calculated as given in equations 3 and4 respectively. Domain block corresponding to which rms value is minimum, is chosen alongwith si and oi. Each pixel of the subsampled domain block is multiplied by si and added to oi to generate a new range block Rinew (see figure 4.4) which is written over Ri. So, the bits are stored in form of mappings from domain region to range region. The number of such mappings is equal to the number of bits stored and hence, equal to the number of range blocks modified. While extracting the message, range blocks are selected in the same order as in hiding process.

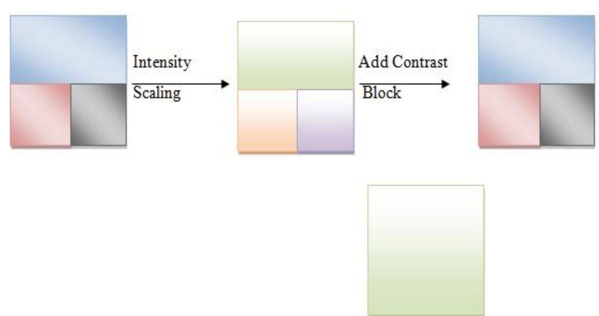


Figure 6: An affine transform applied to a domain block.

Here, we consider all domain blocks for each Ri. We calculate si and oi (from equations 3 and 4) which are applied on Dj and rms value is calculated for Ri. In this way, rms value is calculated between Ri and all Djs in I and III quadrants. If the domain block Dj with minimum rms value is in I quadrant, bit value bi is 0 else if Dj is in III quadrant bit value bi is 1. Figure 7 shows mappings from domain region to range region which store the bit string.

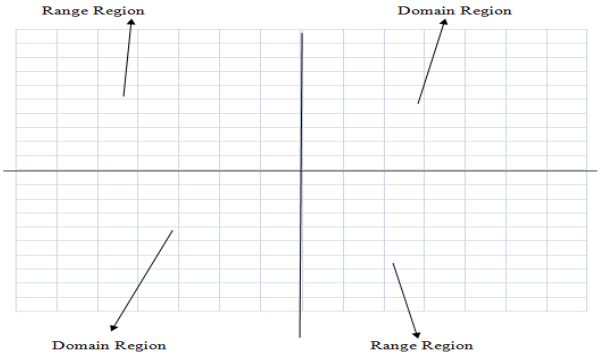


Figure 7: Mappings from domain region to range region

Method 2: Domain Block Selecting Process

Domain block is selected with the minimum difference from Ri and according to the location of domain block bit is decoded. In the retrieving process, for each selected range block Ri we examine all domain blocks. We find b1, bk, a1 and ak such that b1 6= bk. si and oi are calculated.

These parameters are applied on the pixels {ak+1, ak+2, . . . , an} of Dj and compared with the pixels {bk+1, bk+2, . . . , bn} of Ri. Domain block is selected with the minimum difference from Ri and according to the location of domain block bit is decoded. While hiding, if the fractional part of b1new and b knew is not ignored then the value of si and oi will be calculated using their fractional part also. But the value of a pixel cannot have fractional part, so that will be ignored (or rounded-off) while writing over the range block. So, in retrieving process, si and oi are calculated using only the integer values. Therefore, the value of si and oi calculated while hiding will be different from calculated while retrieving. Due to the different value of si and oi, rms value between the range and domain blocks will change and hence the mapping will change. Change in mapping may lead to the retrieval of incorrect information.

Method 3: Least Significant Bit (LSB) Selecting

Here we consider only the least significant bit (LSB) of the pixels. LSB of the pixels of range block Ri are compared with corresponding LSB of the pixels of all domain blocks in the selected quadrant. Domain blocks Dj with the minimum difference is chosen and its least significant bit plane is copied over least significant bit plane of range block Ri. LSB of the pixels of the new range block is same as the LSB of the pixels of the corresponding domain block. So, the mapping is an identity function which is applied on LSB plane of blocks. In this way, a bit is stored. To get the message back, range blocks are selected in the same order as in hiding process. For a range blocks Ri, LSB of its pixels are compared with the LSB of the pixels of all domain blocks. Domain block is selected with the minimum difference from Ri and according to the location of domain block bit is decoded.

5.EXPERIMENTAL RESULTS

5.1 Testing Formulas

The testing of each method was performed on a system with modern Pentium processor and windows operating system. All the images were 256×256, 8-bit greyscale, of Lenna. The signal to noise ratio, equation 5, is a commonly used pixel-based visual distortion metric and this was used to measure the distortion between the original image and the image containing data. A low SNR means that the image has been greatly distorted.

$$SNR = \frac{\sum p^2 x,y}{\sum (Px,y - P' x,y)^2} \quad \text{.....(5)}$$

Where, SNR is the signal to noise ratio, Px,y is a pixel in the original image with coordinates (x,y), and P'x,y is a pixel in the image containing data with coordinates (x,y). The signal to noise ratio is usually measured in decibels and converted using equation 6.

SNR(dB) = 10log10(SNR)……(6)

The percentage accuracy of the methods was calculated by finding the number of correct bits recovered (Dτ) and dividing it by the total number of bits in the actual data (DA).

Accuracy= (Dτ / D A) X 100……(7)

5.2 Results of Method 1

In the first experiment a data of its bits was hidden into an image and results were taken using different range block size. It was observed that if the sizes if the range and domain block is increased signal to noise ratio (SNR) decreases. But for the range block size 2×2, SNR is very low. This is due to the inability to find proper s and o as the range block size is too small. For 4×4 and 8×8 range blocks SNR is good enough. Again for 16×16 range block, SNR is reduced to very low value. It decreases as the data length is increased. Accuracy of getting data back is very less. It is between 40 to 60 percent. This is because while hiding the data, only integer part of the new range block pixel is written over the image. Fractional part is ignored. So, in the retrieving process we don't get back the same value of s and o for a range and domain block pair as calculated while hiding. Hiding time and retrieving time dropped on increasing the block size (see figure). Hiding time increased sharply with the increase in data length, but retrieving time increased slowly.

5.3 Results of Method 2

Secondly, the value of SNR decreased with the increase in the block size. It went down sharply in the beginning, but after range block size 8×8 it decreased slowly. Sudden falls in SNR were observed when the data length was increased with range block size 4×4. But with 8×8 it was nearly constant when data length was more than 28 bits. Some patch was observed on the image after hiding the data. This is because while hiding the data, recalculated parameters s and o are not equivalent to the actual s and o. So, the newly generated range block is not similar to the original range block. For the range block size 2×2, data is not correctly retrieved. But for the larger blocks, it works well. Hiding time and retrieving time decreased gradually with the increase in block size (see figure). But they increased sharply as the data length is increased.

5.4 Results of Method 3

In third experiment, SNR decreased with the increase in block size. As the block size increases, SNR converges to some value for a fixed data. It also decreased, as the length of data was increased. This method did not work for block size 2×2, because there were more than one domain block matching with a range block. So, it was likely to have incorrect matching domain block and so incorrect bit value. But for larger block size, accuracy was good.

Hiding time and retrieving time decreased with the increase in block size. However, retrieving time was nearly same for range block sizes 8×8 and 16×16 (see figure). As the data length was increased, hiding time and retrieving time were increased very slowly. So, for this method, length of data does not affect the time taken for the execution of both hiding and retrieving algorithm. We changed the brightness and contrast of the watermarked image by different amount. It was observed that method 2 is robust against this attack. Watermark was correctly extracted when we used method 2 with range block size 16×16. For range blocks 4×4 and 8×8, watermark was retrieved with the accuracy of more than 92%. Method 3 is not robust to this attack. However, it has shown slightly better a result when brightness and contrast was set to 11 and range block size was 8×8 and 16×16.

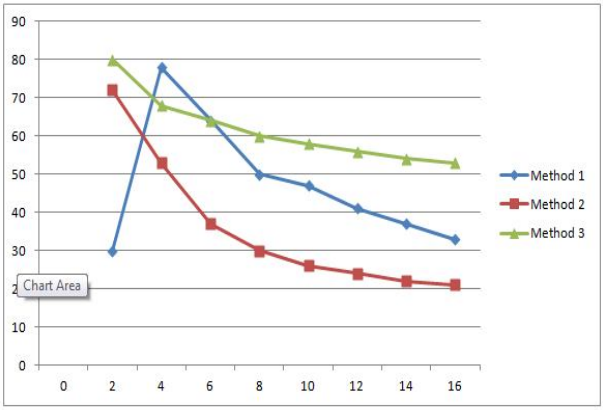
5.5 Attacks on the Watermarked Image

Several attacks were done on the watermark put by the methods 2 and 3 with range blocks 4×4, 8×8 and 16×16. A 256×256 grey-scale Lenna image was watermarked. Length of the watermark was 56 bits. Attacks performed are change of brightness and contrast, addition of noise, JPEG compression and scaling of watermarked image. All these were performed using the "gimp" software.

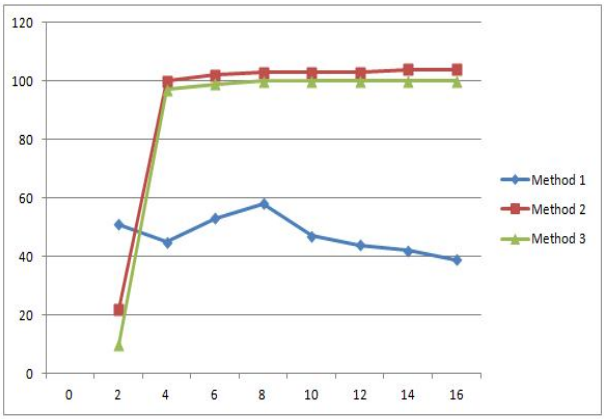
- ***Changing Brightness and Contrast***

We changed the brightness and contrast of the watermarked image by different amount. It was observed that method 2 is robust against this attack. Watermark was correctly extracted when we used method 2 with range block size 16×16. For range

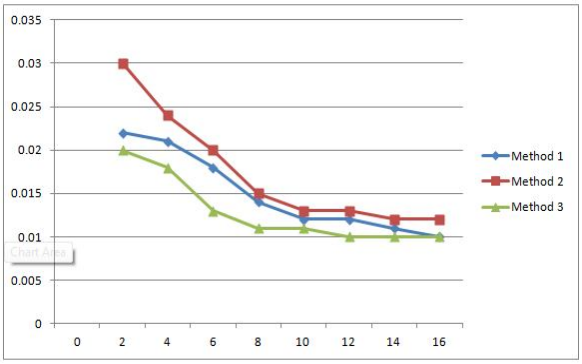
blocks 4×4 and 8×8, watermark was retrieved with the accuracy of more than 90%. Method 3 is not robust to this attack. However, it has shown slightly better results when brightness and contrast was set to 10 and range block size were 8×8 and 16×16.
All these results (Fig 8, 9, 10, and 11) are for 256×256 Lenna image with data length 56 bits and range block sizes 2×2, 4×4, 8×8 and 16×16 shown as below.



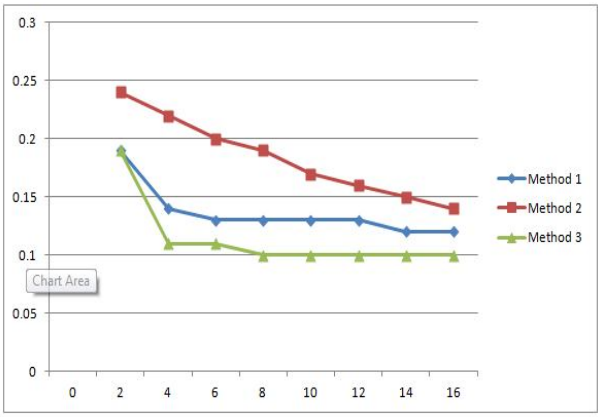
SNR (dB) vs. Block Size (nXn)
Figure 8: SNR verses range block size.



Accuracy (%) vs. Block Size (nXn)
Figure 9: Accuracy verses range block size.



Hiding Time (s) vs. Block Size (nXn)
Figure 10: Hiding Time verses range block size



Extracting Time (s) vs. Block Size (nXn)
Figure 11: Extracting Time verses range block size.

5.6 Comparison existing methods with our results

Several types of tests were carried out and it was seen that a single method is not best under all conditions. SNR was calculated and method 3 came up with the highest value while the method 2 with lowest at all range block sizes. So, there is less distortion in image after putting data into the image, by method 3. Same results were seen in SNR verses number of bits. Here, SNR is very low in case of method 2 as compared to method 3. Methods 2 and 3, for range block size larger than 2×2, shown good results in terms of accuracy in retrieved the data back. Method 3 takes the least time to hide and extract the information from the image for all block size and data length. Method 2 takes quite larger time for larger data length as compared to other methods. On changing the brightness and contrast of the watermarked

image, method 2 shown better results than method 3. On adding noise, method 2 works slightly better than method 3. Watermark put by method 3 is more robust than that of method 2 against JPEG compression. Also, on scaling the watermarked image, method 3 gave better results.

6. CONCLUSION & FUTURE WORKS

In this paper hides the data in spatial domain. This method takes an image file and steganographic data and produces a new image file that contains the steganographic data. The output image is called steganographic image file and it is similar to the input image file. The encoding technique identifies parts of the image that are most suited for data hiding. Image is partitioned in to regions: domain region D and range region R. A key is given by the user. Data which is to be hidden is XOR with key before hiding it. The sub-images within the range region and domain region are called range blocks and domain blocks, respectively.

We developed a new encoding technique, which can find out the possibility to hide maximum amount of data in an image without degrading its quality. Second issue is to make the hidden data robust enough to withstand image processing which do not change the appearance of image. So this technique can also be used for digital watermarking. And also, this technique should be computationally less intensive. The number of bits of data that can be stored depends upon the number of range blocks that have match in domain region. Bigger the range region more is the data that can be stored. But, since range region can not overlap the domain region, on increasing the range region, domain region is reduced which may lead to worse quality of image. So, there is a trade-off between the amount of data and quality of image produced. Increase in tolerance level would allow using all range blocks so that more data can be stored. However low tolerance is desirable in order to give an image that is visually close to the original.

Further research should go towards improving the watermarking program and adding extra functionality. One of these is looking at having multiple watermarks for a single image, so that different parts of the image have a different watermark. There is also the need to further develop the robustness of existing watermarking techniques to combat the ever-increasing attacks on watermarks. We used fixed partitioning scheme. Instead of this, adaptive partitioning scheme can be used, which would yield better results if used as the basis for the data hiding method.

REFERENCES

1. Tsung-Yuan Liu and Wen-Hsiang Tsai. **A New Steganographic Method for Data Hiding in Microsoft Word Documents by a Change Tracking Technique**, IEEE Transactions on Information Forensics and Security, Vol. 2, No. 1, March 2007, pp. 24-30.

2. Dr. Fadhil Salman Abed. **A Proposed Encoding and Hiding Text in an Image by using Fractal Image Compression**, International Journal on Computer Science and Engineering (IJCSE), Vol. 4 No. 01 January 2012, pp. 1-13.

3. Souvik Bhattacharyya, Indradip Banerjee and Gautam Sanyal. **A Novel Approach of Secure Text Based Steganography Model using Word Mapping Method(WMM)**, International Journal of Computer and Information Engineering 4:2 2010, pp. 96-103.

4. Alaa Taqa, A.A Zaidan and B.B Zaidan. **New Framework for High Secure Data Hidden in the MPEG Using AES Encryption Algorithm**, International Journal of Computer and Electrical Engineering, Vol. 1, No. 5 December, 2009, pp. 566-571.

5. Elham Ghasemi, Jamshid Shanbehzadeh and Nima Fassihi. **High Capacity Image Steganography usingWavelet Transform and Genetic Algorithm**, IMECS 2011.

6. Sambasiva Rao Baragada, S. Ramakrishna, M. S. Rao and S. Purushothaman. **Polynomial Discriminant Radial Basis Function for Steganalysis**, IJCSNS International Journal of Computer Science and Network Security, Vol.9 No.2, February 2009, pp. 209-218.

7. Shangping Zhong, Xin Fang, and Xiangwen Liao. **Steganalysis Against Equivalent Transformation Based Steganographic Algorithm for PDF Files**, Proceedings of the 2009 International Symposium on

Information Processing (ISIP'09) Huangshan, P. R. China, August 21-23, 2009, pp. 075-078.

8. Mamta Juneja, Parvinder S. Sandhu, and Ekta Walia. **Application of LSB Based Steganographic Technique for 8-bit Color Images**", World Academy of Science, Engineering and Technology 2009, pp. 423-425.

9. Dr.M.Umamaheswari, Prof.S.Sivasubramanian and S.Pandiarajan. **Analysis of Different Steganographic Algorithms for Secured Data Hiding**, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.8, August 2010, pp- 154-160.

10. Masoud Nosrati, Ronak Karimi and Mehdi Hariri. **An introduction to steganography methods**, World Applied Programming, Vol (1), No (3), August 2011, pp.191-195.