A Steganographic Technique Involving JPEG Bitstream

G. Manikandan[#1], R. Jeya*[2]
[#1]Final year M.Tech student, [*2]Assistant Professor,
[#1*2] Department of Computer Science and Engineering
[#1*2]SRM University, Kattankulathur, Kancheepuram, Tamil Nadu 603 203, India

***Abstract* - Steganography is the technique of hiding secret data into a cover image, audio, text using a key which can be later recovered using the same key. The owner of the sensitive cover image cannot hide the details of the image from the data hider. A Reversible Data Hiding (RDH) technique on a compressed image is proposed along with encryption of the cover image. The cover image is first compressed. The compressed cover image is encrypted with using encryption key. The contents of the image are not disclosed to anyone without the encryption key. The data hider embeds the secret data into the encrypted cover image. At the receiver end the embedded data is extracted from the cover image using the embedding key. The cover image can be decrypted using the encryption key. The cover image can be also decrypted without extraction of the embedded data using the encryption key to produce an approximate image.**

***Index terms* - Steganography, RDH, JPEG bitstream, data hiding, image recovery**

I. INTRODUCTION

Steganography is the process of hiding secret text into a cover medium. The cover medium can be text, audio, image, video, etc. Reversible Data Hiding (RDH) is a steganography approach that emphasizes on the reconstruction of the cover image and the extraction of the secret data hidden in the cover image. RDH can be achieved in using different techniques. Some of the techniques are spatial domain, transform domain, distortion technique, compressed image etc.

Spatial steganography have many versions, all of which involves directly changing some bits in the image pixel values in hiding data. Least significant bit (LSB)-based

Steganography is a simple steganography technique that hides a secret message in the LSBs of pixel values without the introduction of many distortions. Changes made in the LSB value are imperceptible by the human eyes

Transform domain technique is a complex way of hiding information in an image. It requires the use algorithms and transformations that are used on the image to hide information in it. The embedding of data in the frequency domain of a signal is much stronger than embedding in time domain. At present transform domain steganography is widely used, as they hide information in areas of the image that are less exposed to compression, cropping, and image processing.

Distortion techniques requires the knowledge of the original cover image during the decoding process where the decoder functions check for the differences between the original cover image and the distorted cover image which is used for restoring the secret message. During the encoding process, a sequence of changes is added to the cover image by the encoder. Using this technique, a cover image with text hidden is created by the sequence of modification to the cover image. This sequence of modifications is used to match the secret message required to transmit. The secret message is encoded at pseudo-randomly chosen pixels in the cover image. If the cover image with secret data is different from the original cover image at the given message pixel, the message bit is a "1." otherwise, the message bit is a "0."

Compressed image technique suggests the use of spatial, transform and distortion techniques upon a compressed image. In this technique the cover image or the stego-image is subjected to a compression technique before embedding the data into it. The compression of the image increases the security of the steganography process.

II. RELATED WORK

In our present work, we aim to apply RDH technique on a jpeg image bitstream, in order to increase the overall security of the steganography process. Encryption of the cover image so that contents of the cover image are not disclosed. The proposed system also suggests the approximate reconstruction of the original image without extraction of secret data.

A visible watermarking method is proposed in [1]. One way to prove the ownership and the authenticity of the media is digital watermarking. The two commonly used types of watermarking algorithms are visible watermarking and invisible watermarking. In invisible watermarking, the watermark should be transparent and robust. The watermark in visible watermarking should be visible and robust. Traditionally visible watermarking and invisible watermarking were performed by inserting a digital watermark into a digital host signal resulting in a signal which is watermarked. A signal which causes distortion is introduced into the host image during the process of embedding which results in a Peak Signal-to-Noise Ratio (PSNR) loss. In this paper, two lossless visible watermarking algorithms are used. They are Pixel Value Matching Algorithm (PVMA) and Pixel Position Shift Algorithm (PPSA). PVMA uses an intensity mapping function which is bijective to watermark a visible logo. PPSA uses circular pixel shift method to improve the visibility of the watermark in a region of high variance. By using lossless visible watermarking it is easy to prove ownership. It is easy to notice the logo embedded by lossless watermarking without the help of special software. Distortion is not

allowed in military and medical applications. The original host image can be recovered perfectly after the extraction of the watermark by using lossless watermarking technique. Embedding a distortion in visible watermarking is usually larger than that in invisible watermarking technique, so lossless property is used to ensure the signal fidelity. Observation of the logo prevents users from illegal use from the unauthorized users, embedded in visible watermarking, which is visible to the human eyes. However, the embedding of distortion involved in visible watermarking is larger than that of invisible watermarking.

Nasir Memon et al [2] proposed new watermarking method. A watermark is a secret key dependent signal or a distortion that is added to digital data (such as audio, video or an image). This signal or distortion can later be extracted or detected to make an assertion about the data that is embedded. In general, a watermark is either visible or invisible. A visible watermark typically contains a visible message or a company logo denoting the ownership of the image. But the invisibly watermarked content appears identical to the original content, disallowing the user to notice changes to the original content. The presence of an invisible watermark can be determined only by using appropriate watermark extraction or detection algorithm. An interactive buyer–seller protocol for invisible watermarking is proposed in this paper. The seller is not privileged to know the exact watermarked copy that the buyer receives. Hence the seller cannot create duplicate copies of the original content containing the buyer's watermark. If the seller finds an unauthorized copy, the seller can identify the buyer using the watermark in the unauthorized copy, and furthermore the seller can prove this fact to a third party using a dispute resolution protocol. Thus the buyer cannot falsely claim that an unauthorized copy may have originated from the seller. The watermark embedding protocol proposed in this paper is based on a public key cryptography and has little overhead in terms of the total data communicated between the buyer and the seller. The encryption technique used is the RSA cryptosystem is a reliable technique that is believed to be secure if properly used. The dispute resolution protocol is a three-party protocol. The protocol requires the buyer to participate in order to prove his innocence if the seller accuses him of making unauthorized copies. If a buyer refuses to participate then this would be considered as an admission of guilt on the part of the buyer. However the dispute resolution involves a third party which in many cases is undesirable, as the third party can show be biased in case a dispute or argument arises.

N. Ansari, Z. Ni, Y. Shi, et al.,[3] proposed a Reversible Data hiding technique. Data hiding is defined as a process to embed useful data (representing some information) into a cover media. In certain applications, the data to be embedded are closely related to the cover media, such as authentication. In this type of application, invisibility is the major requirement. In most cases, the cover media gets distorted due to data hiding process and cannot be reverted back to the original cover media. That is, distortion exists permanently even after the extraction of the hidden data. In some applications, such as medical diagnosis and law enforcement, there is a preference to reverse the distorted media back to the original media after the hidden data are retrieved. The techniques satisfying the requirement to reverse the marked media back to its original media are referred as reversible or lossless data hiding techniques. This paper proposes the use of a reversible data hiding technique that can embed a large amount of data while keeping high visual quality for all images. The method used in this Reversible Data hiding scheme is called histogram shifting is used on a spatial domain. The zero or the minimum point of the histogram (defined below) and the peak point of the histogram are utilized here. This technique can be applied to almost all types of images. The computation of the proposed technique in this paper is quite simple and the execution time is considerably less. The data is embedded by calculating the zero point or the minimum point and the peak point in the image's histogram. The grey values between the zero point and the point after the peak point are incremented by one. This leaves the grey value after peak point empty. The whole image is again scanned, on encountering the peak point, the data to-be embedded if '1' is added to the pixel value. For the extraction process the whole marked image is scanned. Once the grey value of the maximum point is met, if the value is intact, the "0" is retrieved. If the value is altered, the "I" is retrieved. In this way, the data embedded can be retrieved. The whole image is scanned once again. Once the pixels whose grey value is between the peak point and the zero point is met, the grey value of those pixels will be subtracted by 1. In this way, the original image can he recovered without any distortion

Lokesh Gagnani, et all [4] proposed a steganography method using LSB shifting. Steganography is a data hiding techniques that aims at transmitting a message on a channel where some other information is already being transmitted. The goal of steganography is to hide messages inside the images in such a way that does not allow any unauthorized person to detect that there is a secret message present in the image. Steganography attempts to hide the existence of communication. The basic structure of Steganography is made up of three components are the carrier image, the message and the key. The carrier image or the cover image is used to carry the secret message. The message is nothing the secret message that is to be hided. The key is the embedding key which is used to embed the data into the cover image and also extract the secret data. The Steganography method proposed in this paper is Least Significant Bit (LSB) technique. The cover image is selected and the secret data is embedded in the RGB component of the image. A pixel selection filter to obtain the best areas to hide information in the cover image to obtain a better rate. The filter is applied to Least Significant Bit (LSB) of every pixel to hide information, leaving most significant bits (MSB). The message is hidden using Bit Replacement method. This technique works good for image steganography. To the human eye the stego image will look identical to the carrier image.. For hiding information inside the images, the LSB (Least Significant Bit) method is usually used. To a computer an image file is simply a file that shows different colours and intensities of light on different areas of an image. One of the major disadvantage associated with LSB method is that intruder can change the least significant bit of all the image pixels. In this way hidden message will be destroyed by changing the image quality, a little bit, i.e. in the range of +1 or -1 at each pixel position. LSB method is not immune to noise and compression technique.

Kede Ma et al [5] proposed the room reserving RDH technique. Reversible Data Hiding (RDH) in images is a steganography technique in which the original cover can be losslessly recovered after the embedded message is extracted. This important technique is widely used in medical imagery, military imagery and law forensics, where no distortion of the original cover is allowed. This paper, proposes a RDH method by reserving some space for encryption with a traditional RDH algorithm, and thus it is easy for the data hider to hide data in the encrypted image in a reversible manner. The proposed method can achieve better

reversibility, which is data extraction and image recovery are free of any error. Experiments show that this novel method can embed more payloads for the same image quality as the other methods. In this method the cover image owner either reserves room for embedding secret data before encryption or after. The secret data is then increased. The images containing the secret data is extracted using the embedding key and the decryption of the cover image is done with the help of the encryption key. It is relatively difficult to losslessly vacate room from the encrypted images and sometimes inefficient as manipulation on reserved room can result in losing the secret data. But this technique provides high payload capacity without affecting the quality of reconstruction.

III. SPIHT ALGORITHM

Set partitioning in hierarchical trees (SPIHT) is an image compression algorithm which exploits the similarities from the different sub bands acquired from wavelet decomposition of an image. We use this algorithm to encode or image into a bitstream. The image is first decomposed into a number of sub-bands by using wavelet decomposition. The coefficients of the sub bands obtained are then grouped into sets known as spatial-orientation trees, which efficiently exploit the correlation between the frequency bands. The coefficients of sub bands obtained after wavelet transformation in each spatial orientation tree are then progressively coded from the most significant bit-planes (MSB) to the least significant bit-planes (LSB), starting with the coefficients with the highest magnitude and at the lowest pyramid levels.

The SPIHT algorithm includes the use of 3 lists and sets as following

1. The List of Insignificant Pixels (LIP) contains coefficients that have magnitude lower than the threshold.
2. The List of Insignificant Sets (LIS) are sets of wavelet coefficients that are defined by tree structures and are found to have magnitudes smaller than the threshold (insignificant). The sets exclude the coefficients corresponding to the tree and all sub tree roots and they have at least four elements.
3. The List of Significant Pixels (LSP) is a list of pixels found to have magnitudes larger than the threshold (significant).
4. The set of offspring or the direct descendants of a tree node, o(i , j), in the tree structures is defined by pixel location (i, j). The set of descendants, d(i, j), of a node is defined by pixel location (i, j). m(i, j) is defined as m(i, j) = d(i, j) – o(i, j).

The SPIHT algorithm is as follows

Step1: Initialization: output n= $\log_2 \lfloor (\max_{(i,j)}\{|c_{i,j}|\}) \rfloor$

Set the LSP as an empty list, and add the coordinates (i, j)εH to the LIP,and only those with descendants also to the LIS, as type A entries.

Step2: Sorting Pass:

2.1) for each entry (i, j) in the LIP do:

2.1.1) output Sn(i, j);

2.1.2) if Sn(i, j) = 1 then move (i, j) to the LSP and output the sign of $c_{i,j}$;

2.2) for each entry (i, j) in the LIS do:

2.2.1) if the entry is of type A then

- output $S_n(D(i, j))$;
- if Sn(D(i, j)) = 1 then
  for each (k, l) $\in$ O(i, j) do:
  - output $S_n(k, l)$;
  - if $S_n(k, l) = 1$ then add (k, l) to the LSP and output the sign of $C_{k,l}$;
  - if S_n (k, l) = 0 then add (k, l) to the end of the LIP; if L(i, j) $\neq$ 0 then move (i, j) to the end of the LIS, as an entry of type B, and go to Step 2.2.2);otherwise, remove entry (i, j) from the LIS;

2.2.2) if the entry is of type B then

- output S_n (L(i, j));
- if Sn(L(i, j)) = 1 then add each (k, l) $\in$ (i, j) to the end of the LIS as an entry of type A; remove (i, j) from the LIS.

Step3: Refinement Pass:

for each entry (i, j) in the LSP, except those included in the last sorting pass (i.e., with same n), output the n^{th} most significant bit of $|c_{i,j}|$;Step4: Quantization-Step Update: decrement n by 1 and go to Step 2).

IV. CHAOS MAPPING

Chaos encryption is a simple text encryption technique which is carried out in the following steps

Input: Secret message.

Step 1: Begin

Step 2: A piece of given text information is considered as $w^0 = (w_1^0, w_2^0, \ldots, w_n^0) = \{ w_i^0 | i=1,2,\ldots,n\}$. Firstly, we encode i^{th} character $w_i^0$8 binary bits (w_{i7}, w_{i6}, w_{i5}, w_{i4}, w_{i3}, w_{i2}, w_{i1}, w_{i0}). The secret message must ended (.).

Step 3: Calculate number of rows that text contained and number of letters in each row.

Step 4: The chaotic map to generate the chaotic number that equals to number of rows, the following equation

$$X_{n+1}=rX_n(1-X_n) \quad \ldots(1)$$

Step 5: Convert each chaotic number to binary, that's number of bits in equal to number of bits for letter in each row.

Step 6: In each row, shift sequence of letters that equivalents to '0' to the beginning of row.

Step 7: The new encrypted character is obtained from the bitxor of the new binary value and the existing text value. Steps 4 to 7 are repeated till every text is encrypted.

Step 8: End

Output: Encrypted Secret message

V. SYSTEM ARCHITECTURE

The system architecture of the proposed scheme is sketched in Figure 1.

Image encoding

In this module the cover image is first subjected to SPIHT encoding. SPIHT encoding is a lossless compression technique which reduces the size of the cover image by organizing it into a stricter structure. The SPIHT encoding reduces the size of the image by encoding it into a bitstream.

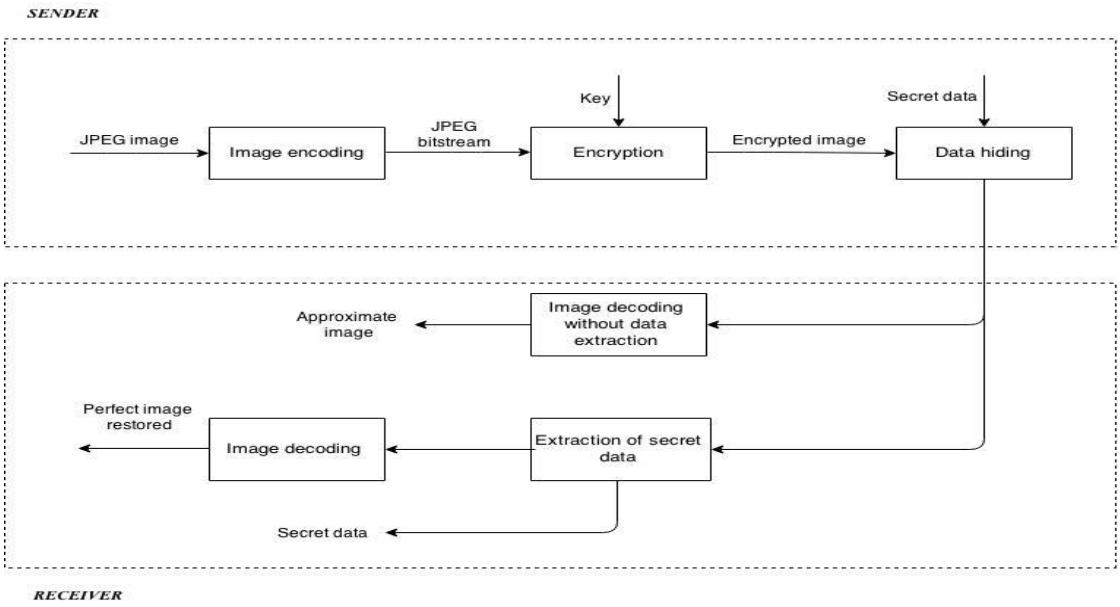


figure 1. System architecture

Image encryption

The encoded cover image is encrypted using bitxor technique. The bitxor technique involves generating a random key from the encoded bitstream of the same length as encoded bitstream. A new bitstream is then generated from the bitxor of encoded bitstream and random key generated.

(a) (b)

Picture (a) is the original image and picture (b) is the encrypted form of the original image

Text encryption

The text to be hidden is encrypted using chaotic mapping. For every individual character from the text to be hidden, the ASCII value of the character and the threshold value from chaotic mapping are taken. A new value is obtained from applying bitxor on these two values. This new value is encrypted text value.

Data embedding

The encrypted secret data is converted to binary format before being embedded into the encrypted cover image bitstream. Every individual value of the bitstream and the text data are compared for the data embedding process. An empty array is first created which is an index array is created. This index array is created to help us in extraction of the secret data. If the value of the image bitstream and encrypted text are found to be same then in the index array we denote it by 'E' and the bit value of the bitstream is unchanged. If the value of the bitstream is 1 and secret text is 0, denote it as 'L' and the text is embedded into the bitstream by changing the value of the bitstream to a new value. The new value is obtained by applying bitxand on the corresponding bitstream data and the encrypted text character. Similarly, if the bitstream data is 0 and the encrypted text value is 1 denote it as 'H' in the index array; the embedding process is done by changing the value of the image bitstream to a new value, obtained from the bitxor of corresponding bitstream data and the encrypted text character.

Data extraction

The data extraction process can be done using the reverse process of the embedding technique, along with the use of the index array.

Image decryption and decoding

The decryption and the decoding of the bitstream can be achieved by reversing the encryption and encoding processes respectively. The decryption process requires the same key used while encrypting the bitstream. The encrypted bitstream after extraction of the text can be perfectly reconstructed to its original image by decrypting the bitstream followed by decoding the bitstream. An approximate image of the original image can be constructed if the decryption and the decoding process are carried out without the extraction of the secret text.

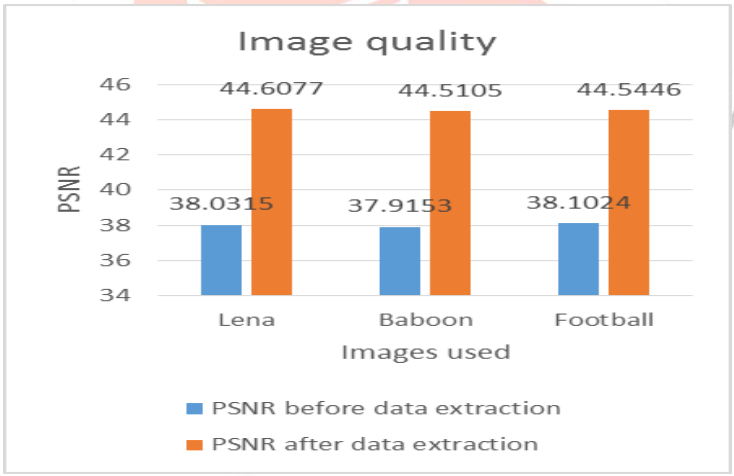
VI. EXPERIMENTAL RESULTS

We conduct our experiment on standard 256×256 grey-scale images. We observe that reconstructed image (b) which has data embedded in it is an approximate construction of the original image; reconstructed image (c) is identical to the original image (a)

(a) (b) (c)

Picture (a) is the original image and picture (b) is the approximate construction of the original image and picture(c) is the prefect reconstruction of the original image

On experimenting with few other 256×256 images, we found that the images reconstructed after the extraction of the secret data have higher PSNR value than the images which were reconstructed without the extraction of the secret data



PSNR values fo various images after reconstruction of images before and after data extraction

VII CONCLUSION

Here we have proposed a steganography technique involving a jpeg bitstream. We have first encoded the image to bitstream. The jpeg bitstream is encrypted using bitxor technique along with the use of a key. We then encrypt the text to be hidden using chaos mappin. The encrypted text is embedded into the encrypted bitstream. The secret text in new bitstream can be extracted by reversing the embedding process. The cover image in the bitstream format can be reconstructed with or without the extraction of the data. The reconstructed image without extraction of the secret data is only an approximate image of the original image with a lower PSNR value; the reconstructed image after extraction of the data is identical to the original cover image and has a relatively higher PSNR value.

REFERENCES

[1] Au, Chi-Wang, Oscar C, Ho Hoi-Ming Wong and Shu-Kei Yip. “Lossless Visible Watermarking”, 2006
[2] N. Memon and P. W. Wong, “A buyer-seller watermarking protocol,” IEEE Trans. Image Process., vol. 10, no. 4, pp. 643–649, Apr. 2001.
[3] N. Ansari, Z. Ni, Y. Shi, et al., “Reversible data hiding,” IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362, Mar. 2006.
[4] Lokesh Gagnani, Rahul Joshi and Salony Pandey, “Image Steganography With LSB”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)., Volume 2, Issue 1, January 2013

[5] K. Ma, W. Zhang, X. Zhao, et al., "Reversible data hiding in encrypted images by reserving room before encryption," IEEE Trans. Inf. Forensics Security, vol. 8, no. 3, 553-562, 2013.
[6] Zhenxing Qian, Xinpeng Zhang, Shuozhong Wang, "Reversible data hiding in encrypted images by reserving room before encryption", IEEE Trans. Multimedia, vol. 16, NO. 5, AUGUST 2014
[7] Ton Kalker and Frans M.J. Willems, "CAPACITY BOUNDS AND CONSTRUCTIONS FOR REVERSIBLE DATA-HIDING",IEEE conference. DSP, vol. 1; 2002
[8] Shiguo Lian,, Zhongxuan Liu et al," Commutative Encryption and Watermarking in Video Compression", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 17, NO. 6, JUNE 2007