Volume 6, Issue 1, January 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Secure Reversible Image Data Hiding with Contrast Enhancement

Farsana N U
M.Tech Student, KMEA, CSE Department,
MG University, Kerala India

Selin M
Associate Professor, KMEA, CSE Department,
MG University, Kerala India

Abstract— In this paper, reversible data hiding (RDH) algorithm is proposed for digital images. To improve the visual quality of the host image, an algorithm is proposed that enhances the contrast of the host image. XOR encryption and decryption algorithms are also used to improve security of the hidden data. For data embedding, highest two bins in the histogram are selected and histogram equalization can be performed by repeating the process. Inorder to completely recover the original image, side information is also embedded along with the message bits in the host image.

Keywords— Contrast enhancement, histogram modification, location map, reversible data hiding, visual quality, XOR cipher.

I. INTRODUCTION

REVERSIBLE DATA HIDING (RDH) is a type of data hiding techniques whereby the host image can be recovered completely. Being lossless makes this technique suitable for medical and military applications. Most of the multimedia data embedding techniques modify the host images in order to insert the additional information. Hence these techniques are irreversible. This highlights the need for Reversible data embedding techniques. Reversible data hiding eliminate this limitation by completely recovering the host image. Reversible data hiding can be used in sensitive applications such as military and medical field, where no permanent modification is allowed on the host image.

Since several years, the protection of multimedia data is becoming very important. The protection of this multimedia data can be done with encryption or data hiding algorithms. Data compression is important to decrease the transmission time. So a method is needed that comprises compression, encryption and data hiding in a single step. So far, a new challenge is introduced in embedding data in encrypted images. A new algorithm is proposed for encrypted images to remove the embedded data before the image decryption with contrast enhancement.

In this paper, a new secure RDH algorithm is proposed with contrast enhancement. Image contrast enhancement can be achieved by histogram equalization. The proposed algorithm helps to perform the data embedding and contrast enhancement at the same time by modifying the histogram of pixel values. Then, image with embedded message is encrypted using XOR cipher. Key generation for both encryption and decryption are performed randomly. Firstly, the highest two bins are selected from the histogram. If there is no bounding value (i.e. 0 or 255), the two peaks can be split into two adjacent bins. The bins between the highest peaks are unchanged while outer bins are shifted outward. To achieve the satisfactory contrast enhancement, the highest bins in the modified histogram can be further chosen to be split. It also increases the embedding capacity. Overflows and underflows occur due to the presence of bounding values. In order to avoid this bounding pixel values are pre-processed and location map is generated. In the host image, location map is embedded along with the message bits and side information so that complete recovery of the original image can be achieved.

II. RELATED WORK

There are many approaches implemented for reversible data hiding. They can be roughly categorized into three types: lossless compression based methods, difference expansion (DE) methods, and histogram modification (HM) methods. Weiming Zhang, Biao Chen, and Nenghai Yu proposed a decompression algorithm [2] for embedding data. This method uses three RDH schemes that use binary feature arrangement as covers. One scheme is used for spatial images, one scheme for JPEG images, and pattern exchange scheme for binary images.

Lixin Luo, Zhenyong Chen, Ming Chen, Xiao Zeng, and Zhang Xiong proposed a method [3] for embedding large amount of covert data into images. This method uses the concept of interpolation error, which is found by subtracting the interpolation image from the original image to embed bit “1” or “0” by expanding it additively or leaving it unchanged.

Vasiliy Sachnev, Hyoung Joong Kim, Jeho Nam Sundaram Suresh, and Yun Qing Shi proposed [4] a reversible watermarking algorithm using sorting and prediction. With less distortion more data can be embedded because of the use of sorted prediction errors and reduced size of location map. Prediction errors are recorded by using sorting technique based on magnitude of its local variance.

Bhaskara Reddy et.al proposed [5] an Effective Algorithm of Encryption and Decryption of Images Using Random Number Generation Technique and Huffman coding. In this paper, an image is converted into pixels matrix. Then replace that matrix into some fixed numbers. Then encryption is performed. The key for encryption is generated using random generation method. The encrypted image is randomly transpositioned; convert it into one dimensional encrypted array

and finally applying Huffman coding. Decryption is the reverse process of encryption. This method is implemented for security in images.

Subhanya R.J , Anjani Dayanandh N proposed [6] a method for Difference Expansion Reversible Image Watermarking Schemes Using Integer Wavelet Transform Based Approach. Authors present a method for image watermarking to guard intellectual properties and to secure the content of digital images. This method embeds image or text data into a video based on Integer Wavelet Transform and minimizes the distortion between the actual and watermarked image and also increases Peak Signal to Noise Ratio. IWT is used to hide the keys in the corresponding gray/colour component of the cover image.

III. SECURE RDH ALGORITHM WITH CONTRAST ENHANCEMENT

A. Data Embedding by Histogram Modification

The algorithm [1] is proposed for gray-level images. Given an 8-bit gray-level image, the image histogram can be calculated by counting the pixels with a gray-level value j for $j \in \{0, 1, \ldots, 254, 255\}$. The image histogram can be is denoted by h_I . The number of pixels with a value j is represented by $h_I(j)$. Assume I consist of N different pixel values. Then in h_I, there are N non-empty bins, from which two peaks (i.e. highest two bins) are selected and the corresponding smaller and bigger values are represented by I_S and I_R, respectively. For a pixel counted in h_I with value i, data embedding is performed by:

$$i' = \begin{cases} i-1, & for\ i < I_S \\ I_S - b_k, & for\ i = I_S \\ i, & for\ I_S < i < I_R \\ I_R + b_k, & for\ i = I_R \\ i+1, & for\ i > I_R, \end{cases} \quad (1)$$

Where i' is the modified pixel value, and b_k is the k-th message bit (0 or 1) to be hidden. By applying Eq. (1) to every pixel counted in h_I, totally $h_I(I_S)+h_I(I_R)$ binary values are embedded. If there is any bounding value (i.e. 0 or 255) pre-processing is needed. Pre=processing is done by modifying the pixel values of 0 and 255 to 1 and 254 respectively. Pre-processing is explained in the section III B. Given that there is no bounding value. So there will be N+2 bins in the modified histogram. The bins between the highest peaks are unchanged while outer bins are shifted outward so that each of the peaks can be split into two adjacent bins (i.e. I_S-1 and I_S, I_R and I_R+1 respectively).

To extract the embedded data, the peak values I_S and I_R needs to be retrieved. By excluding 16 pixels in I from histogram computing, we can keep this peak values. The least significant bits (LSBs) of those 16 pixels are calculated and included in the binary values to be hidden. For data embedding, Eq. 1 is applied to each pixel counted in h_I. After that the values of I_s and I_R (each with 8 bits) is used to replace the LSBs of the 16 excluded pixels by bit-wise operation. The peak values are needed to be retrieved in order to extract the embedded data. The histogram of the marked image I' is calculated excluding the 16 pixels. Then the following operation is performed on any pixel counted in the histogram and with the value of I_S-1, I_S, I_R or I_R+1:

$$b'_k = \begin{cases} 1, & if\ i' = I_S - 1 \\ 0, & if\ i' = I_S \\ 0, & if\ i' = I_R \\ 1, & if\ i' = I_R + 1, \end{cases} \quad (2)$$

Where b'_k is the k-th binary value extracted from the marked image I'. The order of performing the extraction operations are same as that of the order of performing embedding operations. To recover the original value of every pixel counted in the histogram, the following operation is performed according to Eq. (1):

$$i = \begin{cases} i'+1, & for\ i' < I_S - 1 \\ I_S, & for\ i' = I_S - 1\ or\ i' = I_S \\ I_R, & for\ i' = I_R or\ i' = I_R + 1 \\ i'-1, & for\ i > I_R + 1, \end{cases} \quad (3)$$

From the extracted binary values, the original LSBs of 16 excluded pixels are obtained. To recover the original image, the excluded pixels can be restored.

B. Pre-process for Complete Recovery

It is required that all pixels counted in h_I are within {1, …, 254}. Overflow or underflow will be caused, if there is any bounding pixel value (0 or 255) by histogram shifting. Inorder to avoid overflow or underflow, the histogram needs to be pre-processed prior to the histogram modification operations. The pixel value of 0 is modified to 1 and the pixel value of 255 is modified to 254. Possible change of each pixel is ±1, therefore no overflow or underflow will be caused. A location map with the same size as the original image is generated to memorize the pre-processed pixels. It is done by assigning 1 to the modified pixel, and 0 to that of an unchanged one including 16 excluded pixels. The location map can be pre-computed and included into the binary values to be hidden. In the extraction and recovery process, pixels modified in the pre-process can be identified and obtained from the data extracted from the marked image. The original image can be completely recovered, by restoring the original values of those pixels accordingly.

C. Contrast Enhancement

It is required that the numbers of 0s and 1s in the message be almost equal, therefore, each of the two peaks in the histogram is split into two adjacent bins with similar or same heights. By applying the Eq. (1) to every pixel counted in

the histogram, the highest two bins in the modified histogram are further chosen to be split. It will increase the hiding rate. The histogram equalization effect can be achieved by repeating the same process of splitting each of the two peaks into two adjacent bins with the similar heights. So that, data embedding and contrast enhancement are simultaneously performed. Suppose there are L pair number of the histogram peaks to be split, the range of pixel values from 0 to L-1 are added by L while the pixels from 256-L to 255 are subtracted by L in the pre-process. Assume that L is a positive integer. A location map is generated by assigning 1s to the modified pixels, and 0s to the others.

The location map can be pre-computed and compressed to be firstly embedded into the host image. In contrary, the value of L, the size of the compressed location map, and the previous peak values are embedded with the last two peaks to be split, whose values are stored in the LSBs of the 16 excluded pixels. In the extraction process, the last split peak values are retrieved and the data embedded with them are extracted by applying Eq. (2). For restoring the histogram Eq. (3) can be used. The data embedded with the previously split peaks can be extracted by processing them pair by pair. To identify the pixel values modified in the pre-process, the location map is obtained from the extracted data.

D. Encryption and Decryption

We are using XOR for encryption and decryption. In cryptography, a simple cipher is XOR cipher. XOR cipher is based on the following principles:

$$A \oplus 0 = A,$$
$$A \oplus A = 0,$$
$$(A \oplus B) \oplus C = A \oplus (B \oplus C),$$
$$(B \oplus A) \oplus A = B \oplus 0 = B$$

Any string can be encrypted based on the above logic using bitwise XOR operator. In this paper, key generation for both encryption and decryption are performed randomly. The image with embedded message is encrypted using this key. Then we get the encrypted image. At the decryption side, image is decrypted using the same key. Reversible Data Hiding with Encryption provides high security.

E. Procedure of the Proposed Algorithm

The flowchart of the proposed algorithm is shown in Fig.1. Note that number of histogram bins to be split is L.

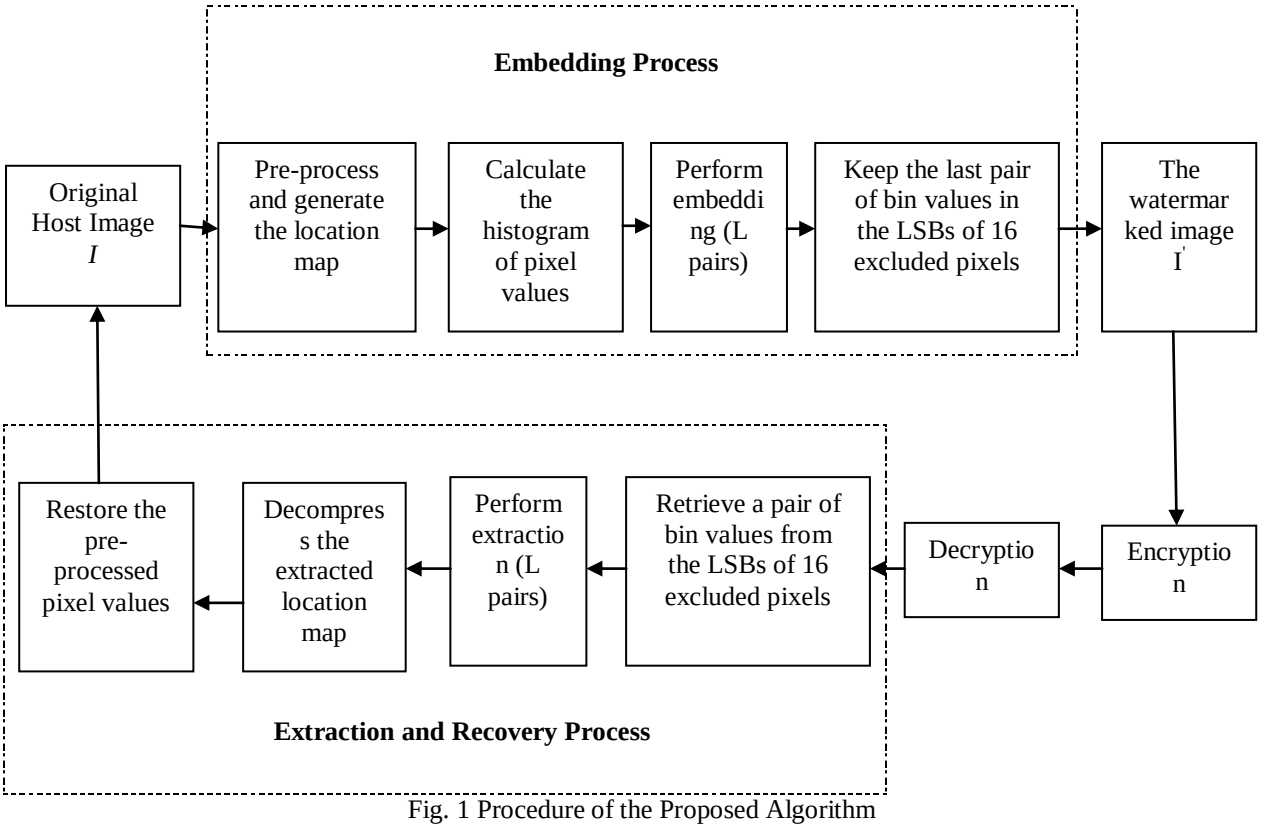

Fig. 1 Procedure of the Proposed Algorithm

Embedding procedure includes the following steps:

1) Pre-process: The pixels in the range of [0, L-1] are added by L and [256-l, 255] are subtracted by L as pre-process excluding the first 16 pixels in the bottom row. A location map is generated to memorize these points. To reduce the length of the location map JBIG2 compression standard is used.
2) Calculation of image histogram without counting the first 16 pixels in the bottom row.
3) Embedding: By applying Eq. (1) to every pixel counted in the histogram, the highest two bins are split for data embedding. The process of splitting the two peaks in the modified histogram is repeated until L pairs are split. Before message bits, the bit stream of the compressed location map is embedded. The value of L, the length of

the compressed location map, the LSBs collected from the16 excluded pixels, and the previous peak values are embedded with the last two peaks to be split.

4) To form the marked image, the LSBs of the 16 excluded pixels are replaced by lastly split peak values.
5) Encryption: The image with embedded message is encrypted using XOR cipher. Key generation for encryption is done randomly.

The extraction and recovery process includes the following steps:

1) Decryption: The encrypted image is decrypted using XOR cipher. Key for decryption is same as that of encryption.
2) To know the last two split peaks, the LSBs of the 16 excluded pixels are retrieved.
3) The value of L, the length of the compressed location map, the original LSBs of the 16 excluded pixels, and previously split peak values are found by extracting the data embedded with the last two split peaks by using Eq. (2). Then by using Eq. (3), the recovery operations are carried out by processing all pixels except the 16 excluded ones. To extract the data embedded, the extraction and recovery process is repeated until all of the split peaks are restored.
4) From the extracted binary values, the compressed location map is obtained and decompressed to the original size.
5) Pixels modified in the pre-process are identified from the decompressed location map. Among these pixels, a pixel value is subtracted by L if it is less than 128, or increased by L otherwise. To avoid ambiguity, the maximum value of L is 64. Finally, original image can be recovered exactly by writing back the original LSBs of 16 exclude pixels.

IV. CONCLUSIONS

Reversible Data Hiding techniques are getting popular because the host image can be completely recovered. In this paper, a secure reversible data hiding algorithm with contrast enhancement is proposed. The image with embedded data is encrypted to increase security.

ACKNOWLEDGMENT

The authors would like to sincerely thank the editors and anonymous reviewers for their valuable comments.

REFERENCES

[1] Hao-Tian Wu, Jean-Luc Dugelay, Yun-Qing Shi, "*Reversible Data Hiding with Contrast Enhancement*," IEEE Signal Processing Lettters, vol. 22, no. 1,January2015.

[2] W. Zhang, B. Chen, and N. Yu, "*Improving various reversible data hiding schemes via optimal codes for binary covers*," IEEE Trans.Image Process., vol. 21, no. 6, pp. 2991–3003, Jun. 2012.

[3] L. Luo et al., "*Reversible image watermarking using interpolation technique*," IEEE Trans. Inf. Forensics Security, vol. 5, no. 1, pp. 187–193,Mar. 2010.

[4] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y.-Q. Shi, "*Reversible watermarking algorithm using sorting and prediction*," IEEE Trans.Circuits Syst. Video Technol., vol. 19, no. 7, pp. 989– 999, Jul. 2009.

[5] Dr. T. Bhaskara Reddy, Miss. Hema Suresh Yaragunti , Mr.T. Sri Harish Reddy, Dr. S. Kiran " *An Effective Algorithm of Encryption and Decryption of Images Using Random Number Generation Technique and Huffman coding*" Hema Suresh Yaragunti et al, Int.J.Computer Technology & Applications,Vol 4 (6),883-891 .

[6] Subhanya R.J (1), Anjani Dayanandh N (2)" *Difference Expansion Reversible Image Watermarking Schemes Using Integer Wavelet Transform Based Approach*". International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 International Conference on Humming Bird (01st March 2014).

