

An Improved Approach for Reversible Data Hiding in Encrypted Images Using Image Expansion

Riya Reji and Dr.N. Vishwanath

Abstract— *Reversible data hiding is a technique to embed additional message into some distortion unacceptable cover media, such as military or medical images, in such a way that the original cover content can be perfectly restored after extraction of the hidden message. In order to securely share a secret image with some other person, the sender/owner may encrypt the image before transmission. Nowadays, more attention is paid to reversible data hiding (RDH) in encrypted images, since the original cover can be recovered without any loss after the embedded data is extracted while protecting the image content's confidentiality. All previous data embedding methods apply RDH algorithms to the original pixels, which may cause some errors during image restoration. In order to avoid errors during restoration of image, we reserve space for embedding the data by image expansion before encrypting the image. Based on the size of data to be embedded the image is expanded by adding additional pixels. These additional pixels are used for embedding the data. After reserving the space, the image is encrypted by using encryption key. Data hider can embed the data into the reserved space in the encrypted image and encrypt the data using the data hiding key. In order to reduce the overhead while transmitting the expanded image, the image is compressed to reduce its size. After decompressing the image, the user can separately extract the data using data hiding key and image using the encryption key.*

Keywords--- *Reversible Data Hiding, Image Expansion, Image Encryption, Data Embedding, Compression*

I. INTRODUCTION

DATA security means safeguarding data from illegitimate users or hackers and providing high security to prevent data alteration. The area of data security has attained more attention due to the enormous increase in data transfer rate over the internet. In order to enhance the security during data transfer through the internet, many techniques like cryptography, steganography etc have been developed.

Data hiding, a form of steganography, embeds data into digital media. But in most cases of data hiding, some

distortion may occur in the cover media due to data hiding and cannot be restored back to the original media. In some applications, such as medical diagnosis and law enforcement, it is necessary to restore the marked media back to the original cover media after the embedded data are retrieved for some legal considerations.

Reversible data hiding facilitates immense possibility of applications to link two sets of data in such a way that the cover media can be losslessly recovered after the embedded data have been extracted out, thus providing an opportunity for handling two different sets of data. Reversible data hiding in images is a technique that embeds data in digital images by altering the pixel values for secret communication, and the embedded image can be recovered to its original state after the extraction of the secret data.

Encryption is an effective and popular means of privacy protection. In order to securely send a secret image through internet, a content owner can encrypt the image before transmission. In some applications of data hiding, the embedded carrier images are further encrypted to prevent the image from being analyzed to reveal the presence of the embedded data. In some other applications, the owner of the carrier image might not want the other person, including data hider, get the idea about the content of the carrier image such as military images or confidential medical images before data hiding is actually performed.

A reversible data hiding scheme for encrypted image is desirable. In reversible data hiding, after encrypting the whole data of an uncompressed image by a stream cipher, the secret message can be embedded into the image by altering a small portion of encrypted data. With an encrypted image containing secret message, one may firstly decrypt it using the decryption key, and the decrypted version is similar to the original image. According to the data-hiding key, using the property of spatial correlation in original image, the embedded data can be successfully extracted and the original image can be perfectly restored.

Some attempts have been made to embed data using reversible data hiding techniques in encrypted images. However, all these techniques make space for embedding the data by applying various techniques to the original pixels of the image. Even though the image can be restored, some errors can occur during image restoration.

When network bandwidth and storage space are limited, an image to be transmitted has to be compressed. It is necessary to protect confidential image data during transmission from unauthorized access. Compression reduces the storage space required to represent a given quantity of

Riya Reji, Department of computer science and engineering, Toc-H Institute of Science and Technology, CUSAT, Ernakulam, India. E-mail: riyareji95@gmail.com

Dr.N. Vishwanath, Department of computer science and engineering, Toc-H Institute of Science and Technology, CUSAT, Ernakulam, India. E-mail: vishwa10370@gmail.com

information. Image compression is of two types: lossy and lossless. Lossless compression schemes are reversible so that the original data can be reconstructed exactly, while lossy schemes accept some loss of data in order to achieve higher compression. Lossless compression can be used for text, medical images and legal documents etc. whereas lossy compression is used for natural images, speech signals etc.

In the present paper, we propose a new method in which the space for embedding the data is reserved by using image expansion. In this method the image is expanded based on size of data to be embedded by adding additional pixels. The data is embedded into these additional pixels, hence the original pixels are not transformed to embed the data. This method achieves real reversibility and large amount of data can also be embedded.

In this paper section 2 discuss about the related works. Section 3 describes about how to reversibly embed the data by using image expansion. Section 4 illustrates the experimental result of the algorithm. Finally section 5 concludes the paper with the highlights.

II. RELATED WORKS

Some attempts have been made for data embedding in encrypted images using reversible data hiding techniques. In [6], Zhang divided the encrypted image into several blocks. Room for embedding the bit was made by flipping 3 LSBs of the half of pixels in each block. The data extraction and image recovery is performed by finding which part has been flipped in one block. This flipped part in the block is identified with the help of spatial correlation in decrypted image.

Zhang's method was improved by Hong *et al.* [7] by further exploiting the spatial correlation using a different estimation equation and side match technique at the receiver side to achieve much lower error rate. In these two methods, the encrypted image should be decrypted first before extracting the data.

To separate the data extraction and image decryption, Zhang[9] emptied out space for data embedding by compressing the encrypted images. The method in [9] compressed the encrypted LSBs of the image to vacate space for embedding data by finding syndromes of a parity-check matrix, and at the receiver side the spatial correlation of decrypted images is used for restoration of the image.

These techniques try to vacate room directly from the encrypted image, so all of them are subject to some error rates on data extraction and/or image restoration. In order to overcome this, Kede Ma[1] proposed a technique in which the space for embedding the data is reserved before encrypting the image. The data hider can embed the data into the reserved space in the encrypted image. At the receiver side the data and image can be extracted separately.

In this technique, the space for embedding the data is reserved using lsb embedding. For lsb embedding the image is

first divided into smooth and complex blocks. The lsb planes of complex block are then embedded into the smooth block using RDH algorithms. After reserving the space for data embedding the image is encrypted. The data hider then embeds the data into the encrypted image. At the receiver side, the data is extracted and the image is restored.

Many works have been made on compressing encrypted images. Mingyu Li *et al.*[8] used a RC5 stream cipher based scalable encryption scheme for low complexity transparent transcoding. CCSDS compression, method is used which consist of two part DWT and Bit plane coding. V.Radha, D.Maheswari [9] proposed image encryption algorithm that consists of two parts: scrambling of plain-image and mixing operation of scrambled image using discrete states variables of chaotic maps. Discrete Cosine transform is used for compression.

III. PROPOSED SYSTEM

In the proposed method, space for embedding the data in the image is reserved before encryption. Space is reserved by expanding the image. After reserving the space, the image is encrypted by using encryption key. Data hider can embed the data into the encrypted image using the data hiding key. After embedding the data, the embedded image is compressed to reduce overhead of transmission. At the receiver side, the user first decodes the image and then extracts the image and data. If a user has data hiding key, then he can extract the data. With the encryption key, user can recover the image. If a user has both the keys, then the user can obtain both the data and image. The proposed system can be described as follows:

• Sender Side

At the sender side, the original image is first expanded to reserve space for embedding the data. The expanded image is then encrypted using the encryption key. Then the data is embedded into the encrypted image based on the data hiding key to create the marked encrypted image. The marked encrypted image is then compressed to create the encoded image. The below diagram shows the working of the proposed system at sender side:

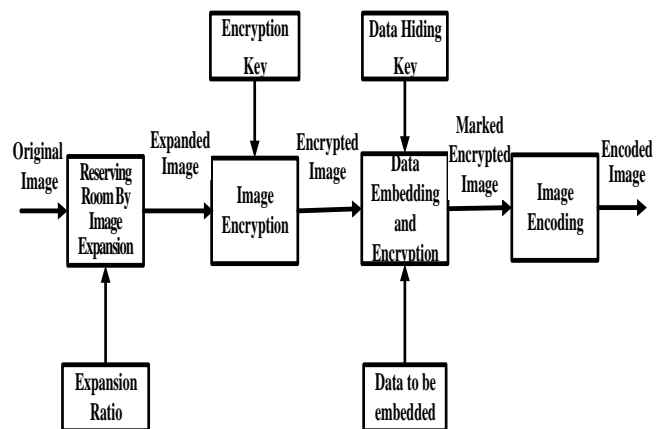


Fig1: Sender Side

A. Image Expansion:

In order to expand the image, the ratio required for the expanding the image is calculated. The ratio is calculated from the size of data to be embedded. In order to calculate the ratio, the message is converted to byte array. The size of expanded image and expansion ratio is calculated as:

$$\text{Expanded image size} = \text{Size of original image} + \text{Length of byte array.}$$

$$\text{Expansion Ratio} = \frac{\text{Size of expanded image}}{\text{Size of original image.}}$$

Based on the ratio, new size of the image is calculated and expanded image is created. The pixels in the original image are mapped into to the pixel position in the expanded image obtained by multiplying the original pixel position with the ratio. The additional pixels will have intensity value equal to the average of values of neighbouring four pixels. The ratio is encrypted using the data hiding key and encryption key and is stored in the image. The expanded image is thus created.

B. Image Encryption:

After image expansion, the image is encrypted using the encryption key. The image is encrypted using blowfish algorithm. Blowfish encryption algorithm is a symmetric block cipher that can be effectively used for encryption and safeguard and of data. The blowfish encryption algorithm has been analyzed considerably and is gaining acceptance as an encryption algorithm. The block size of blowfish algorithm is 64 bits and key can be any length up to 448 bits before encrypting them.

C. Data Embedding and Encryption:

After encrypting the image, the data can be embedded into the free space reserved before encryption. The data is embedded into the additional pixels in the expanded image. After data embedding the message is encrypted using the data hiding key to produce the marked image. The data is encrypted using AES algorithm. The Advanced Encryption Standard (AES) algorithm is capable of using cryptographic keys of 128, 192, and 256 bits and decrypt data in blocks of 128 bits. The AES algorithm is divided into four different phases, which are executed in a sequential way forming rounds. The encryption is achieved by passing the plaintext through an initial round, 9 equal rounds and a final round. In all the phases of each round, the algorithm operates on a 4x4 array of bytes (called the State).

D. Image Encoding(Compression):

After embedding the data, the image is then compressed using Huffman Coding to reduce the overhead during transmission of images. Huffman coding is a form of statistical coding which attempts to reduce the amount of bits required to represent a string of symbols. The algorithm accomplishes its goals by allowing symbols to vary in length. Shorter codes are assigned to the most frequently used symbols, and longer codes to the symbols which appear less frequently in the string.

• Receiver Side:

The working of the proposed system at receiver side is as follows:

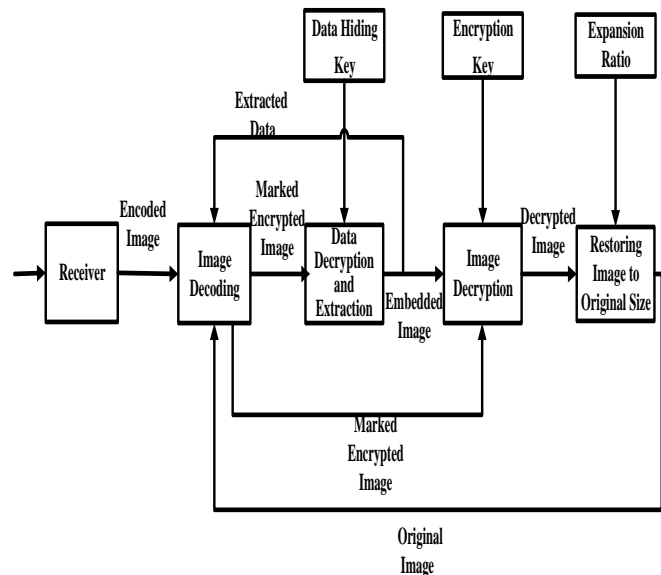


Fig 2: Receiver Side

At the receiver side, the encoded image is first decompressed to obtain the marked encrypted image. The data can be extracted from the marked encrypted image using the data hiding key. In order to restore the image, the marked encrypted image is first decrypted using the decryption key. Then the original image is reconstructed from the decrypted image.

A. Image Decoding:

At receiver side, the image is first decompressed using Huffman decoding to obtain the marked encrypted image from which the data and image can be restored.

B. Data Decryption and Recovery:

Data can be extracted from the embedded image with the help of data hiding key. In order to extract the data we first decrypt the expansion ratio encrypted using data hiding key. Using the expansion ratio, the data is extracted and then it is decrypted to obtain the embedded message.

C. Image Decryption:

In order to restore the image is first decrypted using the Blowfish algorithm. The image can be decrypted after or before extracting the data. The expansion ratio encrypted using encryption key is also extracted.

D. Image Restoration:

After obtaining the decrypted image, it is restored to the original image using the expansion ratio. The additional pixels are removed from the expanded image to obtain the original image.

IV. EXPERIMENTAL RESULT

The proposed system was implemented in Java. From the experiments carried out, it was found that the data and image can be restored without any error. The experimental result is shown below:



Fig 3: Original Image

First the original image is expanded based on the expansion ratio and the expanded image is obtained.



Fig 4: Expanded Image

The expanded image is then encrypted.



Fig 4 Encrypted Image

The data is then embedded into the encrypted image.

```

ImageExpansion - Notepad
File Edit Format View Help
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package imageroom;
import java.awt.*;
import java.awt.image.*;
import java.awt.image.BufferedImage;
public class ImageExpansion {
    BufferedImage from;
    BufferedImage to;
    ImageInfo iinfo;
    public int limit;
    TwoD[] rev_points;
    public ImageExpansion(int requiredLength,BufferedImage fr)
    {
        limit=0;
        from=fr;
        iinfo=new ImageInfo();
        iinfo.exp_width=fr.getWidth();
        iinfo.exp_height=fr.getHeight();
        long current=iinfo.exp_width*iinfo.exp_height;
        double expansion_ratio=requiredLength/(double)current;
        expansion_ratio=Math.sqrt(expansion_ratio);
        if(expansion_ratio<1.0)
        {
            return ;
        }
        int req_height=(int)(Math.ceil(expansion_ratio*iinfo.exp_height));
        int req_width=(int)(Math.ceil(expansion_ratio*iinfo.exp_width));
        iinfo.org_height=req_height;
        iinfo.org_width=req_width;
        to=new BufferedImage(req_width+1, req_height, BufferedImage.TYPE_INT_RGB);
        double rev_expansion=(current)/(double)(req_height*req_width);
        rev_expansion=Math.sqrt(rev_expansion);
        iinfo.rev_expansion_ratio=rev_expansion;
        rev_points=TwoD.getPoints(iinfo);
        for(int i=1;i<req_width;i++)
        {
            for(int j=0;j<req_height;j++)
            {
                color c1=getColor(i-1, j, rev_expansion);
                to.setRGB(i, j, c1.getRGB());
            }
        }
    }
}
    
```

Fig 5: Data to be Embedded

After data embedding, the image is compressed.

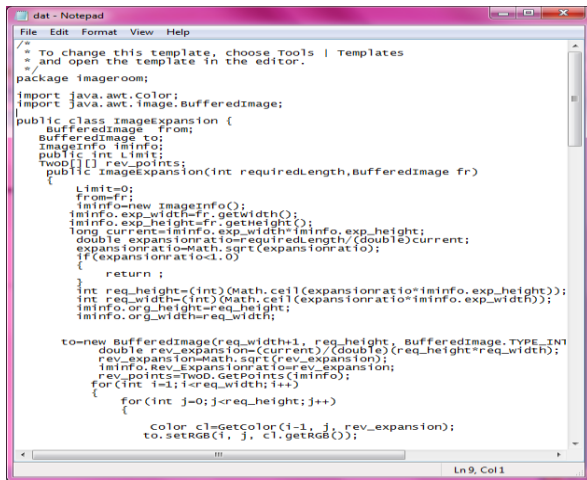


Fig 6 Compressed Image

At the receiver side, the image is first decoded to obtain marked encrypted image. Then the data is decrypted and extracted from the image and the image is decrypted and restored to its original size. Thus the receiver can obtain the data and image without any error.



Fig 7 Extracted Image



```

dat - Notepad
File Edit Format View Help
/* To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package imageerom;
import java.awt.Color;
import java.awt.image.BufferedImage;
public class ImageExpansion {
    BufferedImage fr;
    BufferedImage to;
    ImageInfo iminfo;
    public int limit;
    Twod[] rev_points;
    public ImageExpansion(int requiredLength,BufferedImage fr)
    {
        limit=0;
        fr=fr;
        iminfo=new ImageInfo();
        iminfo.exp_width=fr.getWidth();
        iminfo.exp_height=fr.getHeight();
        long current=iminfo.exp_width*iminfo.exp_height;
        double expansionratio=requiredLength/(double)current;
        expansionratio=Math.sqrt(expansionratio);
        if(expansionratio<1.0)
        {
            return ;
        }
        int req_height=(int)(Math.ceil(expansionratio*iminfo.exp_height));
        int req_width=(int)(Math.ceil(expansionratio*iminfo.exp_width));
        iminfo.org_height=req_height;
        iminfo.org_width=req_width;
        to=new BufferedImage(req_width+1, req_height, BufferedImage.TYPE_INT_ARGB);
        double rev_expansion=(current)/(double)(req_height*req_width);
        rev_expansion=Math.sqrt(rev_expansion);
        iminfo.Rev_expansion=1/rev_expansion;
        rev_points=Twod.GetPoints(iminfo);
        for(int i=1;i<=req_width;i++)
        {
            for(int j=0;j<req_height;j++)
            {
                color c1=getColor(i-1, j, rev_expansion);
                to.setRGB(i, j, c1.getRGB());
            }
        }
    }
}
Ln 9, Col 1

```

Fig 8 Extracted Data

V. CONCLUSION

The implementation of this proposed system demonstrates that the data and image can be restored back to its original state using image expansion. Since we are altering any original pixels, the image can be restored without loss of quality. The proposed method can achieve real reversibility and can separate image data extraction ie, there is no order for extraction of data and image. It also can embed large amount of data. As we are reserving room before encryption the data embedding has become effortless. The encryption of image and data help us to maintain the security and confidentiality.

REFERENCES

- [1] Kede Ma, Weiming Zhang, Xianfeng Zhao, Member, IEEE, Nenghai Yu, and Fenghua Li, "Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption", IEEE Transactions On Information Forensics and Security, vol. 8, no. 3, March 2013.
- [2] J. Fridrich and M. Goljan, "Lossless data embedding for all image formats," in Proc. SPIE Proc. Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents, San Jose, CA, USA, Jan. 2002, vol. 4675, pp. 572–583.
- [3] X. Zhang, "Separable reversible data hiding in encrypted image," IEEE Trans. Inf. Forensics Security, vol. 7, no. 2, pp. 826–832, Apr. 2012.
- [4] Z. Ni, Y. Shi, N. Ansari, and S. Wei, "Reversible data hiding," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362, Mar. 2006.
- [5] P. Tsai, Y. C. Hu, and H. L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," Signal Process., vol. 89, pp. 1129–1143, 2009.
- [6] X. Zhang, "Reversible data hiding in encrypted images," IEEE Signal Process. Lett., vol. 18, no. 4, pp. 255–258, Apr. 2011.
- [7] W. Hong, T. Chen, and H. Wu, "An improved reversible data hiding in encrypted images using side match," IEEE Signal Process. Lett., vol. 19, no. 4, pp. 199–202, Apr. 2012.
- [8] Mingyu Li, Xiaowei Yi and Hengtai Ma, "A Scalable Encryption Scheme for CCSDS Image Data Compression Standard" 978-1-4244-6943-7/ IEEE pp. 646-649, 2010
- [9] V.Radha, D.Maheswari, "Secured Compound Image Compression Using Encryption Techniques", 978-1-4244-5967-4/ IEEE 2010
- [10] 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
- [11] Aditee Gautam, Meenakshi Panwar and Dr.P.R Gupta, "A New Image Encryption Approach Using Block Based Transformation Algorithm", (IJAEST) International Journal Of Advanced Engineering Sciences And Technologies Vol No. 8, Issue No. 1, 090 – 096

- [12] Irfan.Landge¹, Burhanuddin Contractor², Aamna Patel³ and Rozina Choudhary⁴, "Image encryption and decryption using blowfish algorithm," World Journal of Science and Technology 2012, 2(3):151-156.
- [13] 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.
- [14] L. Luo et al., "Reversible image watermarking using interpolation technique," IEEE Trans. Inf. Forensics Security, vol. 5, no. 1, pp. 187–193, Mar. 2010.
- [15] D.M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," IEEE Trans. Image Process., vol. 16, no. 3, pp. 721–730, Mar. 2007.

BIOGRAPHIES



Riya Reji received her B.Tech in Computer Science & Engg from MG University, Kerala and is currently pursuing M.Tech in Computer Science & Engg with specialization in Data Security from CUSAT, Kerala. Her research interest are Data security, Image security, Data hiding.



Dr.N.Vishwanath is currently working as professor, in Toc-H Institute of Science and Technology, Kerala. He has completed his master of technology from Manonmaniam Sundaranar Univeristy. He has got his first Ph.D from Newcastle university and he completed his second Ph.D from Manonmaniam Sundaranar Univeristy. He has published many papers in many reputed International journals.