Data Hiding and Extraction From Images Using Side Match Algorithm.

Mrs. Madhavi. P., MTech., Asst., professor, k. Ammulu, A. Bindhu Sajitha, G. Praveena, G. Sri Nikhila, D. Pavani

Abstract-Digital image processing is a fast-developing arena with increasing applications in science and engineering. One among the most interested field is data hiding in encoded images. The reversibility feature is used in this method. It is an improved version of Zhang's Reversible Data Hiding method in encrypted images which is proposed already, but the only disadvantage is about bit error and robustness. Reversible data hiding embeds the data into a digitally encrypted image by changing the pixel value for secret communication and the embedded image can be improved to its original state after the removal of the secret data. Finally, Recovery can be accomplished by examining the block smoothness. The main aim of this application is where the owner of the carrier might not want the other person, including data hider before the data hiding is performed, such as military images or confidential images. This paper measures the smoothness of blocks and uses the Side-Match Scheme to decrease the error rate of extracted-bits further.

Terms—Encrypted image, reversible data hiding

INTRODUCTION:

Reversible data hiding is a method that implants data in digital images by shifting the pixel principles for secret communication and the embedded images can be recovered to its unique state after the secret data is extracted.

Many reversible data hiding methods have been projected recently embeds data bits by expanding the difference of two consecutive pixels uses a lossless compression tech- where is the XOR operation and. Convert exclusive to create extra spaces for transfer data bits changes the binsof image histograms to leave an unfilled bin for data embedment. obtain the encrypted image adopts the difference expansion and histogram shifting for data embedment embeds data by shifting the histogram of prediction errors while considering the local activity of pixels to extra improve the quality of stego image. Traditionally, data hiding is used for secret communication. In some submissions, the embedded carriers are further encrypted to prevent the transporter from being analysed to reveal the presence of the embedment. Other applications could be for when the owner of the carrier might not want the other person, including data hider, data hiding is performed earlier to know the carrier content, such as military images or confidential medical images. In this case, the owner of the content must translate the data before data can be embedded in the message done by the data hider and the original image is recovered.

Recent researchers have worked on RDH to decrease the error obtained during the process of extraction. Zhang proposed an RDH scheme which splits the encrypted image into blocks of desired sizes and embeds additional data by flipping the 3 LSB's. Excerpts data and cover image based on the smoothness of the image block. Hong enhanced smoothness calculation and proposed side matching mechanism to decrease the number of bits incurred error. Considering the drawbacks of Zhang's method that abandonments the four boundaries in a block and Hong's method taking the successive pixels, Extracted-bit error rate (EER) on Liao's method decreases monotonically with the increase in block size. We propose a method which varies from these methods in the procedure of selecting pixels during embedding and reduces the error during extraction of bits.

To encrypt the cover image report embedding is performed. Cover image can be encrypted using the exclusive-or (XOR) operator to every bit of pixels by using a serial key.

RELATED WORK:

The lossless data embedding structure for encoded images familiarized in has two distinct versions: a mutual method i.e., where the hidden data can be extracted and the image can be restored after the image was decrypted and a discrete method in which the hidden data can be decoded from the encrypted image, but after decryption the host image is recovered.

Let us assume C as an encrypted form of image *I*. Each bit plane of *C*, C_t, $(1 \le t \le 8)$ is computed as:

$$C_t = I_t \oplus r_t \qquad (1)$$

anywhere$\oplus$is the exclusive-or operator and *r* is a pseudorandom bit stream sequence made by an encryption key.

A. The joint method of [8]

This method embeds *L* bits in *nL* pixels of the encrypted image, $n \ge 1$. First *nL* pixels of *C* are selected based on a data hiding key. After an encrypted pixel $C(I,,j)$ is chosen, the pixels that form its calculation context cannot be selected for data embedding. A data bit is fixed by flipping the *t* bit of *n* designated pixels:

$$0 \qquad (\sim C_t(i, j), \quad \text{if } b = 1,$$

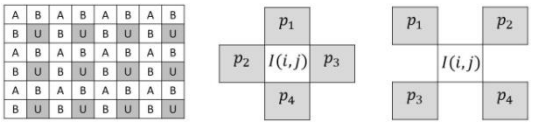
$C_t(i, j)$ = (2) $C_t(i,j)$,

if $b = 0$. where ~is the not operator and $b \in \{0,1\}$ is the hidden bit.

A watermarked version of I is obtained by decrypting each bit plane of C^0 using the encryption key:

$$I'_t = C'_t \oplus r_t \quad (3)$$

If the user has admission to the data hiding key, the entrenched data can be extracted. The *null*watermarked pixels are first



(a) (b) (c)

Fig. 1. The proposed pixel distribution (a), the prediction context used by set *B* and the scheme of [8] (b), the prediction context used by set *A* (c).

designated using the hiding key. For each of those pixels, $I^{00}(i, j)$ is made by flipping $I_t^0(i, j)$. Then $\hat{I}(i, j)$, the predicted valueof $I(i, j)$ (a subjective average on the prediction context), is used to determine b, the hidden bit inserted in each group of n pixels:

$$|x| = \begin{cases} x, & \text{where. X at x >= 0} \\ -x, & \text{if } x < 0 \end{cases}$$

If $b = 1$, all n pixels in the current group are substituted with their corresponding $I^{00}(i, j)$ value, otherwise the pixels continue unchanged. Both n (the number of pixels in a group) and t (the bit plane used for data hiding) have a direct influence on the exactness of equation (4). A larger value for n recovers the precision of (4) but reduces the available space for the hidden data. Similarly, a larger value for t also advances the precision of (4), but the watermarking distortion increases.

$$b = \begin{cases} 0, & \text{if } \sum_{k=1}^{n} \left| I'(i_k, j_k) - \hat{I}(i_k, j_k) \right| \le \sum_{k=1}^{n} \left| I''(i_k, j_k) - \hat{I}(i_k, j_k) \right| \\ 1, & \text{if } \sum_{k=1}^{n} \left| I'(i_k, j_k) - \hat{I}(i_k, j_k) \right| > \sum_{k=1}^{n} \left| I''(i_k, j_k) - \hat{I}(i_k, j_k) \right| \end{cases} \quad (4)$$

B. The discrete method of [8]

In this method based on the data hiding key L additional bits in L pixels (as opposed to nL pixels) are selected to embed.. The secret bit b is directly inserted in the host pixel $C(i,j)$ by substituting its t bit:

$$C'_t(i, j) = b \quad (5)$$

A user with access to the hiding key can decode the hidden data by reading the t bits of the L selected pixels. After (3) is used to decrypt $I^0(i,j)$, the closest value to $\hat{I}(i,j)$ between $I^0(i,j)$ and $I^{00}(i,j)$ is selected as the original value of $I(i,j)$.

Note that all separate lines can correctly decipher the hidden data, but the returning step remains affected by the possibility of errors. Also note that this approach requirements a larger value for t in order to compensate for the embedding in a single pixel instead of a group of n. In [8] the author recommends $t \geq 7$ and adds a filtering stage after decryption in order to remove the distortions announced by the watermark. This additional riddling step draws attention to the existence of the hidden data in the encrypted image and represents a grave security risk. The proposed scheme does not have this drawback.

PROPOSED METHOD:

In this technique, the smoothness of the block is analyzed to obtain accurate data from the data image. In order to calculate the block smoothness, the four borders of each block do not join. This may decrease correctness of data extraction rate, especially when the block size is small. For example, consider a block size 8×8, among 64 pixels only 28 pixels around 43.75% are in the four borders.Border pixels are not involved in the block smoothness calculation, and the percentage increases as the block size is decreased.

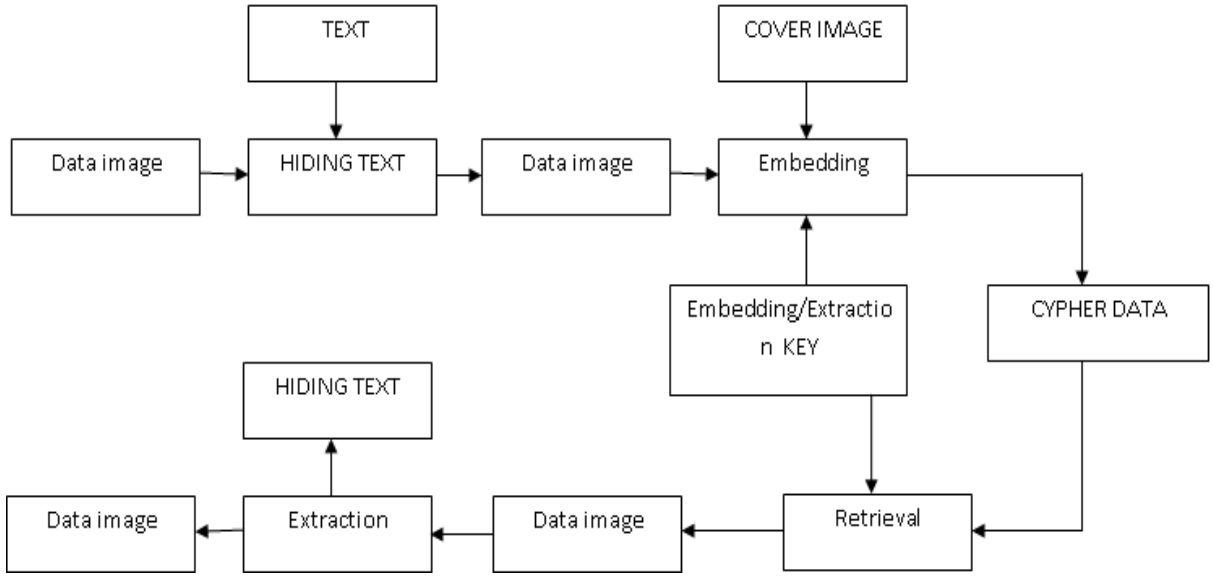


Fig a: proposed method block representation

However, the embedded bits can be extracted by estimating the smoothness of a single block. However, flipping 3 Least significant bits of these complex blocks doesn't cause a significant increase in complexness. By seeing the observations, this paper proposes an improved version for a block smoothness, calculated efficiently. In this new smoothness estimation, the summation of the absolute of two neighbouring pixels is employed. Both the extraction and recovery are calculated from the most noticeable changes to the least ones in smoothness. Besides, the side-match technique is adopted in the evaluate of block smoothness by appending the border of recovered blocks to the unrecovered blocks. Therefore, only the calculation of smoothness and the process of image recovery was adressed.

A. Calculation of Block Smoothness

The smoothness of an image block can be evaluated by calculating the absolute difference of neighbouring pixels. If the summation of absolute differences is larger, the more complex the image blocks are. Therefore, smoothness of the block can be estimated by calculating the summation of the vertical and horizontal absolute differences of pixels in image blocks using the following equation:

$$f = \sum_{u=1}^{s_2} \sum_{v=1}^{s_1-1} |p_{u,v} - p_{u,v+1}| + \sum_{u=1}^{s_2-1} \sum_{v=1}^{s_1} |p_{u,v} - p_{u+1,v}| \quad (7)$$

were $p_{u,v}$ represents the pixel values located at position (u, v) of a given image block of size $s_1 \times s_2$. Equation (7) fully exploits the absolute difference between two consecutive pixels in both vertical and horizontal directions and thus, the smoothness of blocks can be better estimated.

b. Data Extraction and Image Recovery Using Side Match

Let the cover image be *I* and the encrypted image with message embedded be *I(n).* firstly, the data image is taken to embed the text, and which resulted as a data image. The text or message is going to hide using of bit plane analysis i.e., 1 to 8-bit plane. Using of pixel arrangement when we give the bit plane as 1 then the text is going to hide at upper left column of data image and by giving 8th bit plane the text is going to embed at the lower right side of image.

For the 4th level of security the data image is embedded with cover image using the serial key combination. By that it results to the encoding the cover image in zigzag manner. By the using of same serial key combination the decrypter can decrypt the transferred data of image.

By the purpose of cloud computing and high official military defence purposes the size capability of encrypted data is high.

Fig b: The data image along with hided textFig c: cover image to embed with data image

But the precaution we should taken During implementation of hiding the text in the data image and

extraction of text from the same data image an in-depth analysis was carried out to prevent the complexity.
During embedding the data image into the cover image and while extracting the data image from the cover object an utmost care was taken with extreme supervision to extract the data image from the cover image without any loses.

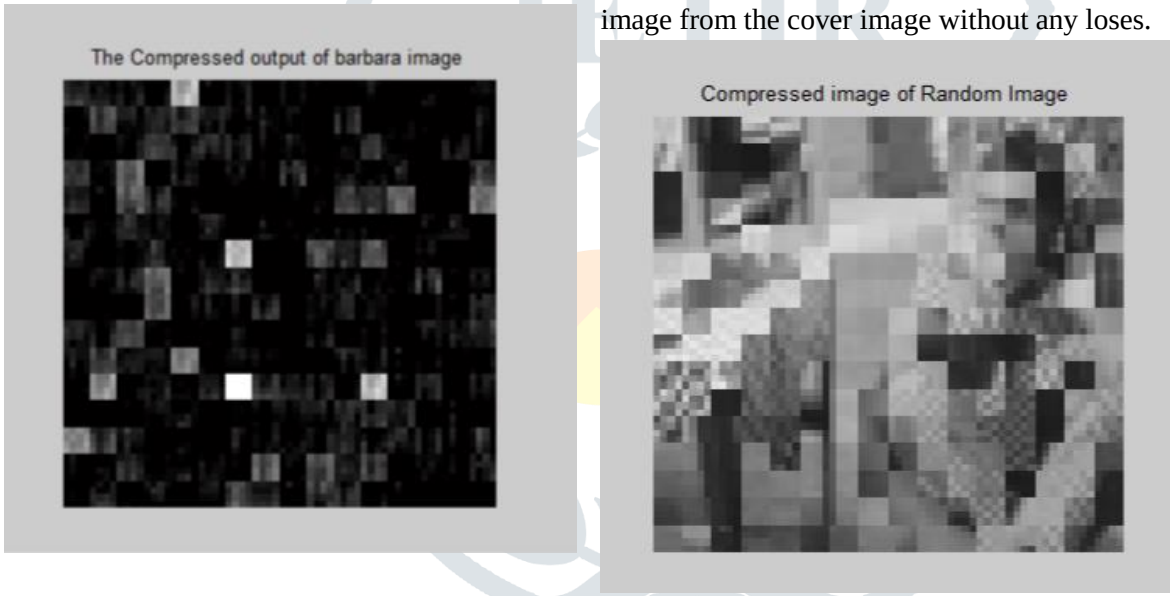


fig d and e: encoded outputs of complete image through sender

EXPERIMENTAL STUDY:

We can use the image level of size 512 x 512, counting Lena, Barbara as the test image. To demonstrate the performance of the proposed method, we take Lena as an example. Fig d and e represented a compressed and encoded image of cover image (Barbara) which does not easily encode with any includer and we can see the proposed method can be recovers the original image block i.e., more accurately than the existing methods.

CONCLUSION:

Reversible data hiding (RDH) is sketch more attention now-a-days because of its aptitude to recover the cover deprived of any distortion. Encryption is also used along with RDH for confidentiality protection. In the currently available approaches RDH is implemented in encrypted images by vacating space after encryption. The verifiedefficiency of the proposed technique can thus help reducing error while securing data at the third-party center. The proposed method can realize real reversibility, separate data extraction and great improvement on the quality of the decrypted image.

In the present systems data is embedded plain text. To increase the security some symmetric key algorithms can be used for encrypting the data to be embedded so that the encrypted data can be embedded in the image. During the first extraction at the receiver side encrypted data is retrieved at the output.

REFERENCES:

Johnson, z. daric, s, jaidia information hiding. Steganography and watermarking – attacks and counter measures: Kluwer academic press. norwall 2000

J. Tian, "Reversible data embedding using a difference expansion," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 8, pp. 890–896, 2003.

M. Celik, G. Sharma, A. Tekalp, and E. Saber, "Lossless generalized lsb data embedding," *Image Processing, IEEE Transactions on*, vol. 14, no. 2, pp. 253–266, 2005.

Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 16, no. 3, pp. 354–362, 2006.

D. Thodi and J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *Image Processing, IEEE Transactions on*, vol. 16, no. 3, pp. 721–730, 2007.

L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, "Reversible image watermarking using interpolation technique," *Information Forensics and Security, IEEE Transactions on*, vol. 5, no. 1, pp. 187–193, 2010.

W. Hong, T.-S. Chen, and H.-Y. Wu, "An improved reversible data hiding in encrypted images using side match," *Signal Processing Letters, IEEE*, vol. 19, no. 4, pp. 199–202, 2012.

S.-W. Jung, L. T. Ha, and S.-J. Ko, "A new histogram modification based reversible data hiding algorithm considering the human visual system," *Signal Processing Letters, IEEE*, vol. 18, no. 2, pp. 95–98, 2011.