Secure and Improved Capacity Reversible Data Hiding Based on Prediction Error Histogram Modification

Sabeena P M[1] , Sreejith S Nair[2]

[1]*M.tech. Scholar,* [2]*Assistant professor, Electronics and Communication Engineering,*

Jawaharlal College of Engineering and Technology, Palakkad, Kerala, India

Abstract - Here present a modified reversible data hiding technique with effective embedding capacity, compressed data structure and ensures security of the embedded message. In Reversible data hiding techniques that are proposed so far, Prediction- error expansion (PEE) is the one of the most efficient technique. This method is based on modification of prediction- error histogram (PEH), in which secret data is embedded by adaptively selecting the expansion bins. To avoid the problematic situations of overflow and underflow, a location map which is a binary sequence of length is established in this method. In proposed implementation run length encoding (RLE) is used for lossless compression of location map. Also in this method, Boundary expansion with row and column expansion is done so that prediction error expansion is exploited considering the boundary pixels too, which enhances the embedding capacity. Here the proposed scheme is coded using VHDL and synthesized using Xilinx ISE. Also MATLAB is used to view the final embedded and extracted image, and to know PSNR. The embedding distortion produced by the proposed algorithm is less. Experiment results show that the proposed algorithm works well in different types of images.

Keywords: Prediction-error expansion, Prediction-error histogram, Reversible Data Hiding, Run length coding

I. INTRODUCTION

Reversible Data Hiding (RDH) is a method to recover the original cover image after the hidden data is extracted. So main requirement of RDH method is reversibility or lossless recovery of original image. In some applications such as medical diagnosis, low enforcement etc. the availability of initial image is on high demand. In such cases original image must be completely recovered. The marking techniques satisfying this requirement are referred to as reversible, lossless, distortion-free, or invertible data hiding techniques [15]. In invertible data hiding using lossless compression introduced by Awrengjeb [1] the space to hide the data is found by compressing the bit planes that offers minimum redundancy. The main disadvantages of the system are low hiding capacity and in noisy images the data become visible. Tian[5] devised a high capacity Reversible data hiding technique called Difference Expansion(DE). DE methods calculate the differences of neighboring pixel values, and select some difference values for the difference expansion (DE). The major drawback of Tian's scheme is the lack of capacity control. Ni et al. proposed a reversible data hiding method [15] based on histogram modification. This method uses peak and zero points to achieve low distortion, but it provides low capacity. However, its EC is quite low and this method does not work well if the cover image has a flat histogram. Ni et al.'s [15] method is improved by Lee et al. [10] by using the histogram of difference image. Thus, a regular-shaped histogram is utilized in Lee et al.'s method. This histogram is centered at origin and has rapid two-sided decay which is more suitable for RDH. Moreover Lee et al's method can also be implemented in equivalent by modifying the two-dimensional pixel-intensity-histogram according to a pixel-pair-mapping (PPM) which is an injective mapping defined on pixel-pairs [12]. In [12] by considering each pixel-pair and its context, a sequence consisting of pairs of difference values is computed. The prediction-error expansion (PE) algorithm was developed by Thodi and Rodriguez [3]. DPM is a natural extension of expansion embedding and shifting techniques used in current histogram-based RDH methods. Although Difference Expansion method improves the embedding capacity but its image quality drops off quickly. Instead of the difference value in DE, the prediction-error is utilized in PEE for expansion embedding. Thus, unlike DE where only the correlation of two adjacent pixels is considered, the local correlation of a larger neighbourhood is exploited in PEE.

II. RELATED WORK

The most effective and extensively exploited RDH technique is the PEE technique which is firstly proposed by Thodi and Rodriguez [5]. Conventional PEE (C-PEE), the PEE with adaptive embedding (A-PEE), the PEE with optimal expansion bins selection (O-PEE), and the PEE which combines both adaptive embedding and optimal expansion bins selection (AO-PEE) are some of the PEE related RDH works that are proposed so far. In PEE the first step is generation of prediction error histogram (PEH) for prediction error and second step is modification of

PEH[13]. Here expansion bins are adaptively selected and secret data is embedded into the selected prediction error. In generation of prediction error histogram the cover image pixels are collected in one-dimensional sequence as $(x_1,x_2,\ldots\ldots\ldots\ldots,x_N)$, where N is total number of pixels. Then prediction value $x_i\hat{}$ is computed for each pixel x_i. And the prediction error is computed by subtracting the prediction value from original pixel i.e., prediction error is given by

$e_i = x_i - x_i\hat{}$, where e_i is the prediction error .

In conventional PEE (C-PEE) the prediction error is modified as follows. Here m € {0,1}.

$$\tilde{e}i = \begin{cases} ei + m & ;for\ ei = 0 \\ ei - m & ;for\ ei = -1 \\ ei + 1 & ;for\ ei > 0 \\ ei - 1 & ;for\ ei < 0 \end{cases} \quad (1)$$

The bins -1 ,0 are expanded to embed data bits the other bins are shifted for creating vacancy that ensures reversibility The cover pixel is modified to $x_i\tilde{} = xi + \tilde{e}_i$ and generate marked pixel. The above method will stop when all the all secret data bits are embed in the cover image.

A-PEE (Adaptive–Prediction error expansion) is an extension to C-PEE which better exploit the image redundancy. In APEE the complexity measurement n_i is computed for each x_i. And for a preselected threshold T > 0 is assumed, then the pixel satisfying ni < T will be embedded. For ni > T x_i is ignored and unmodified.

In O-PEE method, for a prediction error histogram we can select the bins *a* and *b* where *a* < *b* for expansion. In this situation the bins larger than b or smaller than a need to be shifted to ensure reversibility while the pixels between the bins *a* and *b* will remain unchanged. Thus distortion decreases as compared to C- PEE.

AO-PEE is the combination of adaptive embedding and optimal expansion bins selection. This combined embedding provides better performance. In AO-PEE three parameters are determined complexity, threshold, and expansion bins (*a*,*b*).

III. METHODOLOGY

The proposed reversible data hiding method i.e. “Secure and improved capacity reversible data hiding based on prediction-error histogram modification” present a modified reversible data hiding technique with effective embedding capacity. This method is based on modification of prediction- error histogram (PEH). For a prediction error histogram two expansion bins are adaptively selected in prediction-error histogram and data embedding is realized based on prediction error expansion. Based on an estimation of embedding distortion, the expansion bins can be effectively determined such that the distortion is minimized.

In expansion embedding, for a prediction error histogram *h*, the bins *a* and *b* (where *a* < *b*) ,will be expanded, and the bins larger than *b* or smaller than *a* will be shifted, while other bins are unmodified. Specifically, for a cover pixel x_i, prediction-error e_i is modified to get $\tilde{e}$ according to *(a,b)* as

$$\tilde{e}i = \begin{cases} ei & ;for\ a < ei < b \\ ei + m & ;for\ ei = b \\ ei - m & ;for\ ei = a \\ ei + 1 & ;for\ ei > b \\ ei - 1 & ;for\ ei < a \end{cases} \quad (2)$$

where m ∈{ 0,1} is a to-be-embedded data bit. Then, modify x_i to $\tilde{x}i = \hat{x}i + \tilde{e}$, where $\tilde{x}i$ is the marked pixel. The above procedure continue until the all the secret data bits are embedded. The data extraction and image restoration can be summarized as follows. For a marked pixel $\tilde{x}i$ t then the original prediction- error can be recovered according to $\tilde{e}$ and *(a,b)*as

$$ei = \begin{cases} \tilde{e} & ;for\ a < \tilde{e} < b \\ \tilde{e} - 1 & ;for\ \tilde{e} > b \\ \tilde{e} + 1 & ;for\ \tilde{e} < a \end{cases} \quad (3)$$

The block diagram of proposed framework for data embedding and extraction is shown in Fig 1 and Fig 2 respectively. One of the most effective prediction based embedding is Rhombus Prediction method. In Rhombus prediction method a prediction for a pixel x_i is computed from four pixels surrounding the pixel in Rhombus fashion.

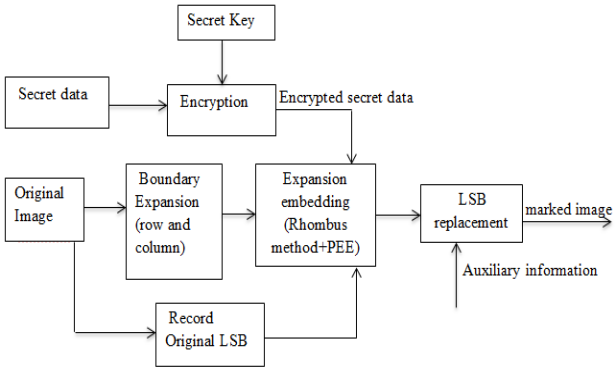


Fig.1: Data Embedding.

Unlike the previous RDH method the secret data is encrypted using secret key. In Data embedding block diagram as in Fig1 the original image is undergone through boundary expansion and then encrypted secret data and Original LSB of last S_{aux} pixels is embedded into shadow

and blank pixels respectively using Rhombus Prediction method. In Data Extraction as in Fig 2 we first extract the auxiliary information from last S_{aux} pixels and then restore the original LSB back. Then we retrieve the encrypted secret data from marked image and by using secret key which is same as that is used in encryption we extract the secret data. Here block cipher method is used as an encryption method.

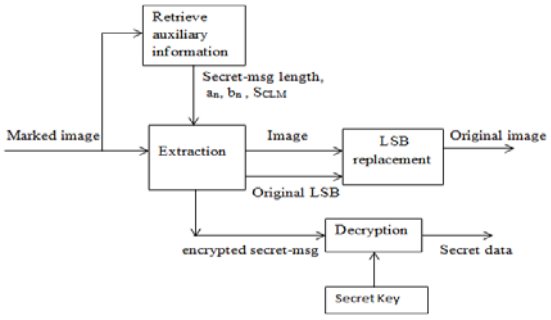


Fig.2: Data Extraction

The proposed framework of Reversible data hiding mainly consists of following stages:

A) Generation of encrypted secret image
B) Boundary expansion of cover image
C) Data embedding(Rhombus method)
D) Location map compression
E) Extraction and Image recovery

A. Generation of encrypted secret image

The main requirement of the data hiding process is the invisibility of hidden data and the protection of data from unauthorized users. When it is desired to send the confidential or secure data over an insecure medium it is customary to encrypt the secret data. This ensures the security of the secret data. Here block cipher encryption method is used. Block cipher method applies on blocks of data rather than single bit. Encryption process converts plain text (original data) to cipher text. Fig 3 shows block cipher encryption of secret data.

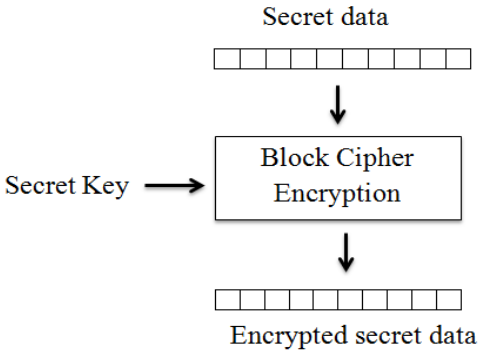


Fig 3: Block cipher encryption of secret data

In proposed method secret data that is embedded into cover image is binary data which is of 20 bits. The size of data can be varied. Here binary form of an image (which is a logo) is taken as secret data.

In block cipher encryption of secret data encrypted data is obtained by XOR secret data with encryption key i.e. we have

$$enc\_msg = sec\_msg \text{ XOR } enc\_key$$

where *enc_ msg* - encrypted secret data,

sec_ msg - secret message,

XOR - binary operator,

enc_ msg- encrypted message

Fig 4 shows Block cipher decryption of secret data which decrypts the original secret data. The same key that is used in encryption process is used in decryption process. So only authorized users who knows the secret key can retrieve the secret data.

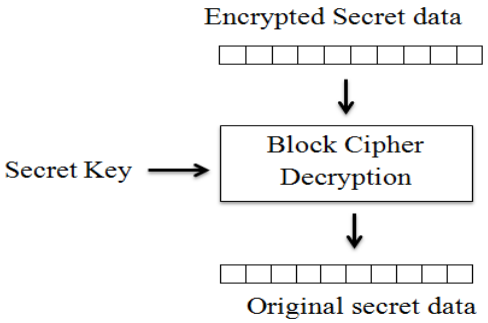


Fig 4: Block cipher decryption of secret data

B. Boundary Expansion of cover image

In the RDH method described in [13] except the pixels located in the borders , the shadow pixels are scanned from left to right and from top to bottom to derive the cover sequence $(x_1,...,x_N)$. So in [13] the border pixels are not considered which is required for the prediction of some pixels. But if we consider the boundary pixels too for prediction process, the numbers of pixels to be embed increases. As a result embedding capacity can be increased The proposed RDH method uses Boundary expansion by expanding the row and column. So 511×511 image become 513×513 after expansion. Here for boundary expansion we make a copy of row and column of original cover image itself.

Steps:

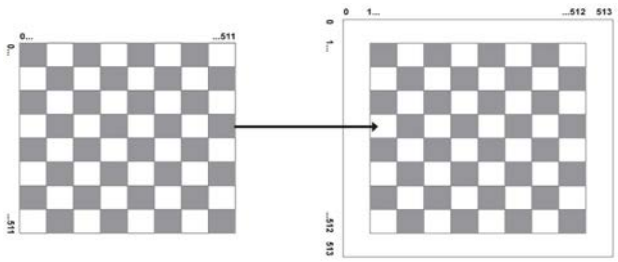
As an example consider the image of size 511×511.By boundary expansion we make it as an new image of size 513×513.This is done by copying the row and column of

the original image. This is described in steps below and is shown in Fig4.

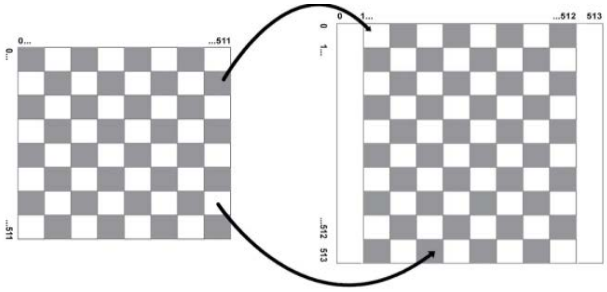
Step1: At first the original cover image is portioned into shadow and blank pixels as in Fig 5.a. Then we copy the original image into blank space of size 513×513 where the original image starts at row 1 and column 1 and ends at row 512 and column 512 of blank space

Step2: In second step we copy the pixels in the second row and pixels in the 512th row into first and last row of new image respectively where the image starts at row1 and column 1 and ends at row 512 and column 512 of blank space as in Fig 5.b

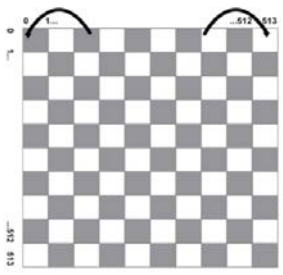
Step3: In third step we copy the pixels in 2nd and 511th column of new image into 0th and 513th column of new image respectively as in Fig 5.c



(a) Step1:Copying of original image of size 511×511 into blank space of size 513×513



(b) Row-wise expansion



(c) column wise expansion

Fig 5: Boundary Expansion Method

Now the image has size 513×513 and we can start the embedding process as in Rhombus prediction where the shadow pixels are scanned from left to right and from top to bottom including the pixels located in borders. For blank pixel embedding we have to repeat the process. This is done because during shadow pixel embedding some of the pixels may change. So for processing blank pixels we have to remove the border pixels of the new expanded image and then repeat the steps above. Here the original image that is used in first step is new expanded image after removing the border pixels. Here the boundary expansion process is done twice for each embedding. Fig 6 shows the full image after boundary expansion.

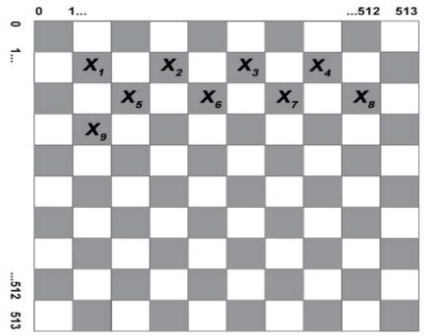


Fig 6: Shadow and blank pixel partition after boundary expansion

In extraction phase, we first remove the border pixels and then process the blank pixels, extract the embedded message and realize the image recovery in blank pixels. And then process the shadow pixels, extract the embedded message and realize the image recovery of shadow pixels. The processing of shadow pixels are done after processing of blank pixels.

C. *Data Embedding*

The proposed RDH method uses Rhombus Prediction and Double layered embedding. In Rhombus prediction the four neighbouring pixels surrounding the pixel in rhombus fashion is considered for prediction of cover pixel x_i. For this cover image is partitioned into shadow and blank pixels. And embedding is done in shadow and blank pixels in such a way that the processing of blank pixels is done after processing of shadow pixels.The four shadow pixels surrounding a pixel *xi* is denoted as v1,v2,v3,v4 which are used in rhombus prediction. The shadow pixels are scanned from left to right and top to bottom to derive the cover sequence $(x_1,...,x_N)$.For each pixel xi, a prediction value is computed and corresponding prediction value is denoted as $\hat{x}_i$. The prediction is computed by equation 4 given below.

$$\hat{x}i = \frac{v1+v2+v3+v4}{4} \qquad (4)$$

The prediction value $\hat{x}_i$ is then subtracted from original pixel *xi* to get prediction error e_i .

D. Location map compression

In some reversible data hiding techniques the embedded bit-stream mainly consists of two parts: one part that conveys the secret message and the other part that contains embedding information for blind detection, including the binary (overflow) location map. To increase embedding capacity, we have to make the size of the second part as small as possible. The necessary auxiliary information in proposed method include the thresholds (a_n, b_n), length of compressed location map (S_{CLM}), secret message length etc. Here, to avoid the overflow and underflow, the pixels valued 0 will be changed to 1, and the pixels valued 255 will be changed to 254. A location map will be established to record these problematic locations. The location map is a binary sequence sized N and it will be losslessly compressed to reduce its size. So when a pixel valued 0 and 255 is encountered it is recorded as 1 in corresponding locations of location map and rest of the location is recorded as 0.In implementation of proposed RDH, run length coding is used for lossless compression. The compressed location map and its size are denoted as CLM and SCLM, respectively. Run-length Encoding (RLE) is a very simple form of data compression in which runs of data that is, sequences in which the same data value occurs in many consecutive data elements are stored as a single data value and count, rather than as the original run

E. Extraction and Image recovery

Although in embedding phase shadow pixels are processed first, in extraction phase, we first extract the embedded message and realize image recovery for blank pixels, and then, extract the embedded message and realize image recovery for shadow pixels. The two layers embedding are processed similarly, in proposed RDH.

IV. IMPLEMENTATION

The proposed scheme of reversible data hiding is coded using VHDL and is synthesized using Xilinx ISE. MATLAB is used here to generate text pixels from cover image, to view the final embedded and extracted image. Here Xilinx XC6SLX16 and MATLABR2014a version is used. The proposed system consists of two stages: embedding and extraction process. For both embedding and extraction process, FPGA read the image pixels as input which is generated using MATLAB. As a first step , the cover image is converted into text file before embedding process using MATLAB, so that image pixels can be read by FPGA and after embedding the output text file generated using Xilinx is converted back to image and is viewed in MATLAB. Data embedding and extraction algorithm is given as below.

A. Data Embedding Algorithm

The steps for data embedding algorithm is as follows:

1. The cover image is first converted to text using MATLAB
2. Read image pixels from text file in FPGA. Read values one by one and store in array (x_txt)
3. Convert the image to matrix form i.e. convert the array to two dimensional array
4. Create Location map and compress location map using run length coding
5. Find Prediction error, $e_{i} = x_i - x_{i\wedge}$.
6. Embed secret message into shadow pixels (Rhombus method)
7. Record LSB of last S_{aux} pixels to obtain a binary sequence S_{LSB} and embed this sequence to blank pixels
8. By using LSB replacement, embed the auxiliary information into first S_{aux} pixels to generate the marked image
9. Save the image as text file. The text file is then converted to image using MATLAB.

TABLE 1 Comparison of embedding capacity of proposed method with and without boundary expansion method.

Image	Embedding Capacity(in bits)	Modified Embedding capacity(in bits)
Lena	43,702	44,069
Peppers	37,325	37,463
Baboon	20,461	20,542
Barbara	38,191	38,517

B. Data Extraction Algorithm

The steps for data extraction algorithm are as follows:

1. Read image pixels from text file in FPGA.
2. Determine the auxiliary information by reading LSBs of the first S_{aux} pixels.
3. Extract the S_{LSB} (original LSBs) from blank pixels(realize the restoration for these pixels) and then replace LSBs of first S_{aux} pixels by S_{LSB}
4. Extract the embedded secret message from shadow pixels(realize restoration for these pixels) .The pixels with i>Nend can be restored as

themselves since they are unmodified in data embedding

5. Image is saved as text file
6. The text file is converted to image using MATLAB.

TABLE 2
Embedding capacity and PSNR values for different values of (a,b)

Parameter(a,b)	Lena		Peppers		Barbara	
	Embedding Capacity(bits)	PSNR (db)	Embedding Capacity(bits)	PSNR (db)	Embedding Capacity(bits)	PSNR (db)
(0,1)	69301	75.0327	40773	69.1231	98320	73.7068
(-1,0)	65493	75.2833	40734	69.1987	47072	73.5714
(-2,2)	44069	76.1148	37463	69.6024	38517	73.6652
(-3,2)	38578	76.2739	35621	69.6421	33752	74.0232

V. RESULT AND DISCUSSION

The top level entity of proposed approach is modelled and synthesized using the Xilinx ISE 14.1. The device family used is Spartan6 (Device: XC6SLX16) with a speed grade of -3. The final extracted and embedded output after simulation is viewed in MATLAB. Fig 7 shows original image, embedded image and extracted image.

(a) (b)

(c)

Fig 7: (a) Original image (b) Embedded image and (c) Extracted image

Several experiments are conducted to demonstrate the performance of the proposed method. Four standard 512×512 sized gray-scale images including Lena, Peppers, Baboon, and Barbara. TABLE 1 shows the comparison of embedding capacity with and without Boundary expansion. From this table we can see that embedding capacity has improved after boundary expansion.

TABLE 2 shows embedding capacity and PSNR values for proposed RDH for different thresholds (*a*,*b*). The image Lena, Peppers and Barbara is tested for different values of (*a*,*b*) given by (0,1), (-1,0), (-2,2), and (-3,2) . We can select the bins adaptively such that $a < b$. From the table 2 we can see that PSNR value increases and embedding capacity decreases as the distance between bins *a* and *b* increases. This is because as the distance between bins increases, the number of pixels that is unchanged during embedding process increases and thus distortion decreases. For example consider bins (-2,2), here bins greater than 2 and less than -2 is shifted whereas between the bins -2 and 2 the pixels are unchanged. In the case of bins (0,1), the bins greater than 1 and less than 0 are shifted whereas between bins 0 and 1 the pixels are unchanged. So compared to (0,1) embedding at bins (-2,2) provides less distortion . Hence when embedding capacity increases image quality decreases, where image quality is measured by PSNR. So there should be compromise between embedding capacity and image quality.

The PSNR values depend on the number of pixels modified during embedding process. Greater the PSNR value greater will be the image quality. PSNR value is not reduced and thus good quality stego image is obtained for different images. The original image is completely recovered back, after the hidden data is extracted. So proposed method is an efficient reversible data hiding which provides high quality stego image.

VI. CONCLUSION

In this paper secure and efficient Reversible Data Hiding method is proposed based on modification of prediction error histogram. Here prediction error is computed for each pixel. The bins of the histogram are adaptively

selected. By boundary expansion embedding capacity is improved. To ensure security the secret data is encrypted. The proposed method is tested for different images and shows improvement in embedding capacity. Also different image is tested by varying the values of bins. The proposed method provides a good quality stego image with better PSNR values. The original image and secret data is completely recovered during extraction. In future more effective reversibility data hiding can be achieved by using better prediction that exploits correlation of larger neighborhood.

REFERENCES

[1] Awrengjeb.M "Overview of Reversible Data Hiding", *International Conference on Computer and Information Technology, pp 75-79,2003*

[2] C.T. Johnston, K. T. Gribbon, D. G. Bailey "Implementing Image Processing Algorithms on FPGAs" *Institute of Information Sciences & Technology, Massey University)*

[3] D.M.Thodi and J.J.Rodriguez, "Expansion embedding techniques for reversible watermarking", *IEEE Trans. Image Process., vol. 16, no. 3, pp. 721–730,2007*

[4] Guangyong Gao, CaixueZhou, and ZongminCu "Reversible Watermarking Using Prediction-Error Expansion and Extreme Learning Machine", *Mathematical Problems in Engineering Volume , Article ID 670535,2017*

[5] Jun Tian "Reversible Data Embedding Using a Difference Expansion", *IEEE Transactions On Circuits And Systems For Video Technology, Vol. 13, No. 8,2003*

[6] Masoud Nosrati, Ronak Karimi, and Mehdi Hariri "Reversible Data Hiding: Principles, Techniques, and Recent Studies", *World Applied Programming, Vol (2), No (5), 2012*

[7] Mehdi Fallahpur Reversible image data hiding based on gradient adjusted prediction, *IEICE Electron. Exp., vol. 5, no. 20, pp. 870–876, Oct. 2008.*

[8] Osamah M. Al-Qershi, Khoo Bee Ee " An Overview of Reversible Data Hiding Schemes based on Difference Expansion Technique", *2009 First International Conference on Software Engineering & Computer Systems, ICSECS09,2009*

[9] Rakesh Karmakar, Rajib Sarkar "Implementation of Run Length Encoding on FPGA Spartan 3E" *International Journal of Computer Science and technology Vol. 5, SPl – 1,2014*

[10] S.-K.Lee, Y.-H.Suh, and Y.-S.Ho "Reversible image authentication based on watermarking", *in Proc. IEEE ICME, pp. 1321–1324 Jul. 2006*

[11] W. Zeng "Digital watermarking and data hiding: technologies and applications,"*in Proc. Int. Conf. Inf. Syst., Anal. Synth., vol. 3, pp. 223–229,1998*

[12] Xiaolong Li, Weiming Zhang, Xinlu Gui, and Bin Yang "A Novel Reversible Data Hiding Scheme Based on Two-Dimensional Difference-Histogram Modification" *IEEE transactions on information forensics and security,Vol.8,No.7, 2013*

[13] Xiaolong Li, Weiming Zhang, Xinlu Gui, and Bin Yang "Efficient Reversible Data Hiding Based on Multiple Histograms Modification", *IEEE Transactions On Information Forensics And Security, Vol. 10, No. 9, 2015*

[14] Yongjian Hu, Heung-Kyu Lee, and Jianwei Li " DE-Based Reversible Data Hiding With Improved Overflow Location Map", *IEEE Transactions On Circuits And Systems For Video Technology, Vol. 19, No. 2,2009*

[15] Zhicheng Ni, Yun-Qing Shi, Nirwan Ansari, and Wei Su"Reversible Data Hiding" , *IEEE Transactions on Circuits and Systems for Video Technology ,2006*