Defence Science Journal, Vol. 73, No. 6, November 2023, pp. 688-698, DOI : 10.14429/dsj.73.5.18991

A Multistage High-Capacity Reversible Data Hiding Technique Without Overhead Communication

Sanjay Kumar[#,*], Gurjit Singh Walia[#] and Anjana Gupta[$]

[#]*DRDO-Scientific Analysis Group, Delhi-110054, India*

[$]*Department of Applied Mathematics, Delhi Technological University, Delhi-110042, India*

[*]*E-mail:* skparsade@gmail.com

ABSTRACT

Reversible Data Hiding(RDH) has been extensively investigated, recently, due to its numerous applications in the field of defence, medical, law enforcement and image authentication. However, most of RDH techniques suffer from low secret data hiding capacity and communication overhead. For this, multistage high-capacity reversible data hiding technique without overhead is proposed in this manuscript. Proposed reversible data hiding approach exploits histogram peaks for embedding the secret data along with overhead bits both in plain and encrypted domain. First, marked image is obtained by embedding secret data in the plain domain which is further processed using affine cipher maintaining correlation among the pixels. In second stage, overhead bits are embedded in the encrypted marked image. High embedding capacity is achieved through exploiting histogram peak for embedding multiple bits of secret data. Proposed approach is experimentally validated on different datasets and results are compared with the state-of-the-art techniques over different images.

Keywords: Data hiding; Reversible data hiding; Histogram shifting; Encryption; Decryption

NOMENCLATURE

RDH	:	Reversible data hiding
PSNR	:	Peak signal-to-noise ratio
dB	:	Decibles
I	:	Grayscale Image
P_i	:	Peak point
Z_i	:	Zero point
b_i	:	Binary secret message sequence
n_i	:	Chunks of n-bits
d_i	:	Decimal number sequence

1. INTRODUCTION

Aim of reversible data hiding is securely transferring the secret information over the communication channel. This is obtained by hiding of secret information into the cover media and, at the receiver end, secret data extracted along with faithful recovery of cover. RDH is broadly used in defense, stereo image coding, medical, law enforcement, image authentication and etc., whereas, distortion in the original cover is not tolerable for this scenario. Up to now, many RDH techniques have been explored by the various researchers in different domain. RDH methods mainly focus on obtaining the high data hiding capacity as well as faithful recovery of cover. Recent review of RDH methods enlisting merits and demerits were discussing by Sanjay[1], *et al.*

Generally, RDH work can be divided mainly into plain[1-2,18-19] and encrypted domain[1,6-9]. Plain domain techniques of RDH emphasize on developing methods for enhancing the data hiding capacity along with tuneable quality of cover media. Plain domain RDH techniques are mainly classified as lossless compression[2-4,17], correlation expansion[18-20], histogram shifting[21,28,34-35] and interpolation[13-16]. In lossless compression, cover media is compressed using compression technique for creating the free space for data hiding. Embedding capacity of these methods are very less due to low compression and mainly used for authentication. In correlation expansion, pixels correlation of cover media is used to conceal the secret information, whereas, peak of histogram bin is used to hide secret information in histogram shifting. In interpolation techniques, secret information is embedded into the interpolated image whereas the original pixels of the image are intact.

On the other hand, reversible data hiding is also broadly investigated for encrypted domain due to its potential application ensuring the confidentiality, integrity, and authentication. Confidentiality keeps the cover and hidden data secret to prevent any attack, integrity maintains the consistency and accuracy of the data, whereas authenticity gives data access to only authorised individuals.

RDH in encrypted domain involved three parties: content owner, data hider and receiver. Content owner encrypts the original cover and sent it to the data hider. Data hider has no permission to access the original cover but embeds data into encrypted cover. Receiver can extract the secret information along with recovery of the original image. Most of the previous RDH-EI works do not incorporate multi-stage embedding and can not utilise the significant embedding scope in the plain

Received : 24 March 2023, Revised : 18 July 2023
Accepted : 29 July 2023, Online published : 03 November 2023

domain. PSNR does not play a significant role in the encrypted image. Encryption and data embedding are completely reversible steps and images are retrieved completely. Multi-stage embedding can lead to improved embedding capacity for the same image size. It ensures access of information at different level to different people. Thus, the same image can be used to propagate different messages to different sources, without the other message being accessed by the other party.

However, in past two decades, RDH work has been broadly investigated among researchers. Hence, the aim of this paper is to improve the data hiding capacity without sending any auxiliary information to the receiver. We propose a multi-stage reversible data hiding approach based on histogram shifting. In this, peak of histogram of the cover is first determined. Then, histogram is shifted according to the embedding bits. Histogram shifting creates overflow problem and to control this problem, we record pixels with their indices. Further, secret information is embedded into the plain domain generate the marked cover.

This marked cover is further encrypted using the specific algorithm, which maintains the correlation among the encrypted pixels. Peaks of encrypted marked image are used to hide communication overhead information of plain and encrypted domain. This communication overhead information helps the receiver to extract the secret information and to ensure the recovery of cover image. In the proposed technique, we embed two types of data in different domain. Secret information is embedded into the plain domain, whereas communication overhead information of both is embedded in encrypted domain. Information of encrypted domain helps the receiver to extract the secret information from the plain domain. Thrusts of proposed technique are given as follows:

- We proposed a novel multi-stage RDH technique incorporating both plain and encrypted domain based on histogram shifting.
- Flag based technique is presented for identification overhead of both domain.
- Proposes technique does not required any overhead communication through a separate channel.

The remaining paper is organised as following: In Section 2, we briefly reviewed some preliminary work related to proposed method. Proposed method discussed in Section 3. In Section 4, we discussed experimental results and analysis involving various embedding bits for smooth and texture images. The comparision of the proposed technique with state-of-the-art techniques is disscussed in Section 5. In Section 6, Concluding remarks and future direction are sketched.

2. RELATED WORK

RDH work can be classified into lossless compression, histogram shifting, correlation expansion and interpolation. Detailed explanations were discussed in the review paper[1,40]. However, the RDH work closely related to proposed method includes histogram shifting. Recent developments in this direction are as follows:

RDH work based on histogram shifting is briefly reviewed. First histogram-based RDH method has been proposed by Ni[21] et al. in 2006. In this method, peak value was chosen to embed the secret data. If the secret message bit one was encountered then histogram shifted one place right hand side, otherwise histogram intact. Total embedding bits were equal to the number of peak values along with PSNR 48.13 dB. This method suffers from the issues of multiple zero points which required the storage for the coordinate of zero points. Embedding capacity and PSNR of Ni[21], *et al.* method was improved by Fallahpour[33], *et al.* using block-wise histogram shifting. In this, cover image divided into blocks and histogram was generated for each block. Then, peak P_i and zero Z_i points were found out for each block. For pair (P_i, Z_i) of each block, if $P_i > Z_i$, then value of each pixel between the interval $[P_i, Z_i+1]$ were reduced by one, and then, pixels of P_i -1 were increased by one, if secret data bit one was encountered, otherwise no changed required. If, $Z_i > P_i$, then every pixel value lies between $[P_i+1, Z_i+1]$ were increased by one.

Then, block was again scan and pixels with value of P_i were increased by one if the secret data bit was encountered one, otherwise no change required. Data hiding capacity were approximately 154 per cent and 46 per cent more than[21] for lena and baboon greyscale image respectively. Zero and peak points were overhead information for this method, which is used at the end of receiver side for data extraction. Furthermore, Lee[27], *et al.*, also improved method[21] by using histogram of difference image. In this, difference $D(i, j)$ of cover grayscale image $I(i, j)$ was determine such as $D(i, j) = I(i,2j+1)-I(i, 2j)$. For difference value $D(i, j)\geq 2$, pixel $I(i, 2j+1)$ were increased by one and, for difference value $D(i, j)\leq -2$, pixels $I(i, j)$ were subtracted by one and other pixels were remain same. Then this modified image was used for secret data embedding. For modified difference value -1 or 1, if the secret data bit 1 was encountered, then these difference value changed −1 to −2 and 1 to 2 for odd pixels. If secret embedding bit zero was encountered, then all pixel skipped of difference image until -1 or 1 encountered. These methods have limited embedding capacity.

In order to improve embedding capacity, histogram modification[29] method based on differences of adjacent pixels was proposed. These differences were used to achieve large embedding capacity along with low distortion. Histogram shifting is also used to prevent overflow and underflow problem and binary tree structure solved multiple pairs of peak issue[21,33]. In this sequence, multiple histogram modification method was proposed by B. Ou[38], *et al.*, along with multiple pairs of bins to achieve high data hiding capacity. Data hiding rate and PSNR were better than the proposed methods[35-36]. Further, modulus operator[37] based RDH was introduced and cover image was partitioned into blocks. Each block was modified to improve the peak points of every block of histogram and these peak points were used for data hiding. Data hiding capacity and PSNR of this method was better than the state-of-the art methods[21,28,33-35]. Prediction error-based histogram method for RDH was introduced[42], in which cover image was divided into disjoint blocks.

Further, median was determined for each block and pixels changed as positive and negative directions from the median according to the data hiding bit. Original pixels were modified to overcome the overflow/underflow problem by using decomposed location message approach. In this sequence, J. He[39], *et al.* presented RDH based negative influence models

for JPEG image. Weighting factor for each frequency was determine along with negative index. Secret data was embedded into small negative indices and weighting factor adjusted to overcome the distortion.

In sum, most of method either suffered from capacity or from overhead issues. Hence to solve this, multistage RDH technique without overhead is proposed. The details of the Proposed method follow in the next section.

3. CORE DESIGN OF MULTISTAGE RDH

Proposed Multi Stage reversible data hiding technique design an effective technique to enhance data hiding capacity. The aim of the proposed method is communicating the secret message without sharing the auxiliary/overhead information to the receiver. For this, multistage data embedding approach is used. General framework of proposed approach is shown in Fig. 1 and details of data embedding, extraction and recovery of cover image are given in following subsections.

3.1 Data Embedding

Data embedding is performed by utilising the peak of plain and encrypted image. In this method, cover is converted into pixels and then, histogram of cover is generated. Then, original cover image is scanned to determine maximum point P_1 i.e pixel value with maximum occurrence and minimum point Z_1 means such pixels does not exist in the image or has minimum number of occurrences. Then, row and column indices values of pixels lies between Z_1 and $Z_1+(2^n-1)$ are stored with pixel value as a overhead O_1 of original cover image or increase values with (2^n-1) places until it become greater than 255. Further, pixels lying between (P_1, Z_1) are shifted upto (2^n-1) places. After that, binary sequence say $b_1, b_2, b_3,\ldots, b_n$ of secret message is divided into number of k chunks of n-bits say $n_1, n_2, n_3,\ldots, n_k$, and these chunks n_i, i=1 to k are separately converted into decimal number sequence $d_1, d_2, d_3,\ldots, d_k$.

Original cover image is again scanned and whenever peak P_1 encountered, then peak value P_1 is incremented by d_i and marked cover image generated. Further, marked cover image is encrypted using affine cipher so that correlation of pixels also maintained in the encrypted image. Encrypted image is scanned to determine maximum point P_2 and zero-point Z_2. Pixels lying between lies (P_2, Z_2) are one place shifted. Then,

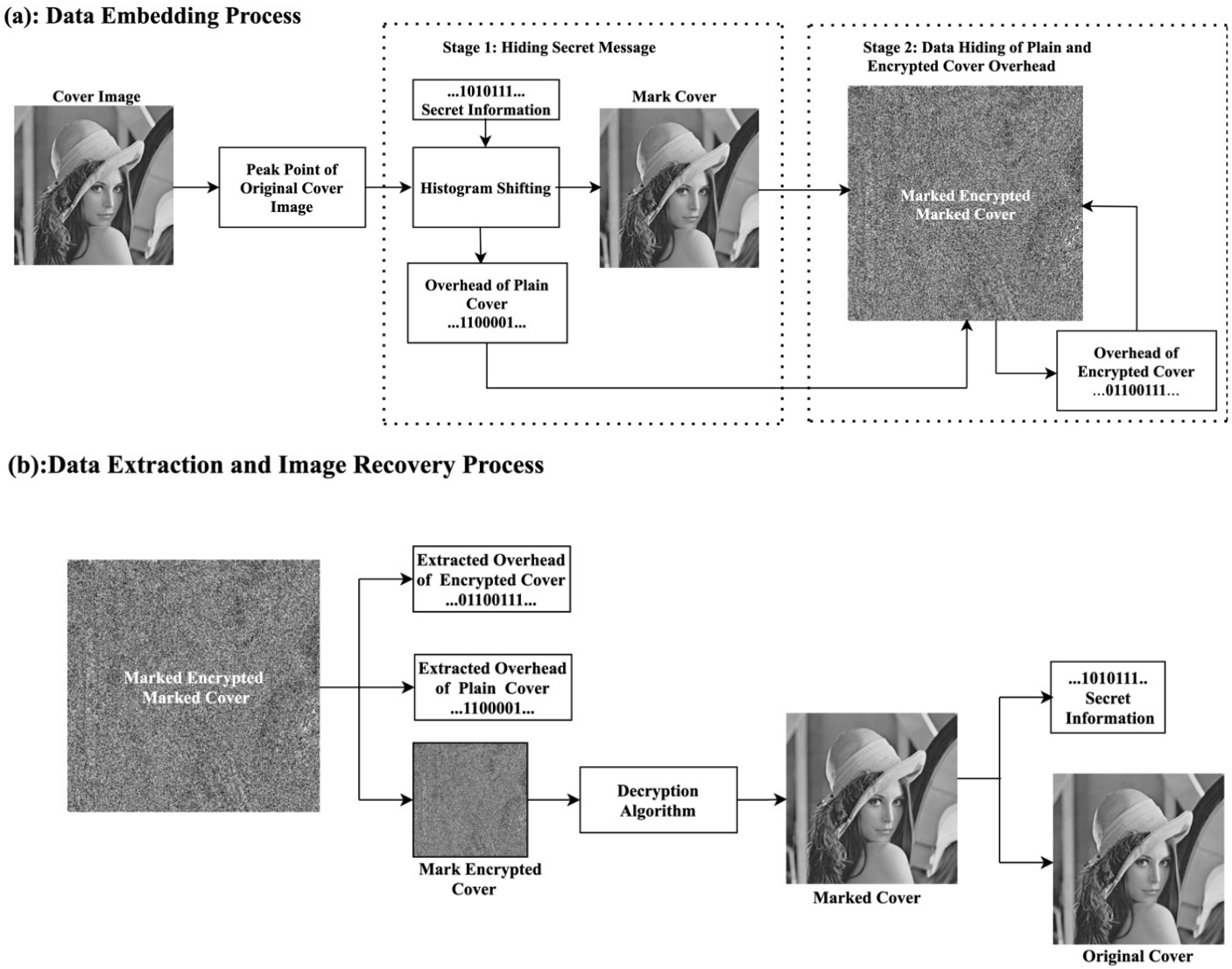

Figure 1. (a) Describes embedding process of secret information in plain image and overhead information in encrypted image and (b) Shows secret information extraction process along with overhead information and recovery of original image.

peak of encrypted image is used to embed overhead data O_1 of original image and O_2 of encrypted image. Overhead data O_1 is embedded in encrypted image. If peak P_2 has sufficient number of counts to embed O_2. Then, first embed O_2 data, flag value, zero point along with their indices. If peak is not sufficient then, next peak P_3 and zero-point Z_3 is calculated and embed remaining overhead in the similar manner. This process repeat until all overhead information is embedded.

3.2 Data Extraction and Image Recovery

In data extraction and image recovery process, receiver receives marked encrypted image from sender. This image is consisted of secret message and overhead information of plain and encrypted images. Next, all peaks are determined in descending order in-terms of frequency. For each peak P_i, whenever, peak pixels P_i and P_{i+1} are occurred, then secret message bit 0 and 1 are extracted respectively. If extracted data contains flag value then, first expected peak is found. If extracted data is consisted only one flag value, then, first byte gives next peak P_2 pixel value and second bytes gives Z_2 and from 17th bit upto flag value indicated overflow O_1, whereas 8-bits after flag value represent peak value P_1 and next 8-bits represent minimum point Z_1 and from 17th till the flag gives indices of minimum point. With the help of peak values and overhead information, secret information is extracted along with original cover image without any error. Figure 2 and Fig. 3 shows flowchart of data embedding and extraction.

3.3 Pseudo Code of Proposed Method

3.3.1 Pseudo Code for Data Embedding

- Input the grayscale image (I), number of bits to be embedded per peak pixel(n), binary data to be embedded (D)
- Generate histogram (H) for I
- Traverse I to find the maximum point (grayscale value with maximum frequency) peak (P_1) and minimum point (value with zero or minimum frequency) Z_1.
- Assumption $P_1 < Z_1$ if Z_1 found, else Z_1 =256.
- Traverse image I to store the plain overhead O_1, by saving the row and column indices of pixels whose value lie between [Z_1, Z_1 + (2^n-1)] or increasing with (2^n-1) will become >256.
- Traverse I and shift pixels with value in (P_1, Z_1) with (2^n-1).
- Traverse I and loop through D to get the next n bits that is to be inserted (say K)
  (a) Convert K to its decimal equivalent
  (b) If pixel value is equal to P_1, increment pixel value with K
  (c) continue traversing and move to the next n bits.
- Apply affine cipher on marked image I and generated encrypted image I
- Traverse Ito find peak (P_2) and zero point (Z_2).
- Assumption $P_2 < Z_2$, if Z_2 found, else Z_2=256 Traverse I' to store the encrypted overhead O_2, by saving the row and column indices of pixels whose value lie between (Z_2, Z_2+1)Z_2 or on increasing with 1 will become > 255.
- Traverse I' and shift pixels with value in (P_2, Z_2) with 1.

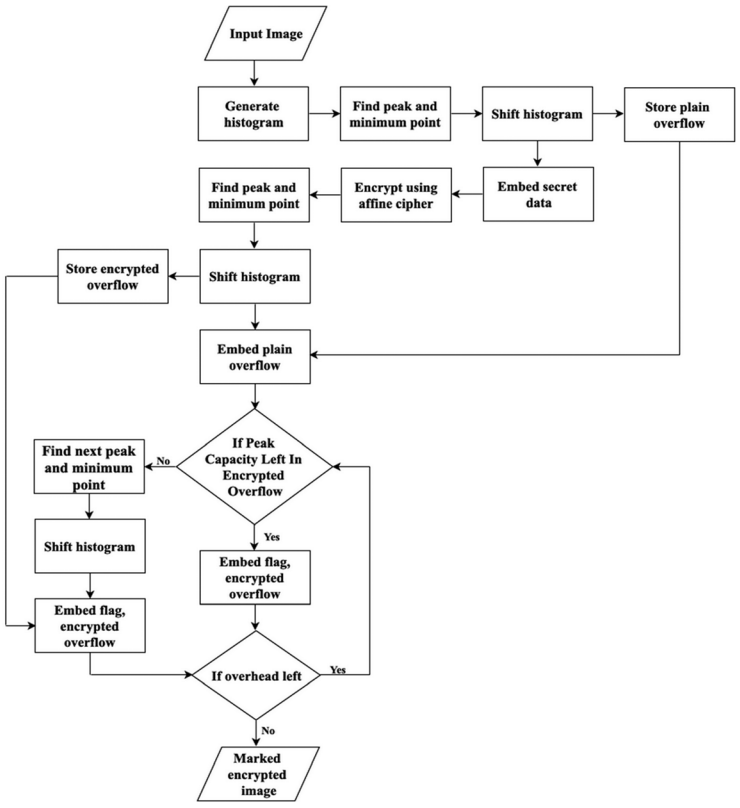

Figure 2. Flow chart for data embedding.

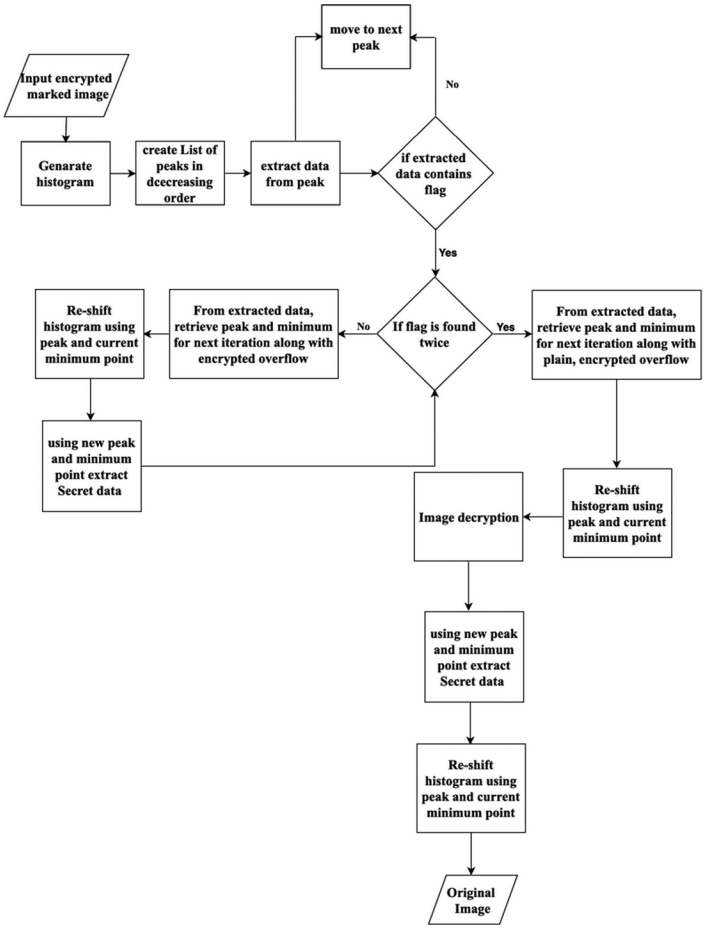

Figure 3. Flow chart for data extraction and image recovery.

- Traverse I' and loop through O_1 to get the next bit (b) that is to be inserted:
  (a) If pixel value is equal to P_2, add b
  (b) continue traversing and move to the next bit.
- After embedding O_1 (let the image be I" if the remaining count of P_2 is sufficient enough to embed O_2, then
  (a) From the peak P_2 pixel where no data is embedded, firstly embed the flag, then O_2, then again, the flag and value of Z_2
- If, P_2 is entirely used, then
  (a) Traverse I" to find the peak P_3 and zero point Z_3
  (b) Assumption $P_3 < Z_3$, if Z_3 found, else Z_3 =256
  (c) Traverse I" to store the encrypted overhead (O_3), by saving the row and column indices of pixels whose value lie between [Z_3, Z_3 +1] or increasing with 1 will become > 255.
  (d) Traverse I and shift pixels with value in (P_3, Z_3) with 1
  (e) Traverse I" and loop through O_3 to get the next bit (b) that is to be inserted
  - If pixel value is equal to P_3, add bit (b) and continue traversing and move to the next bit.
  - After O_3 is embedded, embed the flag and value of Z_3 at peak P_3.

3.3.2 Pseudo Code for Data Extraction and Image recovery

- Let the received image be I
- Generate all peaks of I in decreasing order of frequency
- For each peak P_i, Traverse the data embedded image(I), if pixel value is in range [P_i, P_i +1] the data extracted will be 0 if P_i else, 1 if P_i +1
- Extract data for each peak, if extracted data contains the flag. Then first peak where flag is found, i.e the expected peak P_1.
- Traverse I, if pixel value is in range [P_i +1, Z_1 +1], decrement the value with 1 to get the image I'.
- If the extracted data has one flag, then
  (a) First 8-bits give the peak P_2, next 8 has the minimum point Z_2, then from 17th bit till the flag gives us the encrypted overflow O_1 and 8 bits after flag gives minimum point Z_1

  (b) Traverse the image I', if pixel value is in range (P_2, P_2 + 1), the data extracted will be the difference between pixel value and P_2. Extracted data has first 8 bits of the peak P_3, next 8 has the minimum point Z_3, then from 17th bit till the flag gives us the plain overflow O_2 to (a)

  (c) Traverse I', if pixel value is in range (P_2+1, Z_2 +1), decrement the value with 1.
  (d) Replace the overhead indices stored in O_1 in the image I'.
- If the extracted data has two flags, then First 8 bits give the peak P_3, next 8 has the minimum point Z_3, then from 17th-bit till the flag gives us the plain overflow O_2 and bits between the two flags is the encrypted overflow O_1, 8-bits after flag gives minimum point Z_1
- Decrypt the Image I' to I"
- Traverse the image I", if the pixel value is in range [P_3, P_3+(2^n+1)], the data extracted will be the binary equivalent of the difference between pixel value and P_3

Table 1. Data Embedding for 1, 2 and 3- bits and corresponding overheads for smooth images without blocks

Image (512x512)	For 1, 2 and 3-bits chunks embedding in plain images									Encrypted image		
	EC_1	PR_1	O_1	EC_2	PR_2	O_2	EC_3	PR_3	O_3	O'_1	$O"_2$	$O"_3$
Lena	2711	54.02	16	5422	44.48	16	3133	37.12	16	80	80	96
Baboon	2772	50.45	16	5544	40.90	16	8316	33.50	16	80	80	96
Boat	5796	52.09	16	11592	42.54	94	17388	35.19	458	106	236	1058

Table 2. Data embedding for 1, 2 and 3- bits and corresponding overheads for smooth images with 128x128 blocks

Image (512x 512)	For 1, 2 and 3-bits chunks embedding in plain images									Encrypted image		
	EC_1	PR_1	O_1	EC_2	PR_2	O_2	EC_3	PR_3	O_3	O'_1	$O"_2$	$O"_3$
Lena	4774	50.29	64	9548	40.75	64	14322	33.39	64	384	384	384
Baboon	2961	52.43	64	5922	42.89	64	8883	35.54	64	384	384	1372
Boat	6067	52.01	64	12134	42.47	142	18201	35.12	506	384	592	1320

Table 3. Data embedding for 1,2 and 3- bits and corresponding overheads for amooth images with 64×64 block size

Image (512x512)	For 1, 2 and 3-bits embedding in plain images									Encrypted image		
	EC_1	PR_1	O_1	EC_2	PR_2	O_2	EC_3	PR_3	O_3	O'_1	$O"_2$	$O"_3$
Lena	6895	50.87	256	13790	41.34	256	20685	33.99	256	1280	1280	1280
Baboon	3985	50.77	256	7970	41.24	256	11955	33.89	256	1280	1280	1280
Boat	7951	50.67	334	15902	41.14	334	23853	33.79	672	1358	1358	1696

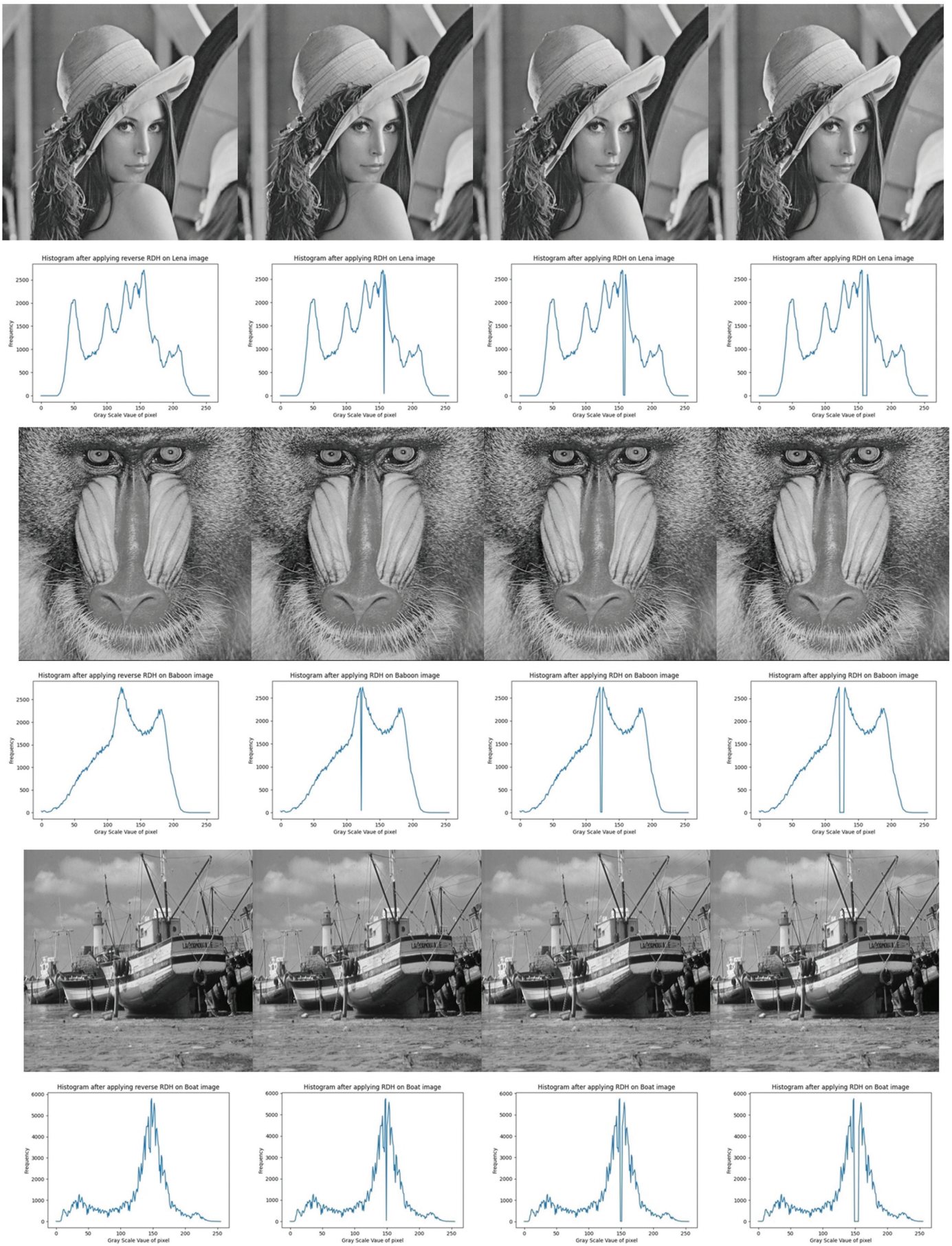

Figure 4. Test image(512x521) set: Lena, baboon and boat with original, 1-bit embedding, 2-bits embedding and 3-bits embedding and corresponding histogram.

Table 4. One-bit chunks embedding

Image (512x 512)	Plain image				Encrypted image		
	Peak value	Pure EC	Overhead	PSNR (dB)	Peak	#Peak count	Overhead
1.1.01	152	2031	16	52.33	26	1986	16
1.1.02	184	3388	16	54.13	11	3305	16
1.1.03	192	3249	16	55.34	11	3238	16
1.1.04	204	4358	16	56.25	108	4292	16
1.1.05	144	4171	16	50.93	157	4118	16
1.1.06	83	3387	16	48.74	8	3304	16
1.1.07	176	4998	16	54.60	13	4936	16
1.1.08	160	8087	16	50.73	99	8004	16
1.1.09	192	6998	16	52.48	67	6915	16
1.1.10	152	3238	16	50.50	43	3155	16
1.1.11	0	13312	16	48.35	3	13229	16
1.1.12	170	4944	16	52.27	26	4925	16
1.1.13	112	2563	16	51.31	118	2518	16
1.2.02	190	3388	16	54.73	181	3305	16
1.2.03	206	3249	16	56.07	196	3238	16
1.2.04	217	4358	16	56.66	248	4292	16
1.2.05	120	4171	16	51.57	30	4118	16
1.2.06	33	3387	16	48.93	42	3304	16
1.2.07	199	4998	16	55.66	43	4936	16
1.2.08	115	8087	16	50.95	232	8004	16
1.2.09	161	6998	16	52.77	170	6915	16
1.2.10	106	3238	16	50.67	105	3155	16
1.2.11	0	13742	16	48.56	3	13659	16
1.2.12	156	4944	16	53.01	82	4925	16
1.2.13	133	2563	16	52.35	243	2518	16

and extracted data is the secret message.

- Traverse I" , if pixel value is in range $[P_3+(2^n+1), Z_3+(2^n+1)]$
- Replace the overhead indices stored in plain overhead O_2 to get the original image decrement the value with (2^n+1).

Proposed method is validated in different grayscale image. Details of experimental results are as follows in turn.

4. EXPERIMENTAL VALIDATION

Experimental validation is performed over different dataset and results are compared with existing RDH methods. To gauge robustness of the proposed method, results are obtained over both smooth and texture images and significant work is deduced through extensive analysis. Details of experimental validation are follows in turn.

4.1 Experimental Set-up

In this subsection, various observations are conducted to illustrate the performance of the proposed technique. We did experiments on smooth as well as texture grey-scale image and found that texture images give the sufficient zero pixels to overcome the overflow problem. It significantly reduces the overheads in comparison to the smooth images. Different dataset are used for smooth and texture images. We considered different block size for data hiding with different data bits to enhance the embedding capacity and corresponding overflow has been observed. Secret message bit sequence is generated using pseudo random number generator.

Proposed technique is implemented in python programming language and tested on MacBook air with following configuration: Apple M1 chip with 8 core, RAM 8GB, and Python 3.7. Results are summarized in the following subsection in tabular form.

4.2 Results and Analysis for Smooth Images

In Table 1, we have shown the embedding capacity, PSNR and over heads corresponding to 1,2 and 3 bits.

Here, plain/encrypted domain overhead for one bit embedding is same for lena, baboon and boat image while overhead is same for lena and baboon but different for boat image. For two and three bits embedding, only boat image

Table 5. Two-bit chunks embedding

Image (512x 512)	Plain image				Encrypted image		
	Peak value	Pure EC	Overhead	PSNR (dB)	Peak	#Peak count	Overhead
1.1.01	152	4062	16	42.79	168	1986	16
1.1.02	184	6776	16	44.59	11	3321	16
1.1.03	192	6498	16	45.80	11	3238	16
1.1.04	204	8716	16	46.71	250	4292	16
1.1.05	144	8342	16	41.39	43	4118	16
1.1.06	83	6774	16	39.20	8	3320	16
1.1.07	176	9996	16	45.06	13	4936	16
1.1.08	160	16174	16	41.20	99	8020	16
1.1.09	192	13996	16	42.94	67	6931	16
1.1.10	152	6467	16	40.97	43	3171	16
1.1.11	0	26624	16	38.81	3	13245	16
1.1.12	170	9888	16	42.74	168	4925	16
1.1.13	112	5126	16	41.78	118	2518	16
1.2.04	217	8716	16	50.53	134	4292	16
1.2.05	120	8342	16	44.63	30	4118	16
1.2.06	33	6774	16	51.70	42	3320	16
1.2.07	199	9996	16	49.62	43	4936	16
1.2.08	115	16174	16	42.10	232	8020	16
1.2.09	161	13996	16	43.95	170	6931	16
1.2.10	106	6476	16	46.84	105	3171	16
1.2.11	0	27484	16	39.60	3	13659	16
1.2.12	156	9888	16	45.59	224	4925	16

overhead is different from the other two image. Average embedding capacities are 3760-bits, 7520-bits, 11280-bits along-with average PSNR 52.19 dB, 42.64 dB and 35.27 dB corresponding to one, two and three bits for lena, baboon and boat images

In Table 2, block-wise data embedding is done of block size 128 × 128 and embedding capacity, PSNR and overhead corresponding to one, two and three bits embedding.

Overhead for one bit embedding is same for all three images in plain image and encrypted image overhead is also same for all images. For two- and three-bits data embedding, overhead of boat image is more than the overhead of other two images in plain image as well as encrypted image. Average embedding capacity of all three image are 8601-bits, 9201-bits, 13802-bits corresponding to average PSNRs 51.57 dB, 42.03 dB, 34.68 dB for one, two and three data hiding bits respectively.

In Table 3, we divide greyscale texture image into the block of size 64 × 64 blocks. For this, PSNR of mark image corresponding to 1 and 2 bits embedding is on and average 50.77 dB and 41.24 dB respectively, while it lies between 33 dB to 34 dB for 3 bits embedding. Overhead for marked lena and baboon image is same for the fixed number of data bits, whereas it different for the boat image. Overhead for encrypted image is also observed same.

4.3 Results and Analysis for Texture Images

We have also conducted experimental results for different texture images along with different embedding bits and block size. Performance analysis of proposed approach in terms of data hiding capacity, overhead and PSNR are shown in Table 4, Table 5 and Table 6 corresponding to one, two and three embedding bits. Details of experimental results for texture image are given in the tabular form.

Table 4 describes the experimental results for one bit chunks embedding. Twenty-five texture image of size 512×512 are used for one bit embedding which shows the overhead of marked and encrypted marked image. Overhead of marked image and encrypted marked images are same.

Overheads of the plain and encrypted are embedded into the encrypted domain, while plain domain is used only for secret data embedding. Encrypted domain is only used for the overhead information of both domain and this overhead information are distinguish via flag value. Plain domain overhead information is embedded before the flag and encrypted domain overhead information is embedded after the flag value. With the help of these overhead information secret message is extracted along with original image without any error.

Table 6. Three-bit chunks embedding

Image (512x 512)	Plain image				Encrypted image		
	Peak value	Pure EC	Overhead	PSNR (dB)	Peak	Pure EC	Overhead
1.1.01	152	6093	16	35.43	196	1986	16
1.1.02	184	10164	16	37.23	11	3337	16
1.1.03	192	9747	16	38.44	11	3238	16
1.1.04	204	13074	16	39.36	151	4307	16
1.1.05	144	12513	16	34.03	243	4120	16
1.1.06	83	10161	16	31.85	8	3336	16
1.1.07	176	14994	16	37.70	211	4947	16
1.1.08	160	24261	16	33.83	99	8036	16
1.1.09	192	20994	16	35.58	67	6947	16
1.1.10	152	9714	16	33.60	43	3187	16
1.1.11	0	39936	16	31.45	3	13261	16
1.1.12	170	14832	16	35.38	168	4925	16
1.1.13	112	7689	16	34.41	118	2518	16
1.2.08	115	24261	16	38.72.10	232	8036	16
1.2.09	161	20994	16	44.17	170	6947	16

Table 7. Comparison table between state-of-the-art methods and our proposed method

Methods	Average embedding capacity for 512×512×grayscale image (bits)	Average PSNR(dB)
Honsinger[10], *et al.*	less than 1024	…
Fridrich[3], *et al.*	less than 1024	…
Vleeschouwer[5], *et al.*	less than 4096	less then 35 39.0
Zhang[7]	4457	55.0
Zheng[25], *et al.*	10747	52.3
Arhm[26], *et al.*	4718	44.03
Proposed method for block size of 128x128 for smooth images with three bits embedding	13802	34.68
Proposed method for block size of 64x64 for smooth images with	18831	33.89
Proposed method for three chunks bits of texture images	15962	36.07

5. COMPARISON WITH STATE-OF-THE-ART METHODS

In this section, comparison between state-of-the-art methods[3,5,10,25-26] and our proposed method has been carried out and results are depicted in Table 7. The results clearly shows that our method provided better data hiding capacity for the both smooth and texture images in comparison to the other RDH.

6. Conclusion

RDH techniques has been extensively investigated and explored for their potential application in different field. However, most of RDH techniques hides the overhead bits in the same domain of cover image which leads to reduction in embedding capacity. Proposed multistage RDH approach considered both plain and encrypted domain for secret data communication without any separate channel for overhead data. Propose approach exploits histogram peaks for embedding the secret data along with overhead bits both in plain and encrypted domain. For experiment, chunks of one, two- and three-bits secret message are only used to conceal secret information. Experimental results are performed on different grayscale images and shows significant improvement over state-of-the-art approaches. Embedding capacity for smooth and texture grayscale images corresponding to different chunks of bits with their overhead bits are analysed. On an average, proposed

approach achieves embedding capacity18831 bits for 64x64 block size of smooth image, whereas 15962 bits without block of image size 512x512 for texture images.

In future, proposed approach can further be extended through utilisation of multilevel embedding wherein multiple histogram peaks can be investigated for embedding secret data.

REFERENCES

1. Kumar, S.; Gupta, A. & Walia, G.S. Reversible data hiding: A contemporary survey of state-of-the-art, opportunities and challenges. *Applied Intell.,* 2021, **52**, 7373-7406. doi: 10.1007/s10489-021-02789-2
2. Barton, J.M. Method and apparatus for embedding authentication information within digital data. U.S. Patent, 1997, **5**, 646, 997.
3. Fridrich, J.; Goljan, M. & Du, R. Invertible authentication. Proc. SPIE, Security and Watermarking of Multimedia Contents, 2001, 4314, 197-208
4. Fridrich, J.; Goljan, M. & Du, R. Lossless data embedding-new paradigm in digital watermarking. *EURASIP J. Adv. Signal Process,* 2002, **2**, 185-196
5. Vleeschouwer, C. De; Delaigle, J.F. & Macq, B. Circular interpretation on histogram for reversible watermarking. in IEEE Int. Multimedia Signal Process. Workshop, France, Oct. 2001, pp. 345-350.
6. Zhang, X. Reversible data hiding in encrypted image. *IEEE Signal Process. Lett.*, 2011, **18**(4), 255-258. doi: 10.1109/LSP.2011.2114651
7. Zhang, X. Separable reversible data hiding in encrypted image. *IEEE Transact. Inf. Forensics and Security*, 2012, **7**(2), 826-832. doi:10.1109/TIFS.2011.2176120.
8. Ma, K.; Zhang, W.; Zhao, X.; Yu, N. & Li, F. Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Transact. Inf. Forensics and Security*, 2013, **8**(3), 553-562. doi: 10.1109/TIFS.2013.2248725
9. Zhang, W.; MA, K. & Yu, N. Reversibility improved data hiding in encrypted images. *Signal Process.*, 2013, **94**, 118-127. doi: 10.1016/j.sigpro.2013.06.023
10. Honsinger, C.W.; Jones, P.; Rabbani, M. & Stffel, J.C. Lossless recovery of an original image containing embedded data. U.S. Patent, 2001, **6**(79), 278.
11. Goljan, M.; Fridrich, J.J.; Du, R. Invertible authentication. Proceedings 4th Inf. HidingWorkshop, 2001, 4314, 197-208.
12. Xuan, G.; Zhu, J.; Chen, J.; Shi, Y.Q.; Ni, Z. & Su, W. Distortionless data hiding based on integer wavelet transform. *IEEE J. Electron. Lett.,* 2002, **38**, 1646-1648
13. Wahed, M.A. & Nyeem, H. Efficient LSB substitution for interpolation based reversible data hiding scheme. *In* 20th International Conference of Computer and Information Technology (ICCIT), 2017, 1-6. doi: 10.1109/IC-CITECHN.2017.8281771
14. Lu, T.C. Interpolation based hiding scheme using the modulus function and re-encoding strategy. *Signal Process.,* **142**, 244-259. doi: 10.1016/j.sigpro.2017.07.025
15. Jung, K.H. & Yoo, K.Y Data hiding method using image interpolation. *Comput. Standards and Interfaces,* 2009, **31**(2), 465-470. doi: 10.1016/j.csi.2008.06.001
16. Loua, D.C.; Choub, C.L.; Weia, H.Y. & Huang, H.F. Active steganalysis for interpolation-error based reversible data hiding. *Pattern Recognition Lett.,* 2013, **34**(9), 1032-1036. doi: 10.1016/j.patrec.2013.01.023
17. Celik, M.U.; Sharma, G.; Tekalp, A.M. & Saber, E. Lossless generalized-LSB data embedding. *IEEE Transact. Image Process.*, 2005, **14**, 253-266. doi: 10.1109/TIP.2004.840686
18. Tian, J. Reversible data embedding using a difference expansion. *IEEE Transact. Circuits and Syst. for Video Technol.,* 2003, **13**(8), 890-896. doi:10.1109/TCSVT.2003.815962
19. Alattar, A.M. Reversible watermark using the difference expansion of a generalised integer transform. *IEEE Transact. Image Processing*, 2004, **13**(8), 1147-1156. doi:10.1109/TIP.2004.828418
20. Thodi, D.M. & Rodriguez, J.J. Expansion embedding techniques for reversible watermarking. *IEEE Transact. Image Process.*, 2007, **16**(3), 721-730. doi: 10.1109/TIP.2006.891046
21. Ni, N.; Shi, Y.Q.; Ansari, N. & Su, W. Reversible data hiding. *IEEE Transact. Circuits and Syst. Video Technol.*, 2006, **16**(3), 354-362. doi: 10.1109/TCSVT.2006.869964
22. Fallahpour, M. & Sedaaghi, M.H. High capacity lossless data hiding based on histogram modification. *IEICE Electron. Express*, 2007, **4**(7), 205-210. doi: 10.1587/elex.4.205
23. Lin, C.C.; Tai, W.L. & Chang, C.C. Multilevel reversible data hiding based on histogram modification of difference images. Pattern Recognition **41**(12), 3582-3591. doi: 10.1016/j.patcog.2008.05.015
24. Sachnev, V.; Kim, H.J.; Nam, J.; Suresh, S. & Shi, Y.Q. Reversible watermarking algorithm using sorting and prediction. *IEEE Transact. Circuits and Syst. Video Technol.,* 2009, **19**(7), 989-999. doi:10.1109/TCSVT.2009.2020257
25. Zheng, S.; Li, D.; Hu, D.; Ye, D.; Wang, L. & Wang, J. Lossless data hiding algorithm for encrypted images with high capacity. *Multimedia Tools and Applications*, 2016, **75**, 13765-13778
26. Arhm, A.; Nugroha, H.A. & Adji,T.B. Multiple layer data hiding scheme based on difference expansion of quad. *Signal Process*, 2017, 137, 52-62.
27. Lee, S.; Suh, Y. & Ho, Y. Reversible Image authentication based on watermarking. *In* IEEE International Conference on Multimedia and Expo, 2006, 1321-1324. doi: 10.1109/ICME.2006.262782
28. Lin, C.C.; Tai, W.L. & Chang, C.C. Multilevel reversible data hiding based on histogram modification of difference images. *Pattern Recognition*, 2008, **41**(12), 3582-3591.

doi: 10.1016/j.patcog.2008.05.015
29. Tai W., Yeh C., Chang C. (2009) Reversible Data Hiding Based on Histogram Modification of Pixel Differences. IEEE Transactions on Circuits and Systems for Video Technology 19(6):906-910. doi: 10.1109/TCSVT.2009.2017409.
30. Gao, X.; An, L.; Yuan, Y.; Tao, D. & Li, X. Lossless data embedding using generalized statistical quantity histogram. *IEEE Transact. Circuits and Syst. Video Technol.*, 2011, **21**(8), 1061-1070. doi: 10.1109/TCSVT.2011.2130410
31. Goljan, M.; Fridrich, J.J & Du, R. Distortion-free data embedding for images. 4th International Workshop on Information Hiding LNCS, 2001, 2137, 27-41. doi: 10.1007/3-540-45496-93
32. Vleeschouwer, C.D.; Delaigle, J.E. & Macq, B. Circular interpretation of histogram for reversible watermarking. 2001 IEEE fourth workshop on multimedia signal processing (Cat. No.01TH8564), 2001, 345-350. doi:10.1109/MMSP.2001.962758
33. Fallahpour, M. & Sedaaghi, M.H. High capacity lossless data hiding based on histogram modification. *IEICE Electron. Express*, 2007, **4**(7), 205-210. doi: 10.1587/elex.4.205
34. Ou, B. & Zhao, Y. High capacity reversible data hiding based on multiple histograms modification. *IEEE Transact. on Circuits and Syst. Video Technol.*, 2019, **30**(8), 2329-2342. doi: 10.1109/TCSVT.2019.2921812
35. Sachnev, V.; Kim, H.J.; Nam, J.; Suresh, S. & Shi, Y.Q. Reversible watermarking algorithm using sort- ing and prediction. *IEEE Transact. on Circuits and Syst. Video Technol.,* 2009, **19**(7), 989-999. doi:10.1109/TCSVT.2009.2020257
36. Li, X.; Zhang, W.; Gui, X. & Yang, B. Efficient reversible data hiding based on multiple histograms modification. *IEEE Transact. Inf. Forensics and Security*, 2015 **10**(9), 2016-2027. doi: 10.1109/TIFS.2015.2444354
37. Qi, W.; Li, X.; Zhang, T. & Guo, Z. Optimal reversible data hiding scheme based on multiple histograms modification. *IEEE Transact. Circuits and Syst. Video Technol.*, 2019, **30**(8), 2300-2312. doi: 10.1109/TCSVT.2019.2942489
38. Wang, W. & Wang, W. HS-based reversible data hiding scheme using median prediction error. *Multimedia Tools and Appl.*, 2020, **79**, 18143-18165. doi: 10.1007/s11042-020-08682-3
39. He, J.; Chen, J. & Tang, S. Reversible data hiding in jpeg images based on negative influence models. *IEEE Transact. Inf. Forensics and Security*, 2020, **15**, 2121-2133. doi:10.1109/TIFS.2019.2958758
40. Shi, Y.; Li, X.; Zhang, X.; Wu, H. & Ma, B. Reversible data hiding: Advances in the past two decades. *IEEE Access*, 2016, 4, 3210–3237.

CONTRIBUTORS

Mr Sanjay Kumar obtained his MSc degree in Mathematics from Chaudhary Charan Singh University, Meerut. He is working as a Scientist at DRDO-SAG, Delhi, India. He is currently a PhD student in Department of Applied Mathematics, Delhi Technological University, New Delhi. His research interest includes: Reversible data hiding and public key cryptography.
In the present study, he has carried out complete design and development of technique. He has also completed the experimental work by evaluating the suggested technique and comparing the results with other state-of-the-art techniques.

Dr Gurjit Singh Walia obtained his PhD degree in the field of computer vision from Delhi Technological University, Delhi. He is working as Senior Scientist in DRDO-SAG, New Delhi. His current research interests include: Machine learning, pattern recognition, and information security.
In the present study, he has guided the author and supervised the work.

Dr Anjana Gupta is a Professor in Department of Applied Mathematics, Delhi Technological University Delhi, India. Her area of research is optimisation Techniques, Fuzzy Logic, MCDM, Computing with Words.
In the present study, she has provided expert guidance in problem formulation and offered necessary direction and overall support to carry out this study successfully.