

## Exploiting Laguerre transform in image steganography

Sudipta Kr Ghosal, ConceptualizationMethodologySupervisionFormal Analysis and Software <sup>a,\*</sup>, Souradeep Mukhopadhyay, Data curationWriting- Original draft preparationInvestigationSoftware and resources <sup>b</sup>, Sabbir Hossain, Data curationWriting- Original draft preparationInvestigationSoftware and Resources <sup>b</sup>, Ram Sarkar, SupervisionWriting- Reviewing and EditingVerification and Visualization <sup>b</sup>

<sup>a</sup> Department of Computer Science & Technology, Nalhati Government Polytechnic, Nalhati, Birbhum, Pin-731243, India

<sup>b</sup> Department of Computer Science & Engineering, Jadavpur University, Kolkata- 700032, India

## ARTICLE INFO

## Keywords:

Steganography  
Spatial domain  
Transform domain  
Laguerre transform  
LSB  
Payload  
PSNR

## ABSTRACT

Steganography, an approach used to conceal information into the digital media, generally works in two domains: spatial and transform. Though spatial domain methods are simpler, but transform domain methods are good at identifying the features which make the end system more secure. In this work, a novel Steganographic scheme based on an integer sequence named Laguerre transform (LT) is proposed. The Cover image is decomposed into non-overlapping m-pixel groups and then each such pixel group is transformed by applying LT. Variable length bits from the secret information are fabricated into the transformed components. A post-embedding adjustment is applied over these components to minimize the distortion. By applying Inverse LT (ILT), the m-pixel groups are re-computed from the resulting adjusted components. Experimental results reveal that disparity between cover and stego-pixels increases as m increases. Proposed scheme offers better stego image and higher payload compared to some state-of-the-art techniques. Code of this method is publicly available [here](#).

## List of acronyms used in this work.

Acronym	Main word	Acronym	Main word
BER	Bit Error Ratio	HI	Hidden Image/ Secret Image
Bpp	Bits Per Pixel	HVS	Human Visual System
BT	Binomial Transform	ILT	Inverse Laguerre Transform
CI	Cover Image	IWT	Integer Wavelet Transform
DCT	Discrete Cosine Transform	LP	Laguerre Polynomial
DHT	Discrete Hartley Transform	LSB	Least Significant Bit
DFT	Discrete Fourier Transform	LT	Laguerre Transform
DNA	Deoxyribo Nucleic Acid	MSE	Mean Square Error
DWT	Discrete Wavelet Transform	PSNR	Peak Signal-to-Noise Ratio

(continued on next page)

\* Corresponding author.

(continued)

Acronym	Main word	Acronym	Main word
EGF	Exponential Generating Function	SI	Stego Image
ER	Embedding Ratio	SSIM	Structural Similarity Index Measure
FIS	Fuzzy Interference System	WPD	Wavelet Packet Decomposition

## 1. Introduction

In the fast-expanding field of internet, protection of sensitive digital information has become a pressing need. Steganography, the practice that can conceal useful content within digital media such as image, audio and video etc., fulfils the said need since long. Among these digital media, image is considered to be most suitable carrier for concealing the secret content.

Steganographic methods are usually classified in two categories: spatial domain and frequency (transform) domain. Due to the robustness and stability in transform domain, researchers have explored a wide variety of Steganographic methods based on popular transforms such as DFT, DCT, DWT, BT and DHT etc. In Table 1, the advancement of the Steganography techniques along with the strengths and weaknesses has been summarized. It enables us to understand the research directions conceived by the researcher in the field of information hiding in last few decades.

Steganographic schemes discussed in Table 1 reveal that there are many weaknesses of those methods despite of having some strengths. These weaknesses include lack of security, fixed payload, low robustness and high computational cost. For instance, DCT produces fractional coefficients which make the calculation slow. Furthermore, DCT based methods are applicable on JPEG files as it considered some statistical distribution which is usually found in the CIs with JPEG extensions only. Further, DFT coefficients generate complex values as output and the usual computational complexity is  $O(n^2)$ . IWT, a specific kind of DWT, exploits the integer based calculation but when it is applied in Steganography domain, the payload obtained is low enough. To address these shortcomings, in this work, we have introduced a novel Steganography method based on LT. The fundamental idea of LT is to obtain an integer polynomial sequence based on pixel level addition and multiplication in coefficient representation. In contrast to the existing transforms, the calculation of LT is integer based which makes the operations faster. Our proposed method supports a number of image formats such as BMP, PPM, PGM and TIFF etc. other than JPEG. LT does not generate complex output as DFT and hence the computational complexity is reduced to  $O(n \log(n))$  [15]. Unlike IWT based scheme, our proposed method offers superior results in terms of payload and visual clarity (i.e., PSNR and SSIM) which can be verified from Table 2 and Table 3.

Some key advantages of LT with reference to the proposed Steganographic method are mentioned below:

- The transform performs integer based calculation and therefore the operations are faster.
- The binary factorization of the transform matrix enables efficient hardware implementation.

**Table 1**  
Comparative study on different transform domain based Steganographic methods.

Method	Strengths of the method	Weaknesses of the method
Bhattacharyya et al. [1] method based on DFT and LSB	Robust against Chi square analysis and histogram detection.	Visual distortion in SI is high.
Kumar et al. [2] DWT based method	Secure against various attacks such as Gaussian noise, Sharping, median filtering and Gamma Correction	PSNR and payload is low.
Biswas et al. [3] method based on LSB and DCT	Good carrier capacity and SI Quality.	Computational complexity is high.
Mandal et al. [4] method based on 2D-DHT	Less computational complexity in processing of real data type.	PSNR is not good with respect to payload, and not secure against attack.
Ghosal et al. [5] method based on BT and LSB	Good payload, calculation complexity is low	Low PSNR value with respect to payload
Mandal et al. [6] method based on Z transform	Good visual clarity SIs are obtained by exploiting Z transform.	Low Payload but high computational complexity.
Raja et al. [7] method based on DWT and IWT	Good payload capacity and improved security.	Lower PSNR for DWT and formatted images.
Xuan et al. [8] method based on IWT	Lossless recovery of secret data	Visual distortion is notable.
Seyyedi et al. [9] method based on IWT	Secure method because it can resist steganalysis.	Low PSNR
Atta et al. [10] method based on WPD and Neutrosophic set	Resist RS-steganalysis and pixel differencing histogram analysis.	High computation complexity but low payload.
Jothy et al. [11] method based on IWT	Integer transform ensures faster calculation. Security is high	PSNR is significantly low.
Ghosal et al. [12] method based on Lah transform	Faster in executionXXHighly Robust	Cannot support compressed image files such as JPEG
Kalita et al. [13] method based on IWT and LSB	No loss of data.XXStable against various electronic attacks.	Visual quality is low due to high distortion. XXPayload is not high.
Nazari et al. [14] method based on FIS and DCT	HVS parameters are applied to augment imperceptibility in optimal number of FIS rules.	Payload is not good, and not much secure against steganalysis.

**Table 2**

Performance of the proposed method in terms of PSNR with respect to 1, 2, 3 and 4 bpp of payload.

CI	Dimension	PSNR with respect to B bpp of payload							
		m = 2				m = 3			
		B = 1	B = 2	B = 3	B = 4	B = 1	B = 2	B = 3	B = 4
Lena	128 × 128	49.43	44.12	38.07	32.01	46.03	38.74	32.88	26.57
	256 × 256	49.57	43.98	37.98	32.01	46.04	38.81	32.88	26.64
	512 × 512	49.30	43.87	37.82	31.95	46.04	38.56	32.79	26.70
Baboon	128 × 128	49.42	44.10	38.05	31.98	46.05	38.75	32.99	26.60
	256 × 256	49.51	44.01	37.92	31.95	46.08	38.89	32.90	26.65
	512 × 512	49.54	43.86	37.81	31.67	46.09	38.57	32.85	26.75
Pepper	128 × 128	49.35	43.78	38.13	32.12	47.09	39.77	32.89	26.76
	256 × 256	49.22	43.76	37.98	31.96	47.06	39.56	32.90	26.87
	512 × 512	49.15	43.71	37.84	31.74	47.07	39.55	32.56	26.50
Splash	128 × 128	49.23	43.67	37.75	32.15	46.15	38.55	32.45	26.70
	256 × 256	49.31	43.80	37.63	32.05	46.09	38.77	32.42	26.97
	512 × 512	49.42	43.89	37.71	31.80	46.19	38.14	32.56	26.93
Sailboat	128 × 128	49.36	43.82	37.81	32.14	47.12	39.34	32.12	26.85
	256 × 256	49.19	43.78	37.69	31.78	46.15	39.48	32.56	26.60
	512 × 512	49.38	43.66	37.61	31.69	46.56	39.85	32.95	26.76
Fishing boat	128 × 128	49.31	43.72	37.78	32.15	46.87	38.67	32.78	26.60
	256 × 256	49.25	43.11	37.55	31.88	46.50	38.79	32.90	26.76
	512 × 512	49.17	43.78	37.67	31.76	46.87	38.98	32.24	26.74
Airplane	128 × 128	49.45	43.75	37.79	32.25	47.12	38.42	32.78	26.99
	256 × 256	49.36	43.84	37.70	31.98	47.19	38.67	32.98	26.87
	512 × 512	49.21	43.8	37.7	31.81	47.57	38.39	32.57	26.49
Aerial	128 × 128	49.32	43.76	37.80	32.14	46.66	39.12	32.58	26.51
	256 × 256	49.26	43.78	37.7	31.91	46.30	38.89	32.38	26.67
	512 × 512	49.18	43.79	37.67	31.7	46.28	38.87	32.62	26.12
Tank	128 × 128	49.17	43.75	37.8	32.15	46.82	38.67	32.34	26.15
	256 × 256	49.25	43.81	37.65	31.86	46.34	38.55	32.57	26.18
	512 × 512	49.08	43.76	37.70	31.73	46.33	38.45	32.58	26.16
Stream and Bridge	128 × 128	49.37	43.84	37.70	32.06	47.55	38.43	32.77	26.19
	256 × 256	49.28	43.83	37.64	31.90	47.77	38.67	32.22	26.34
	512 × 512	49.29	44.06	37.66	31.64	47.80	38.81	32.56	26.56

- The transformed coefficients are calculated based on the pixel values of a subset of the image block and that the forward and inverse transforms are equivalent.
- The computational complexity of the said transform is  $O(n \times \log(n))$  which is faster than DFT's complexity that is  $O(n^2)$ .
- The proposed transform domain Steganographic method supports widely used image formats such as PPM, PGM, BMP, and TIFF.
- Unlike the conventional transform domain Steganographic methods, the proposed scheme offers variable payload (up to 4 bpp).
- The SIs have been tested using StegExpose tool which proves that the robustness of the method is high.

The organization of the paper is as follows: [Section 2](#) gives an overview of LT. Proposed technique is explained in [Section 3](#). Experimental results, analysis and the discussion are reported in [Section 4](#). Complexity analysis and security analysis are provided in [Section 5](#) and [Section 6](#) respectively. Lastly, the paper is concluded in [Section 7](#).

## 2. Laguerre Transform: an overview

LT was first introduced by a famous mathematician named McCully in 1960 [16]. However, the said transform was represented in continuous form and hence, could not be exploited in image Steganography as the images are discrete in nature. In 2007, Paul Barry [17] extended McCully's paper in which he represented the said transform in discrete integer matrix version and its inverse within the context of exponential Riordan arrays.

The exponential Riordan group is represented as a set of infinite lower-triangular integer matrices. Each matrix is defined by a pair of generating functions  $u(x) = 1 + u_1x + u_2x^2 + \dots$  and  $v(x) = v_1x + v_2x^2 + \dots [v_1 \neq 0]$ . The associated matrix is the matrix whose  $k$ -th column has EGF  $u(x)v^k(x)/k!$ .

LT is considered to be the transform with matrix given by

$$Lag = \left[ \frac{1}{1-x}, \frac{x}{1-x} \right] \quad (1)$$

The ILT is given by,

$$Lag^{-1} = \left[ \frac{1}{1+x}, \frac{x}{1+x} \right] \quad (2)$$

**Table 3**

Performance analysis of SSIM values of proposed Steganographic scheme with respect to 1, 2, 3 and 4 bpp of payload.

CI	Dimension	SSIM with respect to B bpp of payload							
		m = 2				m = 3			
		B = 1	B = 2	B = 3	B = 4	B = 1	B = 2	B = 3	B = 4
Lena	128 × 128	0.99	0.99	0.96	0.87	0.99	0.96	0.88	0.71
	256 × 256	0.99	0.98	0.94	0.81	0.98	0.94	0.83	0.61
	512 × 512	0.99	0.98	0.92	0.77	0.98	0.93	0.79	0.54
Baboon	128 × 128	0.99	0.99	0.97	0.91	0.99	0.97	0.86	0.61
	256 × 256	0.99	0.99	0.96	0.91	0.99	0.95	0.83	0.57
	512 × 512	0.99	0.98	0.95	0.90	0.99	0.93	0.78	0.49
Pepper	128 × 128	0.99	0.99	0.95	0.83	0.99	0.96	0.87	0.65
	256 × 256	0.98	0.98	0.92	0.75	0.99	0.99	0.79	0.52
	512 × 512	0.99	0.97	0.88	0.68	0.98	0.92	0.75	0.46
Splash	128 × 128	0.99	0.99	0.94	0.83	0.99	0.96	0.86	0.68
	256 × 256	0.99	0.98	0.92	0.79	0.99	0.95	0.82	0.62
	512 × 512	0.99	0.98	0.91	0.75	0.99	0.93	0.78	0.58
Sailboat	128 × 128	0.99	0.98	0.94	0.83	0.99	0.95	0.82	0.67
	256 × 256	0.99	0.98	0.92	0.79	0.99	0.92	0.82	0.61
	512 × 512	0.99	0.98	0.91	0.75	0.99	0.93	0.79	0.58
Fishing boat	128 × 128	0.99	0.98	0.97	0.93	0.99	0.99	0.93	0.84
	256 × 256	0.99	0.99	0.98	0.91	0.99	0.99	0.92	0.78
	512 × 512	0.99	0.99	0.97	0.91	0.99	0.97	0.93	0.79
Airplane	128 × 128	0.99	0.98	0.91	0.81	0.99	0.95	0.84	0.61
	256 × 256	0.99	0.98	0.92	0.76	0.99	0.94	0.78	0.57
	512 × 512	0.99	0.97	0.90	0.74	0.99	0.92	0.77	0.55
Aerial	128 × 128	0.99	0.99	0.96	0.87	0.99	0.98	0.89	0.70
	256 × 256	0.99	0.98	0.93	0.78	0.99	0.95	0.82	0.59
	512 × 512	0.99	0.97	0.89	0.71	0.99	0.92	0.75	0.50
Tank	128 × 128	0.99	0.99	0.97	0.89	0.99	0.98	0.89	0.76
	256 × 256	0.99	0.98	0.95	0.85	0.99	0.97	0.85	0.71
	512 × 512	0.99	0.98	0.93	0.80	0.99	0.92	0.84	0.66
Stream and Bridge	128 × 128	0.99	0.98	0.95	0.83	0.99	0.97	0.88	0.69
	256 × 256	0.99	0.97	0.92	0.78	0.99	0.95	0.83	0.65
	512 × 512	0.99	0.98	0.91	0.76	0.99	0.93	0.78	0.58

$$\begin{aligned}
\text{General term of LT} &= \text{Lag}(n, k) = \frac{n!}{k!} [x^n] ((1-x)^{-1} x^k (1-x)^{-k}) \\
&= \frac{n!}{k!} [x^{n-k}] ((1-x)^{-1-k}) \\
&= \frac{n!}{k!} \sum_{t=0}^{\infty} \binom{k+t}{t} x^t \\
&= \frac{n!}{k!} \binom{n}{k}
\end{aligned}$$

Let  $p_0, p_1, \dots, p_n$  be an m-pixel group of the CI. Thus if  $t_n$  is the LT coefficients corresponding to the m-pixel group  $p_0, p_1, \dots, p_n$  then we have,

$$t_n = \sum_{k=0}^n \frac{n!}{k!} \binom{n}{k} p_n \quad (3)$$

where, for all n,  $0 \leq n \leq m-1$ .

Hence, for  $m = 3$  pixel group, the transformed component ( $t_i$ ) is obtained using Eq. (3) as follows:

$$t_i = \begin{cases} p_0 & : i = 0 \\ p_0 + p_1 & : i = 1 \\ 2p_0 + 4p_1 + p_2 & : i = 2 \end{cases}$$

Consequently, the EGF of  $t_n$  can be expressed as  $\frac{1}{1-x} f\left(\frac{x}{1-x}\right)$  where  $f(x)$  is the EGF of  $p_n$ . The inverse matrix of Lag i.e., ILT re-computes the m-pixel group as follows:

$$p_n = \sum_{k=0}^n (-1)^{n-k} \frac{n!}{k!} \binom{n}{k} t_n \quad (4)$$

where, for all  $n$ ,  $0 \leq n \leq m - 1$ .

For  $m = 3$  pixel group, the pixel value ( $p_i$ ) can be obtained from Eq. (4) as follows:

$$p_i = \begin{cases} t_0 & : i = 0 \\ -t_0 + t_1 & : i = 1 \\ 2t_0 - 4t_1 + t_2 & : i = 2 \end{cases}$$

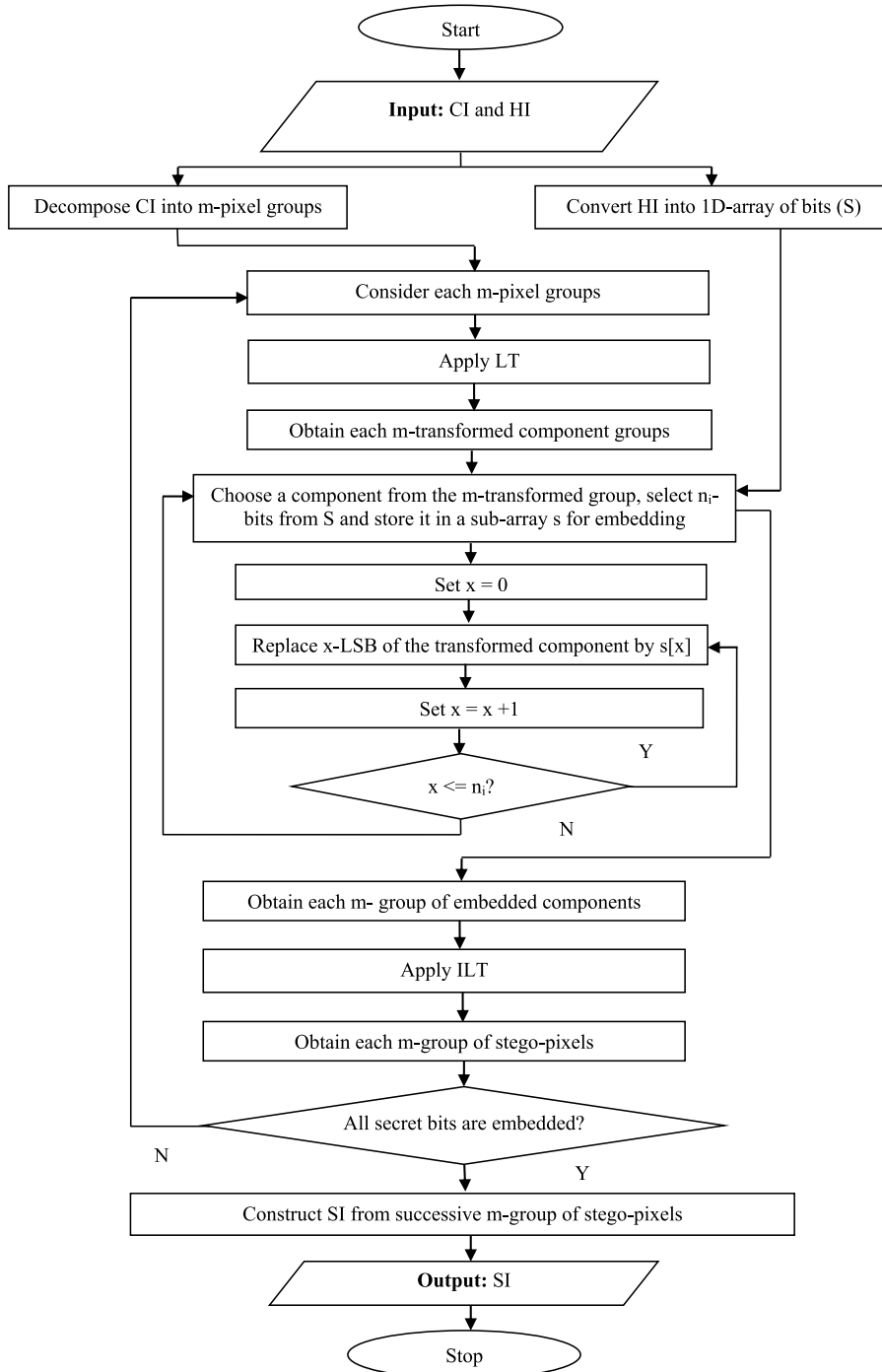


Fig. 1. Flow diagram of the embedding phase used in proposed Steganography scheme.

### 3. Proposed technique

In this section, our novel Steganographic scheme based on an integer sequence called LT has been reported. The CI is decomposed into non-overlapping m-pixel groups which in turn are converted into transform domain using LT in row major order. Variable length bits from the secret information are fabricated into the transformed components. A post-embedding adjustment is applied over the embedded component to minimize the distortion. ILT is applied to re-compute the m-pixel groups in spatial domain. The process is repeated until the secret bits are completely fabricated and the SI is produced. The receiver performs the reverse process to recover the secret bit-stream from SI.

In Sections 3.1 and 3.2, we elaborate the embedding and extracting processes of the proposed scheme, respectively. For easy understanding, an example of the proposed method is also provided in Section 3.3.

#### 3.1. Embedding

In this phase, the CI is decomposed into non-overlapping m-pixel groups in considering a row major order. Each such pixel group is then transformed by LT following Eq. (3). To get a payload  $B$  bits per pixel (bpp),  $n_i$  numbers of secret bits from  $S$  (as obtained from HI) is fabricated into the  $i^{th}$  component of each m-transformed components group. Since, the major objective of proposed method is to accomplish a variable payload and acceptable SI's quality, smaller values of  $m$  i.e.,  $m = 2$  and  $m = 3$  have been used.

For  $m = 2$ ,  $n_i$  can be derived as,

$$n_i = \begin{cases} B : i = 1 \\ B : i = 2 \end{cases} \quad (5)$$

However, for  $m = 3$ ,  $n_i$  can be derived as,

$$n_i = \begin{cases} B : i = 1 \\ B - 1 : i = 2 \\ B + 1 : i = 3 \end{cases} \quad (6)$$

Eq. (5) ensures that the tolerance level against alterations for both components is equal and therefore, same number of bits is embedded into both transformed components. However, for 3-pixel groups, third transformed component has higher tolerance level than second component against alterations. Therefore, more bits are embedded into third component and least bits are embedded into the second component as given in Eq. (6). Subsequent to the embedding process, ILT is applied to each embedded m-transformed group to produce the stego-pixels. This process is repeated until and unless the embedding of entire secret bit-stream  $S$  within all such m-pixel groups is done and the final SI is produced. Fig. 1 depicts the flow diagram of the embedding procedure in great detail.

---

#### Pseudo code of Embedding technique ( $CI_R \times C$ , $HI_W \times z$ , $ER$ ) {taking $m$ ( $= 2$ or $3$ )-pixel groups}

---

```

1. Start
2. Length =  $w \times z \times 8$ ; //  $L$  = length of the secret bit stream.
3. if ( $C \% m \neq 0$ )
4.  $C' \leftarrow (C + m - C \% m)$ ; // How much padding is required?
5. end if
6.  $CI_R \times C' \leftarrow \text{Padding}(CI_R \times C)$ ;
7. for  $k \leftarrow 1$  to  $R$  do
8. for  $l \leftarrow 1$  to  $C'$  do
9.  $CI'(k, l) \leftarrow \text{LT}(CI(k, l), m)$ ; // LT is applied by decomposing the CI into m-pixel groups.
10. end for
11. end for
12. for  $k \leftarrow 1$  to  $w$  do
13. for  $l \leftarrow 1$  to  $z$  do
14.  $S[x] = \text{Binary Conversion}(HI(k, l))$ ; // Forming a secret bit stream array  $S$ [Length].
15.  $x = x + 1$ ;
16. end for
17. end for
18. for  $k \leftarrow 1$  to  $R$ 
19. for  $l \leftarrow 1$  to  $C'$ 
20.  $CI''(k, l) \leftarrow \text{LSB}(CI'(k, l), S[x], ER)$ ; // Embedding secret bit stream in transformed CI.
21. end for
22. end for
23. for  $k \leftarrow 1$  to  $R$  do
24. for  $l \leftarrow 1$  to  $C'$  do
25.  $CI'''(k, l) \leftarrow \text{coefficient adjustment}(CI''(k, l))$ ;
26. end for
27. end for
28. for  $k \leftarrow 1$  to  $R$  do
29. for  $l \leftarrow 1$  to  $C'$  do
30.  $SI(k, l) \leftarrow \text{ILT}(CI'''(k, l), m)$ ; // ILT is applied by dividing the CI into m-pixel groups.

```

(continued on next page)

(continued)

---

```

31. end for
32. end for
33. End

```

---

### 3.2. Extraction

The extraction process is operated by partitioning the SI into non-overlapping  $m$ -pixel groups in row major order. Each such pixel group are then transformed by applying LT as given in Eq. (3). For  $B$  bpp,  $n_i$  numbers of secret bits are extracted from the  $i^{\text{th}}$  transformed component of each  $m$ -transformed components group based on Eqs. (5) and (6). Identical  $m$ -pixel group (i.e.,  $m = 2$  or  $m = 3$ ) have been used for this purpose. Above process is repeated to ensure the re-construction of secret bit-stream  $S$  which in turn constitutes the HI.

---

#### Pseudo code of Extraction technique ( $SI_R \times C'$ )

---

```

1. Start
2. for  $k \leftarrow 1$  to  $R$  do
3. for  $l \leftarrow 1$  to  $C'$  do
4.  $SI'(k, l) \leftarrow LT(SI(k, l, m))$ ; // LT is applied by decomposing the SI into  $m$ -pixel groups.
5. end for
6. end for
7. for  $k \leftarrow 1$  to  $R$  do
8. for  $l \leftarrow 1$  to  $C'$  do
9.  $S[x] \leftarrow LSBExtract(SI'(k, l, ER))$ ; //Extraction of secret bit stream in transformed SI.
10.  $x = x + 1$ ;
11. end for
12. end for
13.  $HI = combine(S[x])$ ; //Generation of HI by secret bit stream
14. End

```

---

Fig. 2 depicts the flow diagram of the extraction process in a great detail.

### 3.3. Example

In this section, we have described both embedding and extraction processes through an example.

**In case of embedding**, let us consider a 3-pixel group ( $p_0, p_1, p_2$ ) of CI containing the pixel values as follows:

$$(p_0, p_1, p_2) = (124, 90, 230)$$

By applying LT as shown in Eq. (3), the transformed components become:

$$(t_0, t_1, t_2) = LT(124, 90, 230) = (124, 124 + 90, 2 \times 124 + 4 \times 90 + 230) = (124, 214, 838)$$

Let us assume, the secret bits to be fabricated are  $(011101111)_2$ . For an average payload of 3 bpp (i.e.,  $B = 3$ ), the secret bits are fabricated in the ratio 3: 2: 4 into  $t_0, t_1$  and  $t_2$  based on the principle of embedding as mentioned in Eq. (6). Here, 3, 2 and 4 LSBs are replaced in first, second and third transformed components respectively.

$$\begin{aligned} (t_0, t_1, t_2) &= LSB(124, 214, 838) = LSB(1111100, 11010110, 1101000110) = (1111011, 11010110, 1101001111) \\ &= (123, 214, 847) \end{aligned}$$

Now a post-embedding adjustment has been applied over the fabricated components to minimize the difference between the cover and stego pixels without hampering the fabricated data. Thus the modified transformed components followed by adjustment become:

$$(t''_0, t''_1, t''_2) = (123, 214, 847 - 2^4) = (123, 214, 831)$$

Now, ILT is applied to re-compute the pixels as follows:

$$(p'_0, p'_1, p'_2) = ILT(123, 214, 831) = (123, 214 - 123, 2 \times 123 - 4 \times 214 + 831) = (123, 91, 221)$$

It is evident that the 3-pixel group of the CI has been modified from  $\{124, 90$  and  $230\}$  into  $\{123, 91$  and  $221\}$ .

It is also observed from the above example that the differences between cover and stego pixels are very less despite of having 3 bpp of payload.

**In case of extraction**, at the receiver's end, the 3-pixel groups ( $p'_0, p'_1, p'_2$ ) of the SI are taken and LT is applied on them. Then the resulting transformed components become:

$$(t''_0, t''_1, t''_2) = LT(123, 91, 221) = (123, 123 + 91, 2 \times 123 + 4 \times 91 + 221) = (123, 214, 831)$$

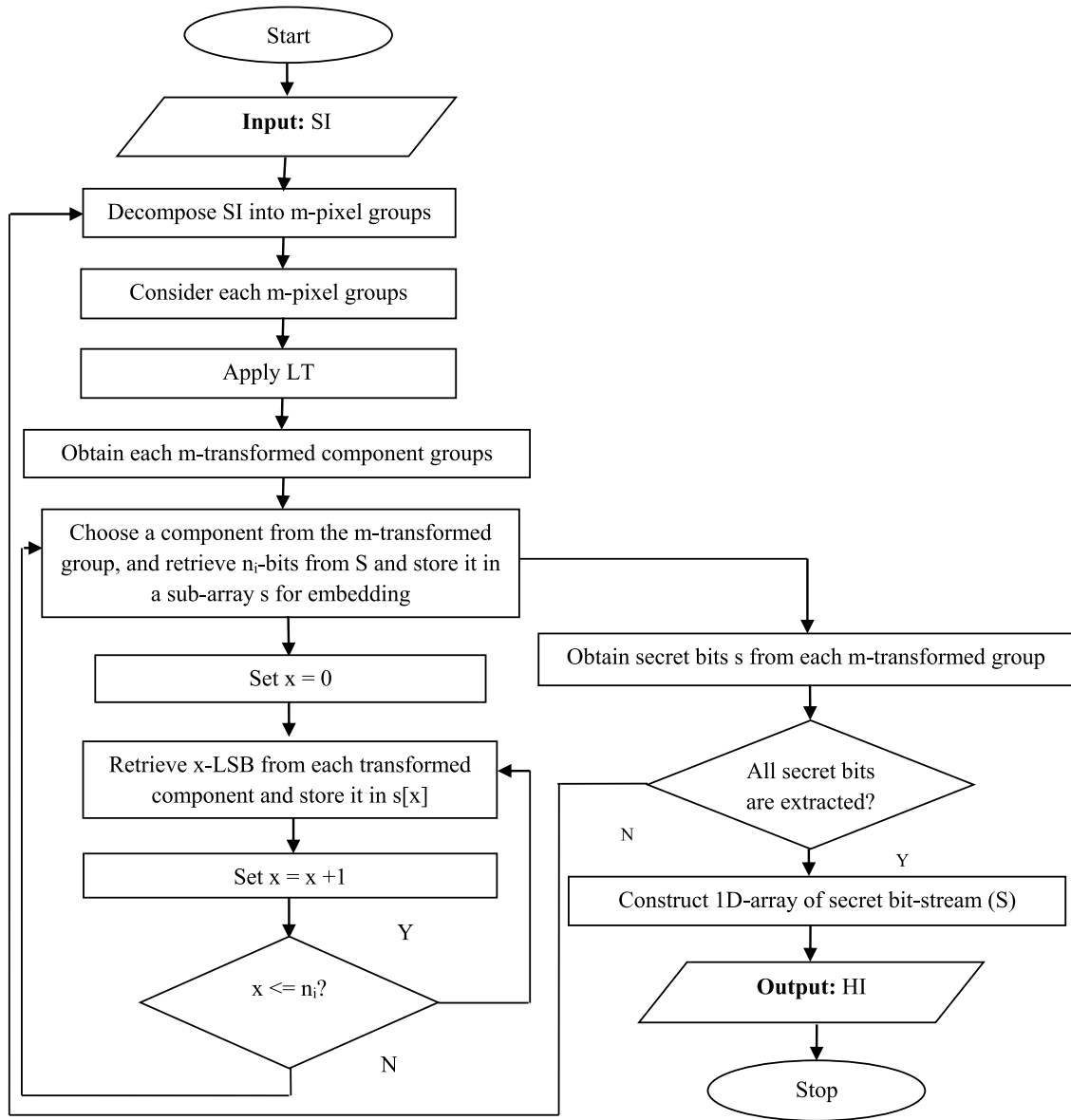


Fig. 2. Flow diagram of the extraction phase used in proposed Steganography scheme.

The secret bits are extracted from the LSBs of first, second and third transformed components in the ratio 3:2:4 as mentioned in Eq. (6).

$$\begin{aligned}
 S1 &= \text{LSBExtraction}(123, 3) = \text{LSBExtraction}(1111011, 3) = (011)_2 \\
 S2 &= \text{LSBExtraction}(214, 2) = \text{LSBExtraction}(11010110, 2) = (10)_2 \\
 S3 &= \text{LSBExtraction}(831, 4) = \text{LSBExtraction}(1100111111, 4) = (1111)_2
 \end{aligned}$$

By concatenating S1, S2 and S3, we obtain the secret bit stream S as  $(011101111)_2$ . From the example, one can easily verify that the recovered secret bit-stream is identical with the original bit-stream which was fabricated during embedding.

#### 4. Results, analysis and discussion

The performance of proposed method is primarily depends on how the pixel group is formed, since each group is composed of two or three pixels. As m i.e. the size of the pixel group increases, difference between cover and stego pixels also increases. Thus the results have been reported for both 2-pixel groups and 3-pixel groups. Here, 10 gray-scale images of USC-SIPI database [18] such as (i) Lena, (ii) Baboon, (iii) Pepper, (iv) Splash, (v) Sailboat, (vi) Stream and bridge, (vii) Tank, (viii) Fishing boat, (ix) Aerial and (x) Airplane have been taken to compute the results. For each of the 10 images, we have considered  $128 \times 128$ ,  $256 \times 256$  and  $512 \times 512$



dimensions to validate the results as shown in Fig. 3. We have used “MATLAB 2018a” as programming platform in Laptop with i5 8<sup>th</sup> generation, graphics AMD Radeon of 2 GB, 8 GB RAM and Windows 10 OS for coding purpose. Three widely acceptable metrics namely, PSNR (dB), SSIM and payload (bpp) have been used to analyze the effectiveness of the proposed Steganographic scheme.

We have evaluated the proposed method over the BMP images of USC SIPI database [18]. The grayscale “Male” is chosen as the hidden image throughout the experiments. However, the dimension of the said image has been resized to achieve variable payload. Let us consider that the  $n \times n$  “Male” image is to be embedded within  $R \times C$  CI to achieve ‘B’ bpp of payload. Then the dimension  $n \times n$  of the hidden image must satisfy the following condition:

$$n \times n \times 8 \leq R \times C \times B \quad (7)$$

In case of maximum payload,  $n$  should be  $n_{max}$

$$n_{max} = \text{floor} \left( \sqrt{\frac{R \times C \times B}{8}} \right) \quad (8)$$

The cover, hidden, stego and extracted images are shown in Fig. 4. The PSNR and MSE of the respective images are summarized too. The PSNR and MSE values are infinity and zero which ensure lossless extraction. Further, we have calculated the BER as  $\frac{\text{Error bits number}}{\text{Total secret bits}} \times 100\%$ , which is 0%. From this, we have concluded that perfect extraction is possible by our proposed method.

In Table 2, PSNR values are summarized with respect to the payload variation from 1 to 4 bpp. The minimum and maximum values of PSNR for 2-pixel group based embedding method are 31 dB and 49 dB respectively. Similarly, for 3-pixel group, minimum and maximum values of PSNR are 26 dB and 46 dB respectively. Now the PSNR above 30 dB for a SI is only considered as good quality image, we can see that embedding method based on 3-pixel group at 4 bpp generates SIs which offers PSNR below the acceptable level. Therefore, in PSNR vs. Payload trade-off, if PSNR becomes the point of concern then 1, 2 and 3 bpp of payloads are considered as the PSNR with respect to these payload values lie in the range [32 dB–46 dB]. Proposed method also offers consistent PSNR for all 10 images of Fig. 3, and as the payload increases, respective PSNR value drops.

In Table 3, SSIM values are summarized with respect to the payload variation from 1 to 4 bpp. The minimum and maximum values of SSIM for 2-pixel group based embedding method are 0.68 and 0.99 respectively. Similarly, for 3-pixel group, minimum and maximum values of PSNR are 0.46 and 0.99 respectively. Now the usual range of SSIM is [0.90–1.00], so the SSIM values obtained at 4bpp are mostly beyond the usual range. Therefore, in SSIM vs. Payload trade-off, if SSIM becomes the main concern, in that case one may select payload of 1, 2 and 3 bpp for 3-pixel groups and 1 and 2 bpp for 3-pixel groups for which the SSIM falls above the acceptable level. Proposed method also offers consistent SSIM for all 10 images of Fig. 3, and as the payload increases, respective SSIM value drops.

The performance of the proposed method is judged against state-of-the-art methods-Baseline method using IWT [8], Seyyedi’s method based on IWT [9] and Atta’s scheme [10]. Five benchmark images such as ‘Lena’, ‘Baboon’, ‘Pepper’, ‘Airplane’ and ‘Sailboat’

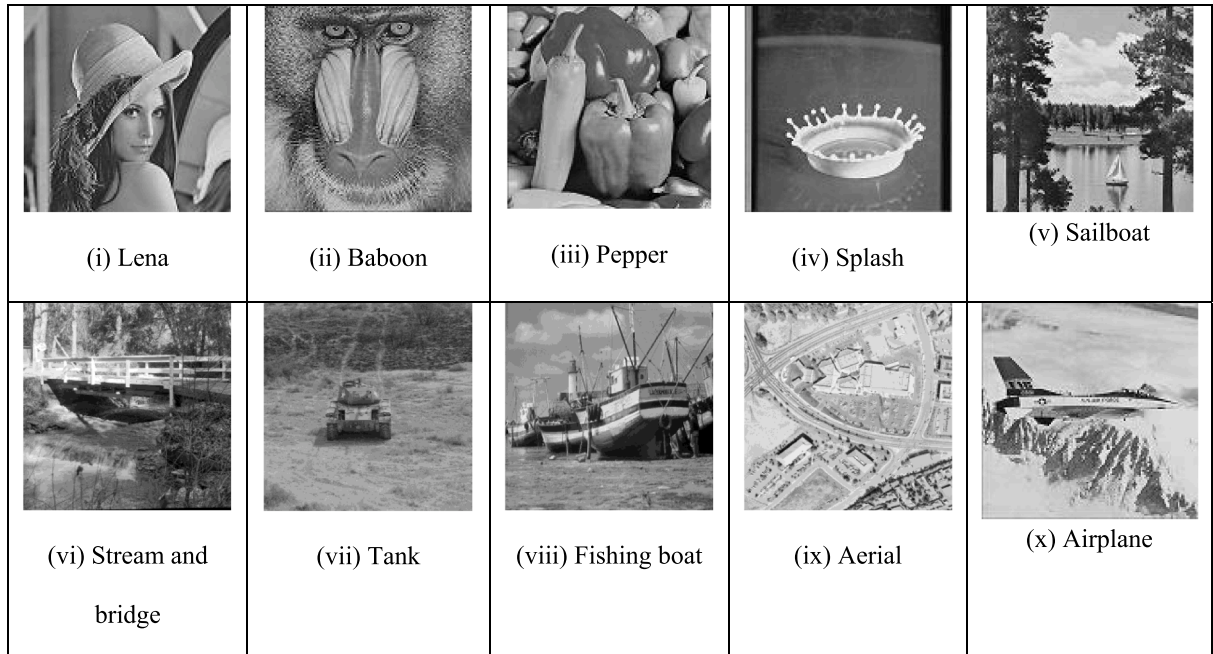


Fig. 3. Different grayscale CIs of dimension  $512 \times 512$ .



CI	Payload (bpp)	HI	SI (512×512) (m=2)	SI(512×512) (m=3)	Extracted HI (m=2)	Extracted HI (m=3)
 Lena (512× 512)	1	 Male (181× 181)	 PSNR: 49.3 dB	 PSNR: 46.03 dB	 Male (181× 181)	 Male (181× 181)
	2	 Male (256× 256)	 PSNR: 43.87 dB	 PSNR: 38.56 dB	 Male (256× 256)	 Male (256× 256)
	3	 Male (313× 313)	 PSNR: 37.82 dB	 PSNR: 32.79 dB	 Male (313× 313)	 Male (313× 313)
	4	 Male (362× 362)	 PSNR: 31.95 dB	 PSNR: 26..7 dB	 Male (362× 362)	 Male (362× 362)

Fig. 4. Cover, hidden, stego and extracted images with respect to different payloads at  $m = 2$  and  $m = 3$ .

have been investigated to evaluate the results as reported in Table 4. Since, the proposed scheme supports payload variation from 1 bpp to 4 bpp, PSNR values are reported for  $B = 1$ ,  $B = 2$ ,  $B = 3$  and  $B = 4$ . Compared to Baseline method using IWT [8], the payload hike of the proposed scheme in average-case is 4 times and the PSNR enhancement is more than 14 dB at  $B = 1$ . In contrast to Seyyedi's method based on IWT [9], the proposed one provides the increment of 0.48 bpp of average payload and 4.45 dB of average PSNR while  $B = 2$  is considered. In Table 4, the average-case payload at  $k = 2$  and  $k = 3$  for Atta's scheme [10] are 1.76 bpp and 2.5 bpp respectively. The proposed scheme does better than Atta's scheme [10] in terms of payload for all  $B$  values however the average PSNR value is slightly dropped. In spite of that it never falls below the acceptable level [i.e., 30 dB]. Therefore, the proposed scheme is considered to be superior over the existing methods reported here in terms of either PSNR or payload, or both. In Liu's [19,20] data hiding scheme, if more than 2 LSBs are substituted then visual quality significantly decreases. In our method, if on an average 4 LSBs are substituted, then also visual quality does not change significantly (PSNR > 32 dB). When we compare our method with M. Kalita's scheme [13] then it can be easily noticed that PSNR of our SI is almost 4dB higher in almost comparable payload (1 bpp). Compared with Nazari's scheme [14], it is found that the PSNR and payload of our method are 2.1 dB and 0.65 bpp higher. Our method also achieves an average PSNR enhancement of 4 dB with double payload than Jothy's method [11].

In Table 5, we have summarized the embedding and extraction time respectively for different size of "Lena" and "Baboon" of different payload. It gives us a better understanding of time complexity of our method with respect to different CIs and different payload.

Further, to prove the superiority of proposed method, we have computed the results of basic DCT, DFT and DWT amalgamating with LSB replacement and compared it with our method in terms of PSNR, payload and computational time. Here, "Lena" image of size  $512 \times 512$  is used for CI and resized "Male" image is used for HIs are exploited for better understanding. In Table 6, we have provided

**Table 4**

Comparison of the proposed scheme with state-of-the-art methods in terms of payload and PSNR.

CI	Technique	Payload (B) (bpp)	PSNR (dB)
Lena	Baseline IWT [8]	0.326	36.64
	Seyyedi's method [9]	1.52	40.54
	Atta's method [10]	1.7(k=2)/ 2.5(k=3)	44.91/38.77
	M. Kalita et al. [13]	1.14	43.9
	Nazari et al. [14]	0.24	49.65
	Jothy et al. [11]	0.5	44.2
	<b>Proposed scheme</b>	<b>1.00/2.00</b>	<b>49.30/43.87</b>
Baboon	Baseline IWT [8]	0.0569	32.76
	Seyyedi's method [9]	1.52	38.07
	Atta's method [10]	1.84(k=2)/ 2.5(k=3)	44.59/39.15
	M. Kalita et al. [13]	1.14	43.9
	Nazari et al. [14]	0.19	49.65
	Jothy et al. [11]	0.5	44.8
	<b>Proposed scheme</b>	<b>1.00/2.00</b>	<b>49.02/43.78</b>
Pepper	Baseline IWT [8]	0.264	29.11
	Seyyedi's method [9]	1.52	40.64
	Atta's method [10]	1.77(k=2)/ 2.5(k=3)	45/39.10
	M. Kalita et al. [13]	1.14	43.9
	Nazari et al. [14]	0.28	45.52
	Jothy et al. [11]	0.5	44.3
	<b>Proposed scheme</b>	<b>1.00/2.00</b>	<b>49.15/43.71</b>
Airplane	Baseline IWT [8]	0.358	36.30
	Seyyedi's method [9]	1.52	40.18
	Atta's method [10]	1.80(k=2)/ 2.5(k=3)	44.44/38.02
	M. Kalita et al. [13]	1.14	43.9
	Nazari et al. [14]	0.68	41.59
	Jothy et al. [11]	0.5	44.4
	<b>Proposed scheme</b>	<b>1.00/2.00</b>	<b>49.21/43.8</b>
Sailboat	Baseline IWT [8]	0.17	35.47
	Seyyedi's method [9]	1.52	39.40
	Atta's method [10]	1.7(k=2)/ 2.5(k=3)	44.59/38.32
	M. Kalita et al. [13]	1.14	43.9
	Nazari et al. [14]	0.36	47.99
	Jothy et al. [11]	0.5	44.2
	<b>Proposed scheme</b>	<b>1.00/2.00</b>	<b>49.38/43.78</b>

**Table 5**

Embedding/Extraction time of proposed Steganographic scheme in terms of different values of payload.

CI	Dimension	Computational Time (s) of Embedding/Extraction w.r.t. B bpp of payload							
		m = 2				m = 3			
		B = 1	B = 2	B = 3	B = 4	B = 1	B = 2	B = 3	B = 4
Lena	128 × 128	0.70/0.50	0.72/0.64	0.78/0.77	0.95/0.96	0.62/0.54	0.63/0.67	0.69/0.72	0.75/0.99
	256 × 256	1.15/0.78	1.22/0.99	1.32/1.25	1.34/1.33	1.05/0.79	1.25/1.01	1.32/1.34	1.35/1.42
	512 × 512	1.55/1.41	1.75/1.65	1.92/1.88	2.26/2.15	1.51/1.34	1.75/1.60	1.87/1.87	2.26/2.11
Baboon	128 × 128	0.71/0.53	0.72/0.67	0.79/0.81	0.90/0.98	0.64/0.59	0.65/0.71	0.67/0.75	0.77/1.01
	256 × 256	1.12/0.79	1.25/1.02	1.35/1.31	1.56/1.40	1.10/0.95	1.22/1.03	1.39/1.36	1.53/1.40
	512 × 512	1.54/1.54	1.74/1.71	1.90/1.88	2.24/2.21	1.53/1.47	1.71/1.67	1.88/1.85	2.25/2.19

the comparative results and visual comparison of SIs generated by DCT, DFT and DWT with our proposed method is made in Fig. 5.

## 5. Complexity Analysis





We have analyzed the complexity of proposed method in this section. Gashkov et al. [15] proved that the calculation of inner multiplication and addition of LT in the additive basis B+ is bounded with upper bound  $O(m \times \ln(m))$  where m denotes the size of the pixel group. The basis containing all monotonic linear functions  $\{ax+by: a, b \in \text{Real}\}$  has lower bounds of the particular form  $\Omega(m \times \log(m))$ . Nevertheless, the computational complexity (T) of the proposed scheme is obtained by decomposing the CI having C columns and R rows into non-overlapping blocks of size  $1 \times m$  and then adding up the time complexities of the various stages sorted below:

- Obtaining m-pixel group (i.e., m-pixel group):-  $O(m)$
- Applying LT over each m-pixel group:-  $O(m \times \log(m))$
- Secret bits fabrication into the z-LSB positions (where  $0 \leq z \leq 4$ ):-  $O(1)$
- Adjustment of embedded coefficients:-  $O(m)$
- Applying ILT over each m-pixel group:-  $O(m \times \log(m))$

**Table 6**

Comparison of the proposed method with some transform based techniques DCT, DFT and DWT.

Transform used for embedding	No. of bits embedded	PSNR (dB)	Embedding Time (s)	Extraction Time (s)
DCT [21]	262144	20.25	1.72	2.01
DFT [22]	400	41.40	1.03	1.14
DWT [23]	131072	40.6	1.21	1.80
Proposed method	<b>400</b>	<b>60.62</b>	<b>0.80</b>	<b>0.18</b>
	<b>131072</b>	<b>50.13</b>	<b>1.12</b>	<b>0.78</b>
	<b>262144</b>	<b>49.02</b>	<b>1.54</b>	<b>1.41</b>

DCT	DFT	DWT	Proposed Method
			
PSNR : 20.25 dB ( 1 bpp)	PSNR : 41.4 dB (0.001 bpp)	PSNR : 40.6 dB (0.5 bpp)	PSNR : 49.02 dB (1 bpp)

**Fig. 5.** Comparison of SIs obtained through DCT, DFT, DWT and our proposed method.

The time complexity in worst-case is represented as,  $T = O\left(\frac{C \times R}{m}\right) \times \text{maximum}\{O(m), O(m \times \log(m)), O(1), O(m), O(m \times \log(m))\}$   
 $= O\left(\frac{C \times R}{m} \times m \times \log(m)\right) = O(C \times R \times \log(m))$ . Best-case yields only one  $m$ -pixel group /  $1 \times m$  pixel group of CI and hence,  $T = O\left(\frac{C \times 1}{m} \times m \times \log(m)\right) = O(C \times \log(m))$ .

## 6. Security Analysis

This section is very important as here we judge the efficacy of our proposed method. We use the StegExpose tool which performs as the combination of many well-known steganalysis techniques that can assist us to take decision whether a given SI is above the threshold or not. StegExpose executes as an amalgamation of all the techniques reported in Table 7 by investigating suspicious files.

Further, it is evident from Table 8 that all sample SIs passed the test which highlights that the SIs obtained through proposed scheme are robust to steganalysis attacks.

**Table 7**

Brief description of various steganalysis methods.

Method	Description
<b>Primary Sets</b>	It recognizes LSB method by forming subsets of pixels where cardinality varies due to inclusion of secret data.
<b>Sample pair analysis</b>	Here, selected multisets form the states of a finite state machine because its inexact modification of state causes between these multisets based on LSB flipping.
<b>RS analysis</b>	It differentiates between the singular and regular sets for the LSB and changed LSB plane.
<b>Chi-square attack</b>	It is applied to check the robustness of a system in terms of the visual and geometrical attacks. It is done by finding the difference between SI and CI by analysing the probability.

**Table 8**

StegExpose results on stego versions of “Lena” with respect to different values of B and m.

Steganalysis Attacks	m = 2				m = 3			
	B = 1	B = 2	B = 3	B = 4	B = 1	B = 2	B = 3	B = 4
Primary Sets	0.010	0.052	0.052	0.0713	0.003	0.018	0.1034	0.055
Chi Square	6.84E-04	7.64E-04	0.001444	1.79E-08	6.45E-04	5.87E-04	0.001311	0.002328
RS analysis	0.0124	0.0252	0.0942	0.071	0.015	0.0231	0.0443	0.0461
Sample pairs	0.0140	0.041	0.048	0.026	0.014	0.0187	0.0602	0.0291
Fusion (mean)	0.009527	0.02983	0.04925	0.0423	0.0085	0.015341	0.052336	0.033255
Secret message size in bytes	836	2618	4321	3713	749	1346	4592	2918
Below stego threshold?	True	True	True	True	True	True	True	True

## 7. Conclusion

In this paper, a new image Steganography method in the transform domain based on an integer sequence called LT has been reported. The primary advantage of using LT is the flexibility of dealing with integer based calculation which guarantees faster operation without data loss. The binary factorization of the transform matrix enables efficient hardware implementation. The transformed coefficients are calculated based on the pixel values of a subset of the image block and the forward and inverse transforms are equivalent. The proposed transform domain Steganographic method supports wide variety of image formats that include PPM, PGM, TIFF and BMP. Unlike the conventional transform domain Steganographic methods, the proposed scheme offers variable payload (up to 4 bpp). The SIs have been tested using StegExpose tool and it is found that the robustness is very high. As far as the weaknesses of LT are concerned, since the CI is partitioned into a set of  $m$ -pixel groups, the variable  $m$  is considered as a key factor in the SI's quality. For the steady rise of LT coefficients, we have observed from the test cases that distortion of the SI becomes significant while  $m > 3$ . Hence, the decomposition is  $1 \times m$  only and cannot support  $2 \times 2$  block based decomposition such as DFT or DWT and  $8 \times 8$  block based decomposition like DCT. In our future work, we aim to investigate some other integer sequence based transforms which have not been used in Steganography till date, and will be able to overcome the limitations of our proposed method. The security and quality of SI are also two aspects to be dealt with in future.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- [1] Bhattacharyya Debnath, Bandyopadhyay Rohit, Bandyopadhyay SK, kim Tai-Hoon. Discrete fourier transformation based image authentication technique. In: Proceedings of the 8th IEEE international conference on cognitive informatics Kowloon; 2009. <https://doi.org/10.1109/COGINF.2009.5250756>.
- [2] Vijay Kumar, Dinesh Kumar. "Performance evaluation of DWT based image steganography." 2010 IEEE 2nd international advance computing conference. 10.1109/IADCC.2010.5423005.
- [3] Biswas Rajib, Mukherjee Sayantan, Bandyopadhyay, Samir Kumar. "DCT domain encryption in LSB steganography. In: 2013 5th International Conference and Computational Intelligence and Communication Networks; 2013.
- [4] Ghosal SK, Mandal JK. Separate discrete hartley transform based invisible watermarking for color image authentication. Adv Intel Syst Comput 2013;2:767–76. [https://doi.org/10.1007/978-3-642-31552-7\\_78](https://doi.org/10.1007/978-3-642-31552-7_78).
- [5] Ghosal SK, Mandal JK. Binomial transform based fragile watermarking for image authentication. J Inform Security Appl 2014;19:272–81. <https://doi.org/10.1016/j.jisa.2014.07.004>. Elsevier.
- [6] Mandal JK. A frequency domain steganography using Z transform (FDSZT. In: International Workshop on Embedded Computing and Communication System; 2012.
- [7] Raja KB, Reddy HSManjunatha. Wavelet based Non LSB Steganography. Int J Adv Netw Appl 2011;1203–9.
- [8] Xuan G, Zhu J, Shi YQ, Ni Z, Su W. Distortionless data hiding based on integer wavelet transform. IEEE Electron Lett 2002;38:1646–8. <https://doi.org/10.1049/el:20021131>.
- [9] Seyyedi SA, Ivanov N. High payload and secure steganography method based on block partitioning and integer wavelet Transform". Int. J Security Appl 2014;8: 183–94. <https://doi.org/10.14257/ijisa.2014.8.4.17>.
- [10] Atta Randa, Ghanbari Mohammad. A high payload steganography mechanism based on wavelet packet transformation. J. Vis. Commun. Image R. 2018;53: 42–54. <https://doi.org/10.1016/j.jvcir.2018.03.009>.
- [11] Jothy N, Anusuyya S. A secure color image steganography using integer wavelet transform. In: 10th international conference on intelligent systems and control (ISCO); 2016. <https://doi.org/10.1109/ISCO.2016.7726948>.
- [12] Ghosal Sudipta Kr, Mukhopadhyay Souradeep, Hossain Sabbir, Sarkar Ram. Application of Lah transform for security and privacy of data through information hiding in telecommunication. Trans Emerging Tel Tech 2020;1–20. <https://doi.org/10.1002/ett.3984>. Wiley.
- [13] Kalita Manashee, Tuithong Themrichon, Majumder Swanirbhar. A new steganography method using integer wavelet transform and least significant bit substitution. The Comput J 2019;62:1639–55. <https://doi.org/10.1093/comjnl/bxz014>.
- [14] Nazari Mahboubeh, Ahmadi Iman Dorostkar. A novel chaotic steganography method with three approaches for color and grayscale images based on FIS and DCT with flexible capacity. Multimedia Tools Appl 2020;79:13693–724. <https://doi.org/10.1007/s11042-019-08415-1>.
- [15] Gashkov SB. Arithmetic complexity of certain linear transformations. Math Notes 2015;97:531–55. <https://doi.org/10.1134/S0001434615030256>.
- [16] McCully J. The Laguerre transform. Soc Indus Appl Math 1960;2:185–91.
- [17] Barry P. Some Observations on the Lah and Laguerre transforms of integer sequences. J Integer Seq 2007;10:2–3.
- [18] Weber AG. USC-SIPI image database: version 5, original release. Signal and Image Processing Institute, University of Southern California, Department of Electrical Engineering; 1997. <http://sipi.usc.edu/database/>. [Accessed 23 September 2019].
- [19] Liu Hongjun, Lin Da, Kadir Abdurahman. "A novel data hiding method based on deoxyribonucleic acid coding. Comput Electr Eng 2013;39:1164–73.
- [20] Liu Guoyan, Liu Hongjun. A. Hiding message into DNA sequence through DNA coding and chaotic maps. Med Biol Eng Comput 2014;52:741–7.
- [21] Suraj Kanya (2020), Watermark DCT, MATLAB Central File Exchange. Retrieved July 25, 2020. (<https://www.mathworks.com/matlabcentral/fileexchange/46866-watermark-dct>).
- [22] Lai Chih-Chin, Tsai Cheng-Chih. Digital image watermarking using discrete wavelet transform and singular value decomposition. IEEE Trans Instrum Meas 2010;59:3060–3.
- [23] Poljicak Ante, Mandic Lidija, Agic Darko. Discrete fourier transform-based watermarking method with an optimal implementation radius. J Electron Imaging 2011;20.

**Sudipta Kr. Ghosal** is a Lecturer in Computer Software Technology at Nalhati Government Polytechnic, West Bengal, India. He received his B. Tech in CSE in 2007, M. Tech in IT (Courseware Engineering) in 2010 and PhD (Engineering) in 2016. He has around 30+ publications in refereed journals/conferences.

**Souradeep Mukhopadhyay** is currently a sophomore of Computer Science and Engineering Dept. at Jadavpur University, Kolkata, India. He became KVPY fellow in 2018. His research interests mainly include Steganography, Optimization theory and Image and video processing.

**Sabbir Hossain** is currently an undergraduate student of Computer Science and Engineering at Jadavpur University, Kolkata, India. His research interests mainly include Steganography, Optimization theory, and Image and video processing.

**Ram Sarkar** received his B. Tech in Computer Science and Engineering from University of Calcutta in 2003. He received his M.C.S.E and PhD (Engineering) degrees from Jadavpur University (JU) in 2005 and 2012 respectively. He joined JU in 2008 where he is working as an Associate Professor. He was awarded Fulbright Nehru Fellowship for post-doctoral research in University of Maryland, College Park, USA during 2014-15. His areas of current research interest are Image Processing, Pattern Recognition Soft Computing, and Machine Learning. He is a member of the IEEE, USA.