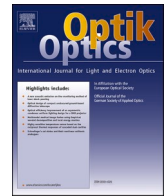




Contents lists available at ScienceDirect

Optik

journal homepage: [www.elsevier.com/locate/ijleo](http://www.elsevier.com/locate/ijleo)

# Image tamper detection and correction using Merkle tree and remainder value differencing

S N V J Devi Kosuru<sup>a</sup>, Gandharba Swain<sup>a,\*</sup>, Naweena Kumar<sup>a</sup>, Anita Pradhan<sup>b</sup>

<sup>a</sup> Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram 522302, Guntur, Andhra Pradesh, India

<sup>b</sup> Department of Computer Science and Engineering, PSCMR College of Engineering & Technology, Kothapeta, Vijayawada 520001, Andhra Pradesh, India

## ARTICLE INFO

### Keywords:

Data hiding  
Tamper detection  
Remainder value differencing  
Watermarking  
Tamper correction

## ABSTRACT

This article proposes an image tamper detection and correction technique. It uses the principle of differencing and addresses the fall off boundary problem (FOBP). It operates on  $2 \times 2$  non-overlapped blocks. From each pixel of a block the decimal value for 4 most significant bits (MSBs), is called as quotient, and the decimal value for 4 least significant bits (LSBs) is called as remainder. The 4 watermark bits (WBs) are computed from 4 quotients as the root of Merkle tree. These 4 WBs are XORed with the 4 bits generated from logistic map sequence to generate the 4 recovery bits (RBs). The 4 WBs and 4 RBs are stored in 4 remainders by LSB alteration and remainder value differencing (RVD). While extracting the watermark, the tampered blocks can be identified and a correction logic can be applied to get the true value of the pixels in the block. The peak signal-to-noise ratio (PSNR) value is 42.02 dB and structural similarity (SSIM) index value is 0.9765. The tampered blocks can be accurately identified and corrected.

## 1. Introduction

A watermarked image can be easily tampered by using the freely available tools [1]. Hence it is important to bring out efficient tamper detection and correction techniques. Some classes of well-known image watermarking techniques in spatial domain are, (i) block-based techniques, (ii) chaotic map (CM) and logistic map (LM) based techniques and (iii) Hamming code based techniques [2]. Singh & Singh [3] developed a fragile watermarking scheme with tamper correction using  $2 \times 2$  blocks. They computed 2 watermark bits (WBs) and 2 recovery bits (RBs) from the most significant bits (MSBs) of the 4 pixels and hid in least significant bits (LSBs) of the same pixels. Cao et al. [4] also used a similar approach with a variation that the different MSBs are used differently as per their priority to effect on visual quality. This technique is also very robust in terms of security. Gull et al. [5] computed the average value of a  $4 \times 4$  block, then applied exclusive or (XOR) operation on the average value and a secret key to generate the WBs. Then the average value and WBs are hidden in LSBs. Feng et al. [6] used  $2 \times 2$  blocks and computed the RBs from block average and WBs by pixel value comparison and parity check. Although these techniques can detect the tampered blocks, but cannot correct them efficiently. Qin et al. [7] proposed a fragile image watermarking technique for tamper detection using  $3 \times 3$  overlapped pixel blocks. They computed the average value from a  $3 \times 3$  block. The 6 LSBs of this average value is used to generate the RBs and the WBs. The WBs are embedded in

\* Corresponding author.

E-mail addresses: [2002030003@kluniversity.in](mailto:2002030003@kluniversity.in) (S.N.V.J. Devi Kosuru), [gandharba.swain@kluniversity.in](mailto:gandharba.swain@kluniversity.in) (G. Swain), [naweena@kluniversity.in](mailto:naweena@kluniversity.in) (N. Kumar), [dranitapradhan@kluniversity.in](mailto:dranitapradhan@kluniversity.in) (A. Pradhan).

<https://doi.org/10.1016/j.ijleo.2022.169212>

Received 8 January 2022; Received in revised form 24 April 2022; Accepted 27 April 2022

Available online 30 April 2022

0030-4026/© 2022 Elsevier GmbH. All rights reserved.

LSBs of central pixel and the RBs are embedded in rest of the pixels. It recovers the tampered regions effectively.

To secure the watermarks we can use various CMs in association with the WBs. Rawat & Raman [8] used Arnold's map to scramble the image so that security could be improved. They XORed the bits from LM sequence with WBs and hidden in LSBs of the pixels. As per the analysis by Botta et al. [9], Rawat & Raman's scheme is not capable of identifying the tampered pixels effectively. This problem was addressed by computing the WBs from 7 MSBs and hiding in LSBs. Prasad & Pal [12] also used CMs to improve the security. They used LM, shift operator and mod operator. In their scheme the WBs are derived from 5 MSBs and hidden in the pixel using shift operator and mod operator. Tampered zones are identified efficiently. Sahu [15] used XOR operation and LM for watermark embedding and tamper localization. It has been claimed that this technique is tolerant to various attacks including S&P noise, cropping attack and copy-paste attack. Tong et al. [16] also used CM for improving the security of watermarks. They used 2 dimensional cross-CM with the WBs. They derived the WBs from MSBs of the pixels in a  $2 \times 2$  block, then XORed with the CM bits and hidden in LSBs. They have claimed that their technique accurately identifies the tampered pixels.

If we can construct the original image (OI) from the watermarked image (WI) along with the watermark extraction, then we call it as reversible data hiding (RDH). Li et al. [19] proposed a RDH technique using pixel value ordering (PVO) and prediction error expansion (PEE). They found the maximum and minimum values by ordering the pixel values and then hid the data via PEE. They achieved good peak signal-to-noise ratio (PSNR) value. Peng et al. [20] also proposed a PVO by considering new differences in an improved way to achieve greater hiding capacity (HC) along with reversibility. Hong et al. [21] also developed a RDH scheme based on PVO. They derived the WBs from hash value of block characteristics and camouflaged in LSBs. Gao et al. [22] also proposed a RDH scheme with tamper detection, wherein the feature bit matrix is generated from region of interest (ROI) and these bits are hidden in LSBs. The tamper detection is possible along with the extraction of OI. This technique achieves better performance in terms of tamper detection and contrast enhancement. Hurrah et al. [23] proposed a RDH scheme for medical images. They encrypted the WBs using binary grey code, Arnold map and advanced encryption standard (AES) algorithms. The encrypted WBs are then hidden in LSBs of the pixels in a  $2 \times 2$  block. The PSNR is good and the embedding and extraction can be done in few seconds.

Chang et al. [10] proposed a watermarking scheme, wherein the WBs are derived from 4 MSBs using Hamming code and 2-pass LM. The WBs are camouflaged in LSBs. This scheme avoids various attacks, effectively detects tampered regions. Trivedy & Pal [11] used Hamming weight and LM to derive the WBs. The watermark bits are camouflaged in the pixels either by incrementing/decrementing or keeping the pixel value unchanged. This technique efficiently detects the tampered zones in the image. Prasad & Pal [13] used LM, Hamming code, and pixel value differencing (PVD) for watermarking. In a  $1 \times 2$  block ( $P_1, P_2$ ), the 3 WBs are generated from 2 MSBs of  $P_1$  and 2 MSBs of  $P_2$  in association with Hamming code. For the Hamming code, 2 MSBs of  $P_1$  and 2 MSBs of  $P_2$  are the data bits and the 3 parity bits generated are the WBs. These 3 WBs are then embedded in 6 LSBs of  $P_1$  and 6 LSBs of  $P_2$  by PVD. Prasad & Pal [14] also proposed pixel level tamper detection using Hamming code and LM. Generated the WBs from 4 MSBs of a pixel after multiplying with a generator matrix. The generator matrix produces 7 bits output, the last 3 bits are treated as WBs and embedded in 3 LSBs. Nazari et al. [17] also used CM for watermarking in  $2 \times 2$  blocks. From the 5 MSBs of the 4 pixels of a block, an information array is created and embedded in LSBs. The length of the information array is variable based on the smoothness of the block. The security is improved due to the CM. Sreenivas & Kamakshiprasad (Sreenivas & KP) [18] also proposed a fragile watermarking scheme using CM. From a  $2 \times 2$  block, 12 WBs are created using CM, further it is reduced to 4 bits to provide space for RBs. For each block 2 sets of RBs of lengths 5 and 3 are computed. The WBs and RBs are embedded in LSBs of the pixels of the block. The tampered regions are detected and corrected accurately.

The paper is organized in the following manner. In Section 2, research contributions by the authors are highlighted. In Section 3, the watermark embedding, tamper detection, and tamper correction procedures are narrated. In Section 4, the experimental results are furnished and in Section 5 the research is concluded.

## 2. Research contribution by authors

The primary goal of an image watermarking technique is to detect the tampered pixels and correct them. It has been a continuous effort by researchers to improve the accuracy of tamper detection and correction techniques. This proposed watermarking technique has the following features.

- It takes  $2 \times 2$  blocks. Considers the 4 MSBs of a pixel as a quotient and the 4 LSBs of a pixel as a remainder. The WBs are computed from the 4 quotients by Merkle root. These WBs are XORed with the 4-bit sequence generated from LM to generate the RBs. The 4 WBs and 4 RBs are embedded in the 4 remainders by LSB substitution and remainder value differencing (RVD).
- Out of 4 remainder values in a block one remainder is used as central remainder and the other 3 are used as associate remainders. If any one of the 3 associate remainders are changed/tampered, then it can be identified and corrected.
- Here, while masking WBs and RBs in the remainders, the RVD concept is used. Precautions are taken to check FOBP and avoid it.

## 3. Proposed MT and RVD based tamper detection technique

In Section 3.1 the use of LM is illustrated. In Section 3.2 the watermark camouflaging mechanism is narrated. In Section 3.3 the watermark extraction, tamper detection and correction mechanisms are discussed.

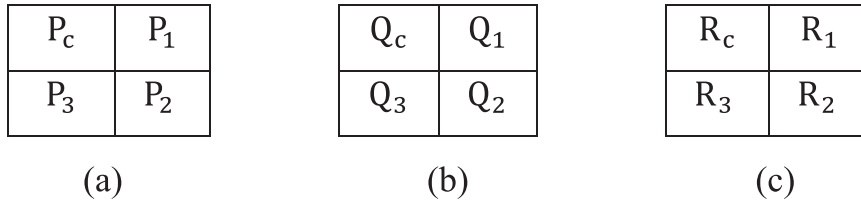


Fig. 1. (a) Pixel block, (b) Quotient block, and (c) Remainder block.

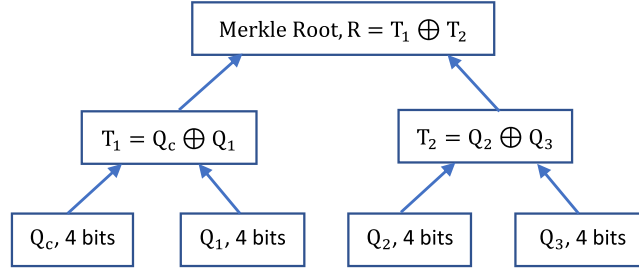


Fig. 2. Merkle Root calculation.

Table 1

Range division.

Range	{0, 3}	{4, 7}	{8, 11}	{12, 15}
LB	0	4	8	12
UB	3	7	11	15

### 3.1. Logistic map

LM is a simple chaotic function. It generates a disorder list of elements with two seed values. The seed values are  $\alpha_0$  and  $\beta$ . These seed values should be chosen by satisfying the conditions  $0 \leq \alpha_0 \leq 1$  and  $0 < \beta \leq 4$ . For  $i = 1$  through  $n$ , we can compute the subsequent values of the sequence using Eq. (1).

$$\alpha_i = \beta \times \alpha_{i-1} \times (1 - \alpha_{i-1}) \quad (1)$$

Here we have considered  $\alpha_0$  value as 0.0396 and  $\beta$  as 3.58. The map is chaotic when  $3.57 < \beta \leq 4$ . After generating the sequence of values  $\alpha_i$ , we multiply by 255 to get a sequence  $X_i$  and round up the value of each  $X_i$  to produce the sequence  $Y_i$ , then apply mod 16 on  $Y_i$  to get  $Z_i$  using Eq. (2). Each value of  $Z_i$  is an integer less than or equal to 15. Now convert  $Z_i$  to 4 binary bits to obtain  $S_i$ .

$$X_i = \alpha_i \times 255, Y_i = \text{Round}(X_i), \text{ and } Z_i = Y_i \bmod 16 \quad (2)$$

### 3.2. Watermark embedding

The image is scanned in raster scan order and divided into  $2 \times 2$  non-overlapped blocks. In each block WBs are created from the 4 MSBs of the pixels and embedded in 4 LSBs of the pixels. The watermark creation and embedding procedure is as follows. Consider a  $2 \times 2$  block shown in Fig. 1(a). The four pixels are  $P_c, P_1, P_2$ , and  $P_3$ . From the pixel block we shall create quotient block and remainder block using Eq. (3). Fig. 1(b) represents the quotient block and Fig. 1(c) represents the remainder block. The “div” stands for the quotient division and “mod” stands for remainder division. In other words, the quotient values are the decimal representation for 4 MSBs of the pixels and the remainder values are the decimal representation for the 4 LSBs of the pixels.

$$Q_c = P_c \text{ div } 16, R_c = P_c \bmod 16, Q_i = P_i \text{ div } 16, \text{ and } R_i = P_i \bmod 16, \text{ for } i = 1, 2, \text{ and } 3 \quad (3)$$

Compute the Merkle root,  $R$  using Eq. (4). Here,  $\oplus$  stands for bit-by-bit XOR operation. This computation is shown in Fig. 2.

$$T_1 = Q_c \oplus Q_1, T_2 = Q_2 \oplus Q_3, \text{ and } R = T_1 \oplus T_2 \quad (4)$$

Suppose the 4 bits in  $R$  are  $b_4b_3b_2b_1$ . Hence, WBs =  $b_4b_3b_2b_1$ . Take the next  $S_i$  value from the LM and let its 4 bits are  $t_4t_3t_2t_1$ . Now compute the 4 RBs  $b_8b_7b_6b_5 = t_4t_3t_2t_1 \oplus b_4b_3b_2b_1$ . Now we shall camouflage the 4 WBs ( $b_4b_3b_2b_1$ ) and 4 RBs ( $b_8b_7b_6b_5$ ) in 4 remainders. Camouflage  $b_2b_1$  in LSBs of  $R_c$ , and let the new value be represented as  $R'_c$ . Compute the decimal value of  $b_4b_3$  as  $V_1$ , decimal value of  $b_6b_5$  as  $V_2$ , and decimal value of  $b_8b_7$  as  $V_3$ . Compute three difference values  $d_1, d_2$ , and  $d_3$  as in Eq. (5).

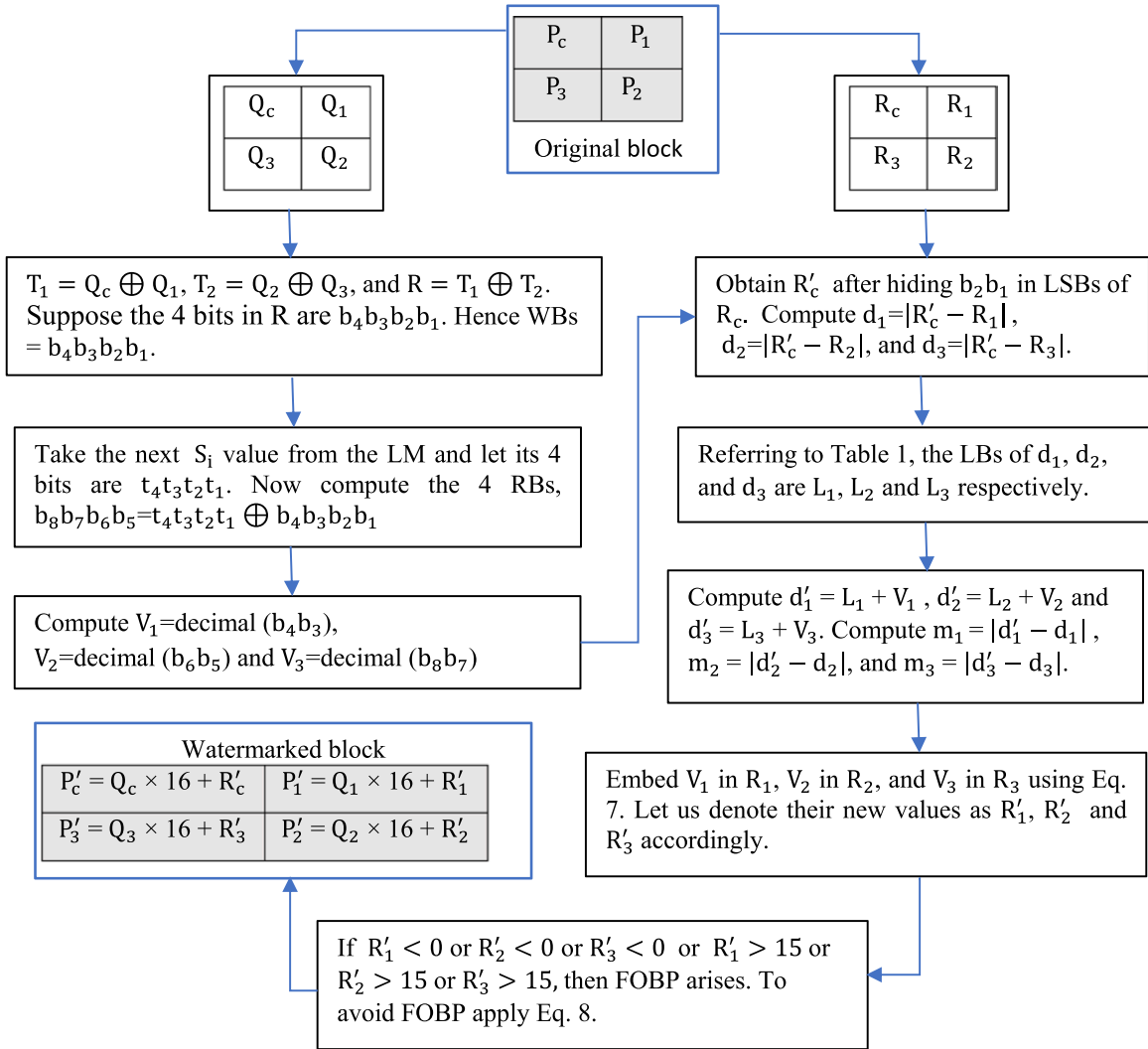


Fig. 3. Diagrammatic representation of watermark embedding procedure.

$$d_1 = |R'_c - R_1|, d_2 = |R'_c - R_2|, \text{ and } d_3 = |R'_c - R_3| \quad (5)$$

Consider the range Table 1. There are four ranges,  $\{0, 3\}$ ,  $\{4, 7\}$ ,  $\{8, 11\}$ , and  $\{12, 15\}$ . The  $d_1$  falls in a range of the Table 1, let the lower bound (LB) of that particular range is  $L_1$ . The  $d_2$  falls in a range of the Table 1, let the LB of that particular range is  $L_2$ . Similarly, the  $d_3$  falls in a range of the Table 1, let the LB of that particular range is  $L_3$ . Now compute three new difference values ( $d'_1$ ,  $d'_2$  and  $d'_3$ ) and  $m_1$ ,  $m_2$ , and  $m_3$  values using Eq. (6).

$$d'_i = L_i + V_i, \text{ and } m_i = |d'_i - d_i|, \text{ for } i = 1, 2, \text{ and } 3 \quad (6)$$

For  $i = 1, 2$ , and  $3$ , let us embed  $V_i$  in  $R_i$  using Eq. (7) and obtain the new values as  $R'_i$ .

$$R'_i = \begin{cases} R_i - m_i, & \text{if } R'_c \geq R_i, \text{ and } d'_i > d_i \\ R_i + m_i, & \text{if } R'_c < R_i, \text{ and } d'_i > d_i \\ R_i + m_i, & \text{if } R'_c \geq R_i, \text{ and } d'_i \leq d_i \\ R_i - m_i, & \text{if } R'_c < R_i, \text{ and } d'_i \leq d_i \end{cases} \quad (7)$$

After applying Eq. (7), for  $i = 1, 2$  and  $3$ , if for any  $i$ ,  $R'_i < 0$  or  $R'_i > 15$ , then FOBP arises. To avoid it we shall apply Eq. (8).

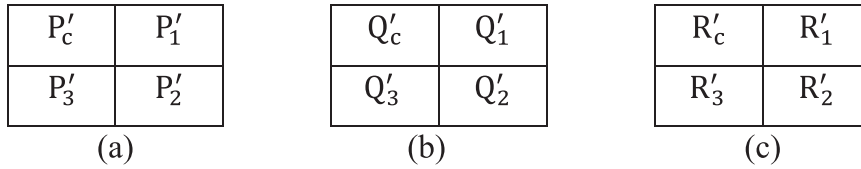


Fig. 4. (a) Watermarked pixel block, (b) Quotient block, and (c) Remainder block.

Table 2

Logics to correct  $R'_1$ ,  $R'_2$ , and  $R'_3$ .

L = {0, 4, 8, 12}, U = {3, 7, 11, 15} are two arrays, x, y, and T are variables		
Logic 1 to correct $R'_1$	Logic 2 to correct $R'_2$	Logic 3 to correct $R'_3$
<pre> For (x = 1; x ≤ 15; x++) begin   d<sub>1</sub><sup>*</sup> =  R'<sub>c</sub> - x    For (y = 1; y ≤ 4; y++)     begin       If (d<sub>1</sub><sup>*</sup> ≥ L[y] &amp;&amp; d<sub>1</sub><sup>*</sup> ≤ U[y])         T = d<sub>1</sub><sup>*</sup> - L[y]       end     end   If (T == V<sub>1</sub>)     begin       R'<sub>1</sub> = x       Break     end   end end </pre>	<pre> For (x = 1; x ≤ 15; x++) begin   d<sub>2</sub><sup>*</sup> =  R'<sub>c</sub> - x    For (y = 1; y ≤ 4; y++)     begin       If (d<sub>2</sub><sup>*</sup> ≥ L[y] &amp;&amp; d<sub>2</sub><sup>*</sup> ≤ U[y])         T = d<sub>2</sub><sup>*</sup> - L[y]       end     end   If (T == V<sub>2</sub>)     begin       R'<sub>2</sub> = x       Break     end   end end </pre>	<pre> For (x = 1; x ≤ 15; x++) begin   d<sub>3</sub><sup>*</sup> =  R'<sub>c</sub> - x    For (y = 1; y ≤ 4; y++)     begin       If (d<sub>3</sub><sup>*</sup> ≥ L[y] &amp;&amp; d<sub>3</sub><sup>*</sup> ≤ U[y])         T = d<sub>3</sub><sup>*</sup> - L[y]       end     end   If (T == V<sub>3</sub>)     begin       R'<sub>3</sub> = x       Break     end   end end </pre>

$$R'_i = \begin{cases} R'_c - V_i, & \text{if } (R'_c > R_i, \text{ and } R'_c - V_i \geq 0) \\ R'_c + V_i, & \text{if } (R'_c > R_i, \text{ and } R'_c + V_i < 0) \\ R'_c + V_i, & \text{if } (R'_c \leq R_i, \text{ and } R'_c + V_i \leq 15) \\ R'_c - V_i, & \text{if } (R'_c \leq R_i, \text{ and } R'_c - V_i > 15) \end{cases} \quad (8)$$

Now we shall compute the watermarked values of the pixels using Eq. (9). Thus, the watermarked pixels are  $P'_c$ ,  $P'_1$ ,  $P'_2$ , and  $P'_3$ . Fig. 3 represents a diagrammatical representation of the watermark embedding procedure.

$$P'_c = Q_c \times 16 + R'_c, \quad P'_1 = Q_1 \times 16 + R'_1, \quad P'_2 = Q_2 \times 16 + R'_2, \quad \text{and } P'_3 = Q_3 \times 16 + R'_3 \quad (9)$$

Fig. 3 represents a diagrammatical representation of the watermark embedding procedure.

### 3.3. Watermark extraction, tamper detection, and correction

The WI is scanned in raster scan order and divided into  $2 \times 2$  non-overlapped blocks. Consider a  $2 \times 2$  watermarked block shown in Fig. 4(a). The four pixels are  $P'_c$ ,  $P'_1$ ,  $P'_2$ , and  $P'_3$ . From the pixel block we shall create quotient block and remainder block using Eq. (10). Fig. 4(b) represents the quotient block, and Fig. 4(c) represents the remainder block.

$$Q'_c = P'_c \div 16, \quad R'_c = P'_c \bmod 16, \quad Q'_i = P'_i \div 16, \quad \text{and } R'_i = P'_i \bmod 16, \quad \text{for } i = 1, 2, \text{ and } 3 \quad (10)$$

Compute the Merkle root,  $R'$  using Eq. (11). Here,  $\oplus$  stands for bit-by-bit XOR operation. The 4 bits in  $R'$  are  $b'_4 b'_3 b'_2 b'_1$ .

$$T'_1 = Q'_c \oplus Q'_1, \quad T'_2 = Q'_2 \oplus Q'_3, \quad \text{and } R' = T'_1 \oplus T'_2 \quad (11)$$

Hence the computed WBs (CWBs) are  $b'_4 b'_3 b'_2 b'_1$ . Take the next  $S_i$  value from the LM and let its 4 bits are  $t_4 t_3 t_2 t_1$ . We obtain the 4

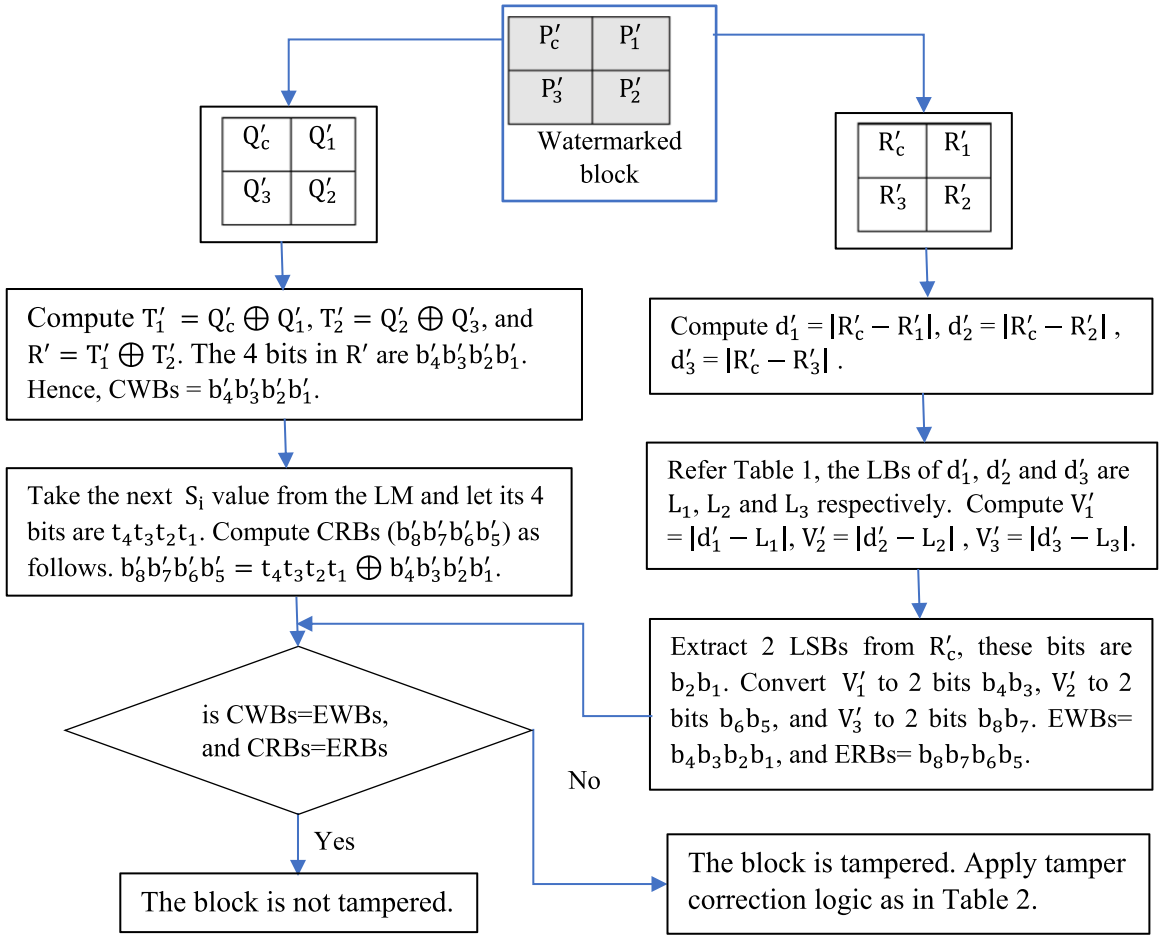


Fig. 5. Diagrammatical representation of watermark extraction procedure.

computed RBs (CRBs) as  $b'_8b'_7b'_6b'_5 = t_4t_3t_2t_1 \oplus b'_4b'_3b'_2b'_1$ .

For  $i = 1, 2$ , and  $3$  compute  $d'_i = |R'_c - R'_i|$ . The  $d'_i$  falls in a range of the Table 1, let the LB of that particular range is  $L_i$ . For  $i = 1, 2$ , and  $3$  compute  $V'_i = |d'_i - L_i|$ .

Extract 2 LSBs from  $R'_c$ , let these bits be represented as  $b_2b_1$ . Convert  $V'_1$  to 2 binary bits  $b_4b_3$ , convert  $V'_2$  to 2 binary bits  $b_6b_5$ , and convert  $V'_3$  to 2 binary bits  $b_8b_7$ . Now the extracted WBs (EWBs) are  $b_4b_3b_2b_1$ , and the extracted RBs (ERBs) are  $b_8b_7b_6b_5$ . If  $CWBs=EWBs$ , and  $CRBs=ERBs$ , then the block is not tampered. Otherwise, the block is tampered. Apply the logics given in Table 2 to correct  $R'_1$ ,  $R'_2$ , and  $R'_3$  if any tampering has been done.

Note that if any tampering has happened in  $R'_c$ , then the logic-1, logic-2, and logic-3 listed in Table 2 will not be effective. So, these logics are useful if and only if  $R'_c$  is not tampered. Fig. 5 represents a diagrammatical representation of the watermark extraction and tamper detection procedure.

#### 4. Results and discussion

This scheme has been practically implemented using MATLAB in a laptop computer equipped with windows 10 operating system. It has 1000 GB storage, i5 processor and 8 GB RAM. The images from SIPI image data-base are used for testing. Fig. 6 lists some 8-input images taken from SIPI data base. Fig. 7 lists the respective WIs.

The efficacy of watermarking techniques is measured by PSNR, HC, structural similarity (SSIM) index, accuracy (ACC), embedding time (EmT), and extraction time (ExT) [25]. The PSNR estimates the distortion in the WI using Eq. (12) [24], where  $P_{ij}$  and  $Q_{ij}$  represent the pixel values of the OI and WI respectively. The PSNR is measured in decibels (dB).

$$PSNR = 10 \times \log_{10} \frac{m \times n \times 255 \times 255}{\sum_{i=1}^m \sum_{j=1}^n (P_{ij} - Q_{ij})^2} \quad (12)$$

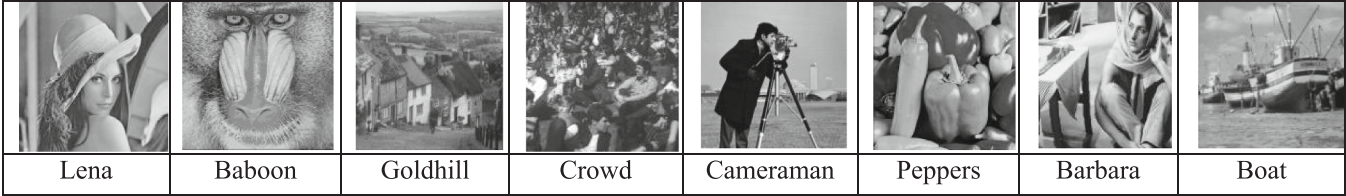


Fig. 6. OIs.

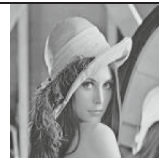







							
PSNR = 41.82 SSIM = 0.9682	PSNR = 42.01 SSIM = 0.9898	PSNR = 41.92 SSIM = 0.9785	PSNR = 42.05 SSIM = 0.9792	PSNR = 42.17 SSIM = 0.9668	PSNR = 41.85 SSIM = 0.9690	PSNR = 42.11 SSIM = 0.9813	PSNR = 42.25 SSIM = 0.9794

Fig. 7. WIs.



**Table 3**

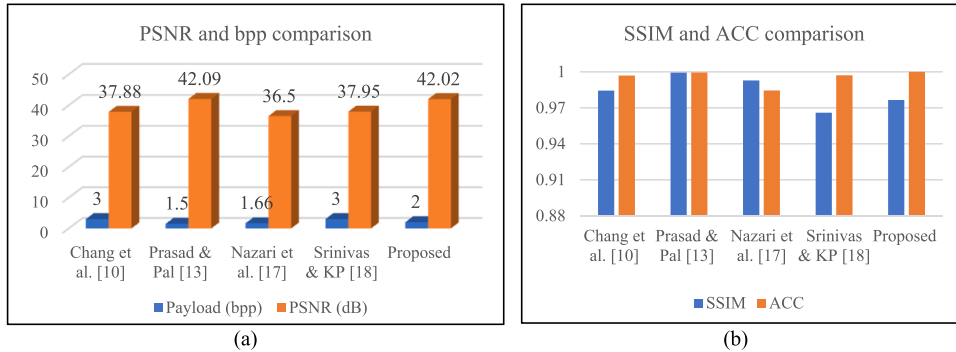
Efficacy measurement of the proposed technique.

Image (512 × 512)	HC (bpp)	PSNR (dB)	SSIM	ACC	EmT (secs)	ExT (secs)
Lena	2.0	41.82	0.9682	1	5.55	5.17
Baboon	2.0	42.01	0.9898	1	5.13	4.98
Goldhill	2.0	41.92	0.9785	1	4.48	4.11
Crowd	2.0	42.05	0.9792	1	4.46	4.12
Cameraman	2.0	42.17	0.9668	1	4.49	4.15
Pepper	2.0	41.85	0.9690	1	4.47	5.10
Barbara	2.0	42.11	0.9813	1	4.51	4.11
Boat	2.0	42.25	0.9794	1	4.52	4.11
Average	2.0	42.02	0.9765	1	4.70	4.48

**Table 4**

Comparison with existing techniques.

Technique	HC (bpp)	PSNR (dB)	SSIM	ACC
Chang et al. [10]	3.0	37.88	0.9844	0.9969
Prasad & Pal [13]	1.5	42.09	0.9994	0.9995
Nazari et al. [17]	1.66	36.50	0.9928	0.9845
Srinivas & KP [18]	3.0	37.95	0.9659	0.9971
Proposed	2.0	42.02	0.9765	1

**Fig. 8.** Comparison of PSNR, bpp, SSIM and ACC.

The HC is the number of watermarked bits camouflaged in various blocks throughout the image. SSIM measures the likeness between the WI and its OI. It is computed using Eq. (13) [13]. Here  $\bar{P}$  is the average pixel value of OI and  $\bar{Q}$  is the average pixel value of WI. The constants  $c_1$  and  $c_2$  are used to make the terms  $(\bar{P}^2 + \bar{Q}^2 + c_1)$  and  $(\sigma_x^2 + \sigma_y^2 + c_2)$  non-zero. The constant  $c_1 = (K_1 L)^2$ , where  $L = 255$  for gray image and  $K_1 \ll 1$ . Similarly,  $c_2 = (K_2 L)^2$ , where  $K_2 \ll 1$ . If the WI is very close to the OI, then SSIM value will be closure to 1.

$$SSIM = \frac{(2\bar{P}\bar{Q} + c_1)(2\sigma_{xy} + c_2)}{(\bar{P}^2 + \bar{Q}^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (13)$$

Here,  $\sigma_{xy} = \sum_{i=1}^m \sum_{j=1}^n (P_{ij} - \bar{P}) \times (Q_{ij} - \bar{Q})$ ,  $\sigma_x = \sum_{i=1}^m \sum_{j=1}^n (P_{ij} - \bar{P})^2$ ,  $\sigma_y = \sum_{i=1}^m \sum_{j=1}^n (Q_{ij} - \bar{Q})^2$ .

We can compute the number of true negative (TN) pixels, the number of true positive (TP) pixels, number of false positive (FP) pixels, and number of false negative (FN) pixels. Based on these values we can compute ACC using Eq. 14 [15]. TN is the number of pixels identified as not tampered and indeed they are not tampered. TP is the number of pixels identified as tampered and indeed they are tampered. FP is the number of pixels identified as tampered and indeed they are not tampered. FN is the number of pixels identified as not tampered and indeed they are tampered.

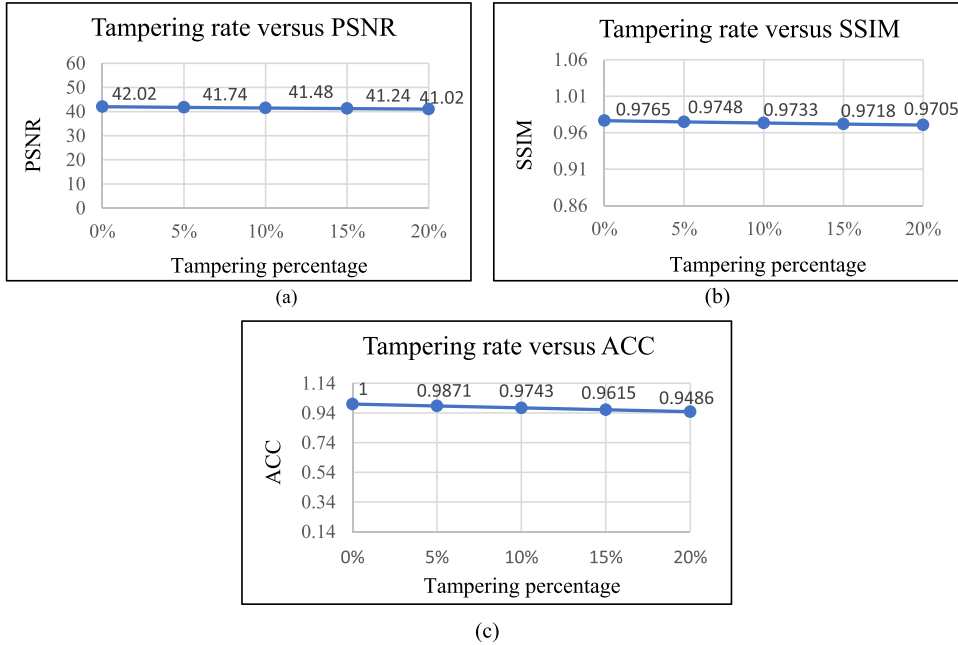
$$ACC = \frac{TN + TP}{FP + TN + FN + TP} \quad (14)$$

Table 3 represents the performance of the proposed watermarking technique in terms of various quality parameters. The HC is 2.0 bpp and PSNR is 42.02 dB. The SSIM is greater than 0.97, it implies that the WIs are very similar to their respective OIs. ACC value is 1, it implies that the tampered blocks can be identified accurately. The embedding time and extraction time of watermark is 4.7 s (secs)

**Table 5**

Average efficacy measurement at various tampering rates.

Tampering percentage	bpp	PSNR (dB)	SSIM	ACC
5%	2.0	41.74	0.9748	0.9871
10%	2.0	41.48	0.9733	0.9743
15%	2.0	41.24	0.9718	0.9615
20%	2.0	41.02	0.9705	0.9486

**Fig. 9.** PSNR, SSIM and ACC values at different tampering rates.

and 4.48 s respectively. All these parameter values are impressive, so it is conclusive that the proposed watermarking technique is efficient.

Table 4 represents a comparison of various efficacy parameters of the proposed technique with techniques of Chang et al. [10], Prasad & Pal [13], Nazari et al. [17] and Srinivas & KP [18]. These comparisons are also graphically represented by bar graphs in Fig. 8. From Table 4 and Fig. 8(a), it can be noticed that the PSNR of the proposed technique is 42.02 dB, it is as good as Prasad & Pal's technique and greater than all the remaining 3 existing techniques. The HC of Chang et al. and Srinivas & KP is larger than the proposed technique, but their PSNR is much lesser than the proposed technique. From Table 4 and Fig. 8(b) it can be noted that in proposed technique the SSIM value is 0.9765, it implies that the WIs and their OIs are very much similar. All the 4 existing techniques are showing good SSIM value in between 0.9659 and 0.9994. The ACC value of the proposed technique is 1, and ACC value of the existing techniques are in between 0.9845 and 0.9995.

Table 5 records the average efficacy parameter values at various tampering rates for the 8 sample images. Fig. 9(a) shows a plot of the PSNR values at different tampering rates. This plot indicates that even after increasing the tampering rates beyond 20%, the PSNR will sustain more than 40 dB. Fig. 9(b) shows a plot of the SSIM values at different tampering rates. This plot indicates that even after increasing the tampering rates beyond 20%, the SSIM will be maintained more than 0.97. Fig. 9(c) shows a plot of the ACC values at different tampering rates. This plot indicates that even after increasing the tampering rates beyond 20%, the ACC will sustain more than 0.90.

## 5. Conclusion

This article proposes a watermarking mechanism using Merkle tree, LM, and RVD. The image is scanned in raster scan order and divided into  $2 \times 2$  non-overlapped blocks. From each pixel of a block the decimal value for 4 MSBs, is called as quotient, and the decimal value for 4 LSBs is called as remainder. The 4 WBs are computed from 4 quotients as the root of Merkle tree. These 4 WBs are XORed with the 4 bits generated from LM sequence to generate the 4 RBs. These 4 WBs and 4 RBs are stored in 4 remainders by LSB alteration and RVD. While extracting the watermark, the tampered zones can be identified and a correction logic is applied to get the true value of the pixel. The quality of the proposed technique is estimated by various quality parameters. The values of these pa-

rameters are improved as compared to the related existing literature. The experimental results prove the ACC value is 1. It means that the tampered locations are identified accurately. Furthermore, it also maintains a balance between HC and PSNR. The PSNR value is 42.02 and HC value is 2.0. The SSIM value is 0.9765, it implies that the OI and WI are very much structurally similar. While masking the WBs and RBs in the remainders, the principle of differencing is used. Precautions are taken to check FOBP and to avoid it. The limitation of this proposed technique is that, the tamper correction is limited to the 4 LSBs of the three pixels  $P'_1$ ,  $P'_2$ , and  $P'_3$ . The tamper correction is not possible if the central pixel  $P'_c$  is tampered.

## Declaration of Competing Interest

There is no conflict of interest about the publication of this article. It is an independent work by the authors.

## Acknowledgment

This research is not sponsored by any government or private sponsoring agency or institution.

## References

- [1] N.B.A. Warif, A.W.A. Wahab, M.Y.I. Idris, R. Ramli, R. Salleh, S. Shamshirband, K.W.R. Choo, Copy-move forgery detection: survey, challenges, and future directions, *J. Netw. Comput. Appl.* 75 (2016) 259–278, <https://doi.org/10.1016/j.jnca.2016.09.008>.
- [2] L. Singh, A.K. Singh, P.K. Singh, Secure data hiding techniques: a survey, *Multimed. Tools Appl.* 79 (2020) 15901–15921, <https://doi.org/10.1007/s11042-018-6407-5>.
- [3] D. Singh, S.K. Singh, Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability, *J. Vis. Commun. Image Represent.* 38 (2016) 775–789, <https://doi.org/10.1016/j.jvcir.2016.04.023>.
- [4] F. Cao, B. An, J. Wang, D. Ye, H. Wang, Hierarchical recovery for tampered images based on watermarking self-embedding, *Displays* 46 (2017) 52–60, <https://doi.org/10.1016/j.displa.2017.01.001>.
- [5] S. Gull, N.A. Loan, S.A. Parah, J.A. Sheikh, G.M. Bhat, An efficient watermarking technique for tamper detection and localization of medical images, *J. Ambient Intell. Humaniz. Comput.* 11 (2020) 1799–1808, <https://doi.org/10.1007/s12652-018-1158-8>.
- [6] B. Feng, X. Li, Y. Jie, C. Guo, H. Fu, A novel semi-fragile digital watermarking scheme for scrambled image authentication and restoration, *Mob. Netw. Appl.* 25 (2020) 82–94, <https://doi.org/10.1007/s11036-018-1186-9>.
- [7] C. Qin, P. Ji, X. Zhang, J. Dong, J. Wang, Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy, *Signal Process.* 138 (2017) 280–293, <https://doi.org/10.1016/j.sigpro.2017.03.033>.
- [8] S. Rawat, B. Raman, A chaotic system based fragile watermarking scheme for image tamper detection, *Int. J. Electron. Commun. (AEU)* 65 (2011) 840–847, <https://doi.org/10.1016/j.aeeu.2011.01.016>.
- [9] M. Botta, D. Cavagnino, V. Pomponiu, A successful attack and revision of chaotic system based fragile watermarking scheme for image tamper detection, *Int. J. Electron. Commun. (AEU)* 69 (1) (2015) 242–245, <https://doi.org/10.1016/j.aeeu.2014.09.004>.
- [10] C.C. Chang, K.N. Chen, C.F. Lee, L.J. Liu, A secure fragile watermarking scheme based on chaos-and-hamming code, *J. Syst. Softw.* 84 (9) (2011) 1462–1470, <https://doi.org/10.1016/j.jss.2011.02.029>.
- [11] S. Trivedy, A.K. Pal, A logistic map-based fragile watermarking scheme of digital images with tamper detection, *Iran. J. Sci. Technol. Trans. Electr. Eng.* 41 (2017) 103–113, <https://doi.org/10.1007/s40998-017-0021-9>.
- [12] S. Prasad, A.K. Pal, A secure fragile watermarking scheme for protecting integrity of digital images, *Iran. J. Sci. Technol., Trans. Electr. Eng.* 44 (2020) 703–727, <https://doi.org/10.1007/s40998-019-00275-7>.
- [13] S. Prasad, A.K. Pal, A tamper detection suitable fragile watermarking scheme based on novel payload embedding strategy, *Multimed. Tools Appl.* 79 (2020) 1673–1705, <https://doi.org/10.1007/s11042-019-08144-5>.
- [14] S. Prasad, A.K. Pal, Hamming code and logistic-map based pixel-level active forgery detection scheme using fragile watermarking, *Multimed. Tools Appl.* 79 (2020) 20897–20928, <https://doi.org/10.1007/s11042-020-08715-x>.
- [15] A.K. Sahu, A logistic map based blind and fragile watermarking for tamper detection and localization in images, *J. Ambient Intell. Humaniz. Comput.* (2021), <https://doi.org/10.1007/s12652-021-03365-9>.
- [16] X. Tong, Y. Liu, M. Zhang, Y. Chen, A novel chaos-based fragile watermarking for image tampering detection and self-recovery, *Signal Process. Image Commun.* 28 (2013) 301–308, <https://doi.org/10.1016/j.image.2012.12.003>.
- [17] M. Nazari, A. Sharif, M. Mollaefar, An improved method for digital image fragile watermarking based on chaotic maps, *Multimed. Tools Appl.* 76 (2017) 16107–16123, <https://doi.org/10.1007/s11042-016-3897-x>.
- [18] K. Sreenivas, V. Kamakshiprasad, Improved image tamper localisation using chaotic maps and self-recovery, *J. Vis. Commun. Image Represent.* 49 (2017) 164–176, <https://doi.org/10.1016/j.jvcir.2017.09.001>.
- [19] X. Li, J. Li, B. Li, B. Yang, High fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion, *Signal Process.* 93 (1) (2013) 198–205, <https://doi.org/10.1016/j.sigpro.2012.07.025>.
- [20] F. Peng, X. Li, B. Yang, Improved PVO-based reversible data hiding, *Digit. Signal Process.* 25 (2014) 255–265, <https://doi.org/10.1016/j.dsp.2013.11.002>.
- [21] W. Hong, M. Chen, T.S. Chen, An efficient reversible image authentication method using improved PVO and LSB substitution techniques, *Signal Process. Image Commun.* 58 (2017) 111–122, <https://doi.org/10.1016/j.image.2017.07.001>.
- [22] G. Gao, X. Wan, S. Yao, Z. Cui, C. Zhou, X. Sun, Reversible data hiding with contrast enhancement and tamper localization for medical images, *Inf. Sci.* 385 (2017) 250–265, <https://doi.org/10.1016/j.ins.2017.01.009>.
- [23] N.N. Hurreh, S.A. Parah, J.A. Sheikh, Embedding in medical images: an efficient scheme for authentication and tamper localization, *Multimed. Tools Appl.* 79 (2020) 21441–21470, <https://doi.org/10.1007/s11042-020-08988-2>.
- [24] G. Swain, Two new steganography techniques based on quotient value differencing with addition-subtraction logic and PVD with modulus function, *Opt. - Int. J. Light Electron Opt.* 180 (2019) 807–823, <https://doi.org/10.1016/j.ijleo.2018.11.015>.
- [25] R. Sonar, G. Swain, Steganography based on quotient value differencing and pixel value correlation, *CAAI Trans. Intell. Technol.* (2021) 1–16, <https://doi.org/10.1049/cit2.12050>.