Journal of Information Security and Applications 72 (2023) 103407

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Separable reversible data hiding in an encrypted image using the adjacency pixel difference histogram

R. Anushiadevi [a], Rengarajan Amirtharajan [b,*]

[a] School of Computing, SASTRA Deemed University, Thirumalaisamudram, Thanjavur 613401, India
[b] School of Electrical and Electronics Engineering (SEEE), SASTRA Deemed University, Thirumalaisamudram, Thanjavur 613401, India

ARTICLE INFO

Keywords:
Reversible data hiding
Image encryption
Histogram shifting
Lossless data hiding
Steganography

ABSTRACT

Information security is a practice of encrypting data in movement and on hold, improving discretion and integrity. One can protect, encrypt and decrypt critical data in several ways. One of them is reversible data hiding in an encrypted image. The technique enables the user to encrypt images that need authentication and restore them to their original form of images. This technique returns a lossless image as an output making it the most suiTable for medical images and the military. Histogram shifting of pixel difference is an effectual reversible data hiding method in information security. Each image's pixel is encrypted when a user wants to safely store a digital image in an open environment like a cloud. The authentication or any other relevant information related to that image is embedded in the pixel difference histogram. The proposed approach's advantage is that the grayscale image transfer is carried out exceedingly safely, with near-zero correlation and Entropy closer to 8. The Peak Signal Noise Ratio (PSNR) for a directly decrypted image with an embedding capacity of 0.0807 bpp is 50.84 dB. Moreover, the secret and cover images are retrieved without error.

1. Introduction

Cryptography is a technique to make reliable and secure communication links between the sender and recipient to view or modify the data sent over the links. Encryption is a process that encodes a message or a file so that it can be readable only by a specific set of individuals. The process provides security to the messages or data and cannot be opened by anyone except the intended recipient or personnel. Images can also be encrypted in the same way as text without tampering with the structure or any loss of coloured pixels in a given image. But the above can tamper, so we need to hide secret information so that message is not easily decoded. The type of technique which is used to hide the data is known as steganography. Modern steganography uses concealment messages within the encrypted data or concealed along within some random data, which creates an unbreakable cipher like a one-time key that looks perfect, making the technique unable to tamper. Even images can be concealed with lower bits or random bits of 0′s and 1′s. While decrypting these images may result in lossy pixels, and there will be a loss of quality in the image on the receiver side. The loss of pixels resulting in less accuracy can be minimised through Reversible Data Hiding (RDH) technique. RDH is a type of data hiding process that utilises steganography to hide the host image's confidential data and be recovered precisely at the receiver's side.

The first RDH approach was introduced in the spatial domain [1], using modulo addition. Lossless compression is used in [2] to build free space for data embedding. These methods [1,2] embed only authentication-related information; much more data embedding is not possible; this is inappropriate for applications requiring a high embedding capacity. Therefore, a high embedding capacity algorithm [3] is introduced. After that, several methods emerged with a high embedding capacity. Difference Expansion (DE) [4] is one of the most common data embedding methods with high embedding capacity. The data are embedded by doubling the difference between adjacent pixels—another histogram shifting technique [5] where the information is hidden in the histogram's peak value. The pixel difference histogram is used in [6] to increase the embedding capacity instead of the pixel histogram. The image is divided into smaller blocks, and data are embedded in each block without changing the order of pixel values in [7,8]. A new two-dimensional histogram is generated in [9] to enhance the embedding. The prediction errors of each block are considered a set, and the sets are adjusted according to the data embedding based on flexible histogram modification [10]. Images can be embedded using the mean

* Corresponding author.
E-mail address: amir@ece.sastra.edu (R. Amirtharajan).

https://doi.org/10.1016/j.jisa.2022.103407

Available online 17 December 2022

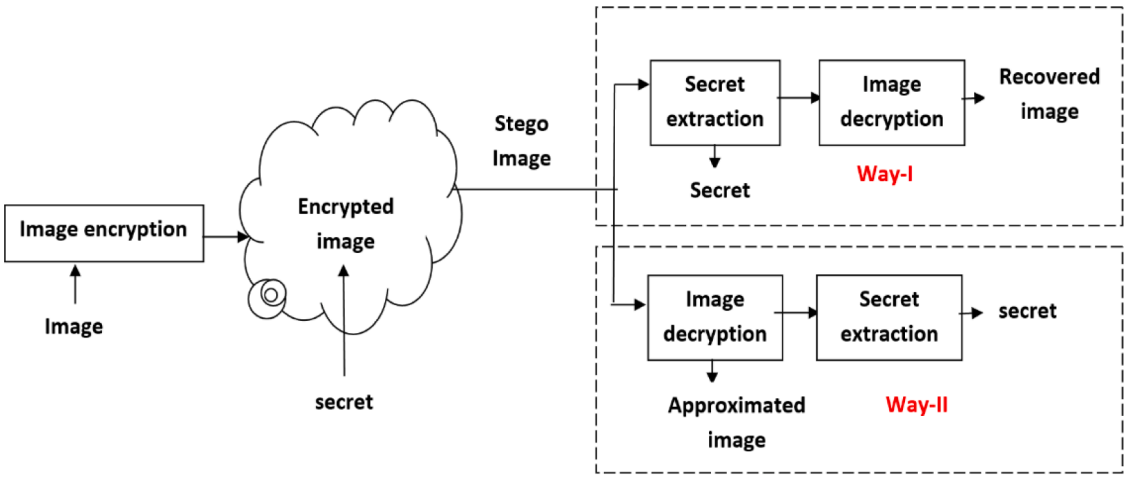

Fig. 1. SRDHEI-HPD Framework.

square error (MSE) of pixels and reduce the distorted pixel values [11]. Pixel values are expanded for data embedding in [12].

Each of the above-mentioned RDH methods can protect the secret but not cover it because; the plain images are used to embed the secrets. The person may depend on any third party for data embedding, such as the cloud service provider. In some cases, the cover must also be protected; in that circumstance, giving cover to a third party is hazardous. As a result, data embedding is performed on an encrypted image called reversible data hiding in an encrypted image (RDHEI). The RDHEI was first implemented by Zhang [13], in which the last three bits are flipped for data embedding, and the image is retrieved through a neighbourhood pixel correlation. The method [13] is enhanced in [14] by using the block smoothness and adjacency block boundary correlation. The extraction bit error rate of methods [13,14] is reduced by computing the mean difference of closer pixels [15]. Partially encrypted pixels are compressed to produce space intended for embedding, and the remaining pixels are not changed to retrieve the cover [16]. In [13–16], image decryption and secret extraction are done in an identical place; nevertheless, some applications require the decryption and extraction method to be performed separately, so separable reversible data hiding in an encrypted image (SRDHEI) is introduced.

The space for data embedding in an encrypted image is created in [17] to introduce the SRDHEI concept. The least significant bit of encrypted images is compressed to create room for embedding. Then secret extraction and image decryption can be done separately without affecting each other. The LSB compression in an encrypted image reduces the PSNR value of the deciphered image. Hence the deciphered images' PSNR value is increased by creating space for embedding before encryption in [18]. Separable reversible data hiding is performed using additive homeomorphism in methods [19,20], In addition to this, the differential expansion method is used in [19], and pixel order is used in [20]. The PSNR value of the directly decrypted image is increased in [21] by combining the integral wavelet transformation, histogram shifting and orthogonal decomposition to embed the secret. Mean value and pixel differences are used for embedding in [22,23]. Pixels are grouped as a reference and non-reference pixels; then-secret, extraction and cover recovery are made by non-reference pixel prediction error [24]. In [25], the clinical data are protected by block truncation coding. Homomorphic encryption and pixel prediction are used in [26], and prediction error estimation is used in [27] for image recovery and secret extraction. Two-level embedding is used in [28] to increase the embedding capacity. The SRDHEI is implemented in [29–32]. The embedding capacity of these methodologies is good; however, image encryption is only done by diffusion; hence it cannot provide great security for an image. In [33]the data embedding is implemented using Huffman code; in this methodology, transferring the Huffman codebook is not described, and the code varies for different images; therefore, embedding capacity may be high, but the pure embedding capacity is low. The method [34] is built for binary images. In method [35] the receiver extracts the Huffman coding sequence from the LSB plane; after extracting it the LSB plane can't be recovered in its whole; hence the image can't be fully restored.

In this proposed separable reversible data hiding method, the adjacency pixel differences histogram and homomorphic encryption are used to increase the PSNR value of the directly decrypted image. The image is encrypted by homomorphic encryption, and the data is embedded by plotting the histogram of the difference between adjacent pixels. Considering the peak bin on the histogram for embedding a secret message or value, the pixel differences after the peak are shifted accordingly to provide a better embedding area for a secret message. By performing the reverse process at the client-side, one can restore an original encrypted image without loss. This proposed method is named Separable Reversible Data Hiding in an Encrypted Image using Histogram of Pixel Difference (SRDHEI-HPD).

Using homomorphic encryption, the data hider can embed the secret without the knowledge of cover and secret. Moreover, an encrypted image is highly secured because every encrypted pixel value is added with the encrypted random numbers. As a result, the encrypted image correlation value is lower, the entropy values are equal to the image's bit depth, and the histogram distribution is uniform; thus, the statistical attack on the cover image is very difficult.

The significant contributions of the SRDHEI-HPD are given below

- The encrypted image is generated with less correlation value; entropy values are equal to the image's bit depth, and a uniform histogram distribution contributes to high security for the cover.
- The peak signal-to-noise ratio of the directly decrypted image is 50.84 dB for the average 0.0807bpp embedding capacity.
- The cover and secrets are retrieved with zero error.

This paper is worded as follows, Section 2 describes the proposed method, and Section 3 outlines the experiments' results. The paper is finally concluded in Section 4.

2. SRDHEI-HPD

The existing problem of a breach in data security has led to many financial losses and misleading information. The following system is suggested to forgo all the loopholes in the present-day system. The paper considers the reversible data hiding with histogram shifting of pixel differences. The approach is explained as follows, and the proposed method framework is shown in Fig. 1. The proposed method contains

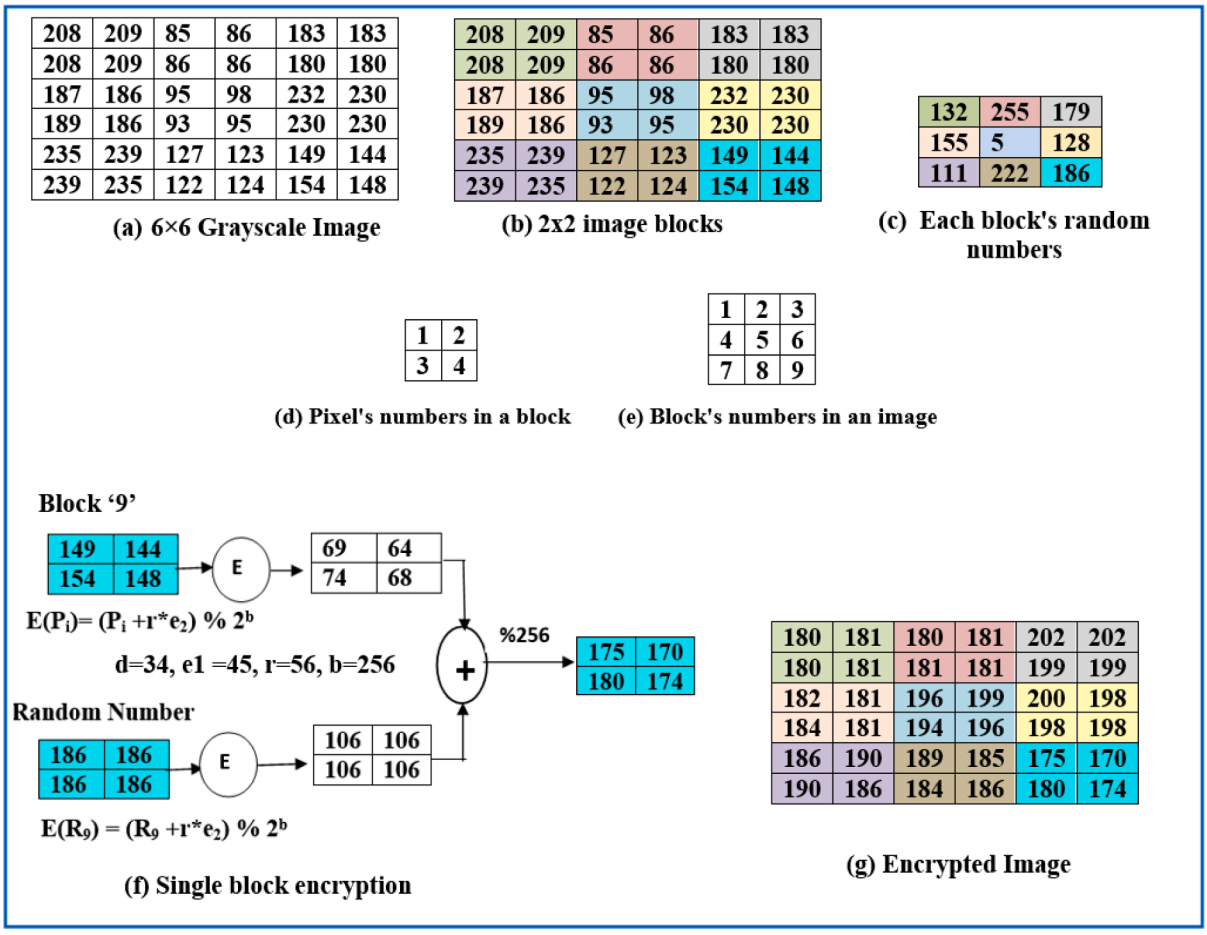

Fig. 2. Image encryption.

three phases. Image Encryption is carried out in the first phase. After encryption, anyone can embed the secret into an encrypted image in the second phase without knowing the original image. Finally, in the third phase, the secret can be mined, and the image can be recovered without any error.

2.1. Phase I - image encryption

The image is divided into 2 × 2 matrix blocks, each having 4-pixel values (P_1, P_2, P_3, P_4). Each pixel is encrypted with the sender's private key(r) and receiver's public key(e2) as shown in Eq. (1), For each block, one random number (R_X) is chosen, and that random number is also encrypted with same keys as shown in Eq. (2).

$$E(P_i) = (P_i + r \times e_2)\%2^b, i = 1,2,3,4 \tag{1}$$

$$E(R_X) = (R_X + r \times e_2)\%2^b \tag{2}$$

Where 'i' is a pixel number in a block, 'X' is a block number in an image and 'b' is a bit representation of an image. The receiver's public key 'e_2' is generated by using Eq. (3)

$$e_2 = \ e_1 \times d \tag{3}$$

where 'd' is the receiver's private key, e_1 is a random number. Secondly, each block is encrypted by adding encrypted pixel $E(P_i)$ with encrypted random number $E(R_X)$, as shown in Eq. (4). Finally, the mentioned additive homomorphic encryption is done as given below in Fig. 2.

$$E(P_{ix}) = E(P_i) + E(R_X), i = 1,2,3,4 \tag{4}$$

where' P_{ix}' is the i^{th}pixel value of block number X

2.2. Phase II - secret embedding

After combining all encrypted blocks, the encrypted image (EI) is obtained. Now data has to be embedded in an encrypted image by block histogram shifting. Firstly, the differences in pixel values in a block have to be found to plot the histogram. Three differences are found among the neighbouring pixels in a block(X), namely D_1, D_2, D_3. The pixel values are numbered in each block(X), as shown in Fig. 2. D_1 is the difference between the first pixel of a block and it's immediate below pixel. D_2 is the difference between the second pixel of a block and its immediate below the pixel. D_3 is the difference between the first and second pixel. The difference calculations are shown in Eqs. (5)–(7)

$$D_1 = P_1 - -P_3 \tag{5}$$

$$D_2 = P_2 - -P_4 \tag{6}$$

$$D_3 = P_1 - -P_2 \tag{7}$$

After finding differences, i.e. D_1, D_2, and D3, all blocks plot the histogram for individual differences. Find each histogram's peak value or peak bin and store it as T_1, T_2, T_3, respectively. The example calculation of T_1, T_2, and T_3 is shown in Fig. 3.

Before data can be embedded, the pixel value has to be shifted only if the pixel's difference is greater than the peak value. Whenever pixels difference meets a greater value than peak(T), pixel value has to be shifted by 2^n-1, where 'n' is the number of bits embedded per pixel. When pixel difference is equal to peak(T), then 'n' bits are embedded in it.The integer representation of the binary bits '00,' '01,' '10′,'11′ are 0,1,2 and 3, respectively. Binary bits are converted to integers and then added to the peak. For example, if $T_1 = 2$, the blocks whose D_1 difference

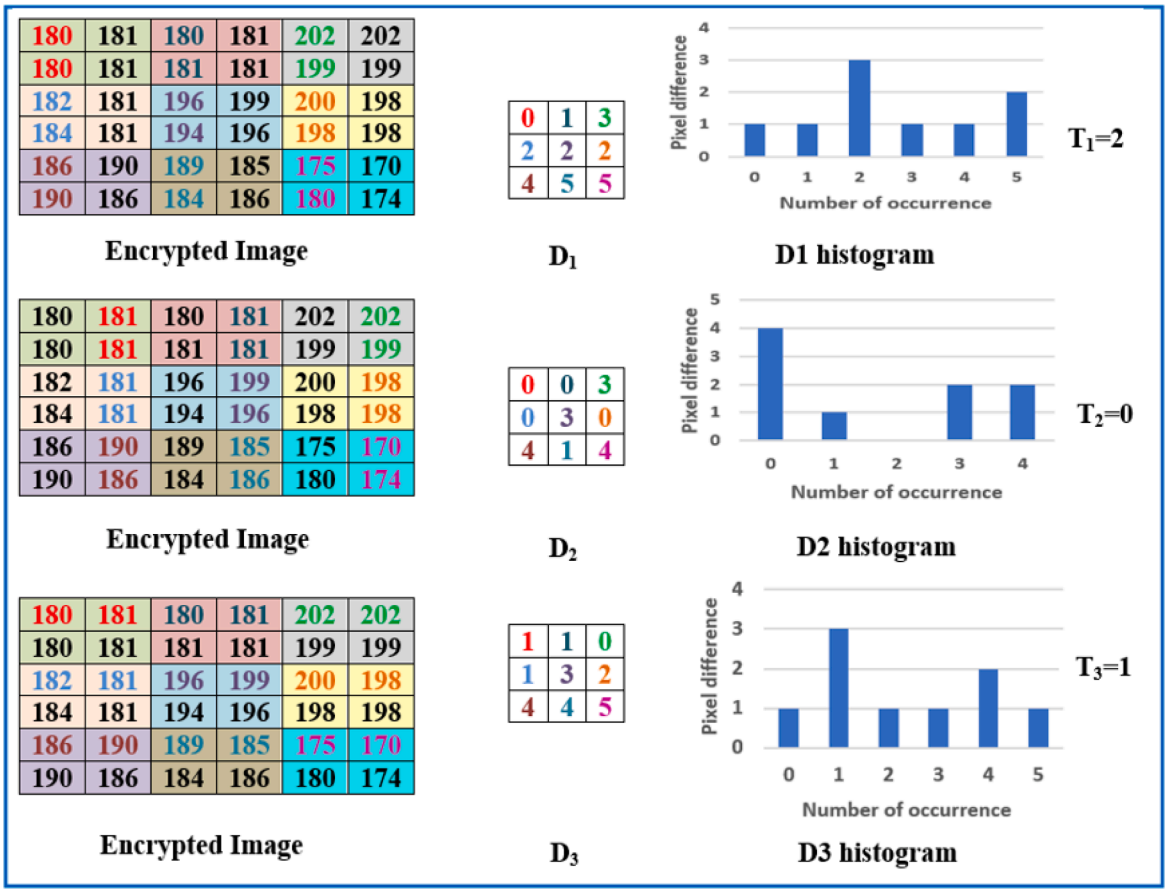

Fig. 3. Pixel difference histogram.

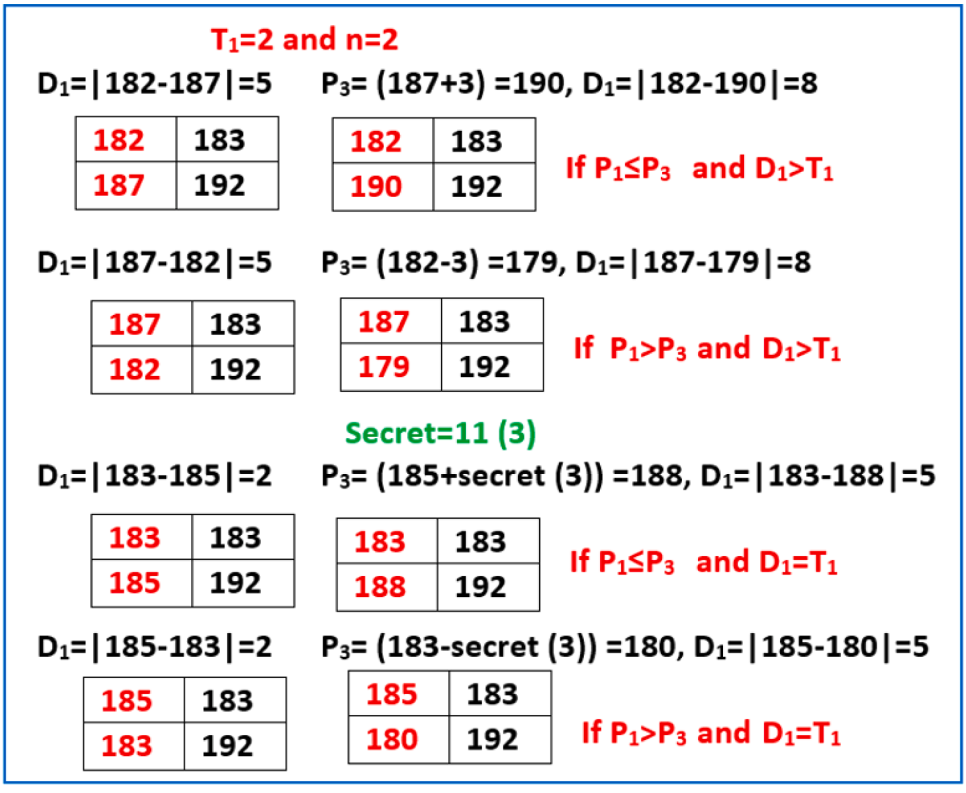

Fig. 4. D_1 histogram shifting and secret embedding.

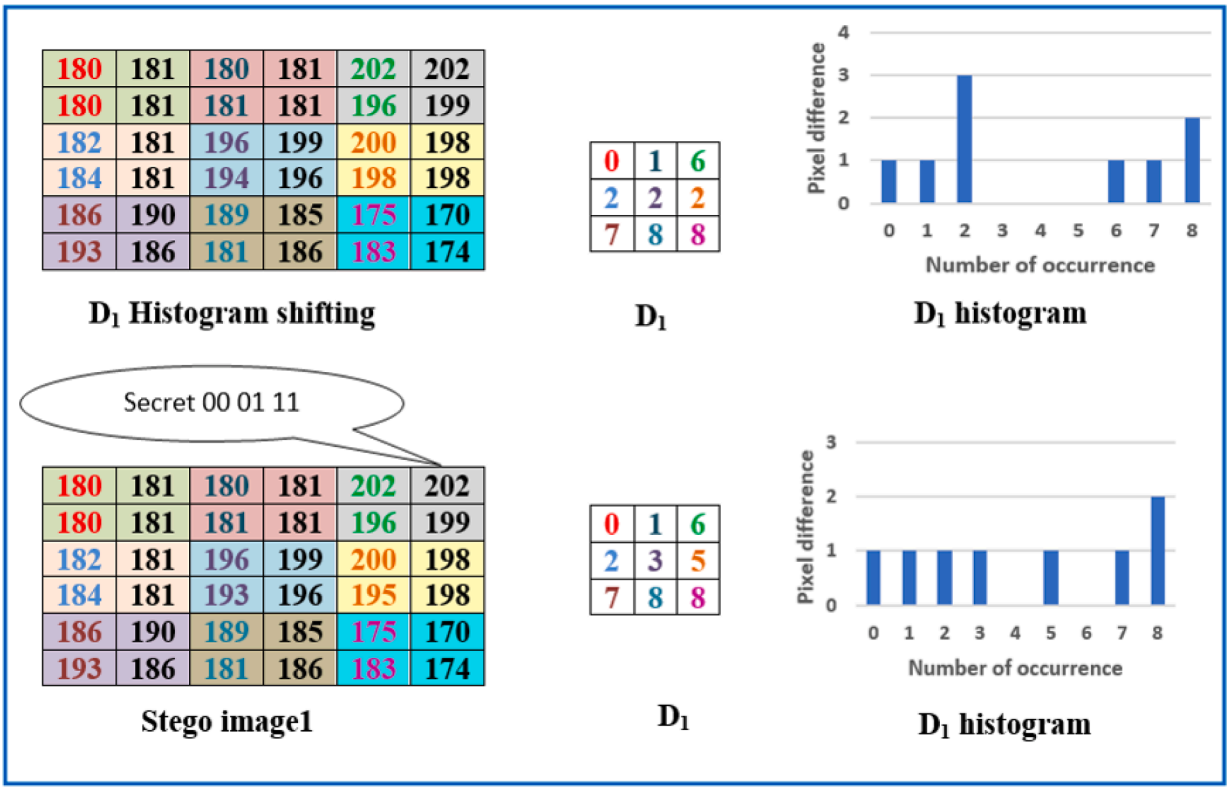

Fig. 5. Secret Embedding in D1 histogram.

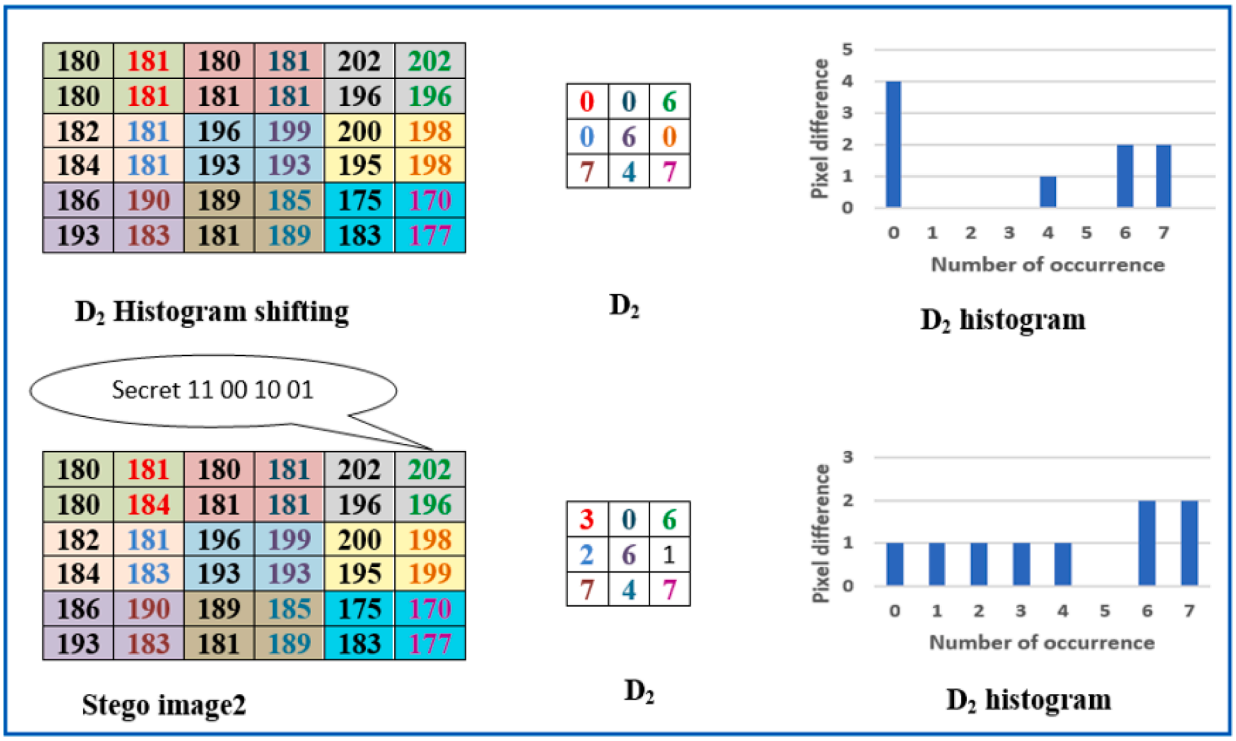

Fig. 6. Secret Embedding in D_2 histogram.

is greater than two are identified, then the 2^n-1 shift has to be taken in D_1. For difference D_1, pixel P_1 is fixed, and pixel P_3 has to update, as shown in Eq. (8). The example of histogram shifting and secret embedding is shown in Fig. 4.

$$P_3 = \begin{cases} P_3 + (2^n - 1) & if P_1 \le P_3 \quad and \quad D_1 > T_1 \\ P_3 - (2^n - 1) & if P_1 > P_3 \quad and \quad D_1 > T_1 \\ (P_3 + Secret) & if P_1 \le P_3 \quad and \quad D_1 = T_1 \\ (P_3 - Secret) & if P_1 > P_3 \quad and \quad D_1 = T_1 \end{cases} \tag{8}$$

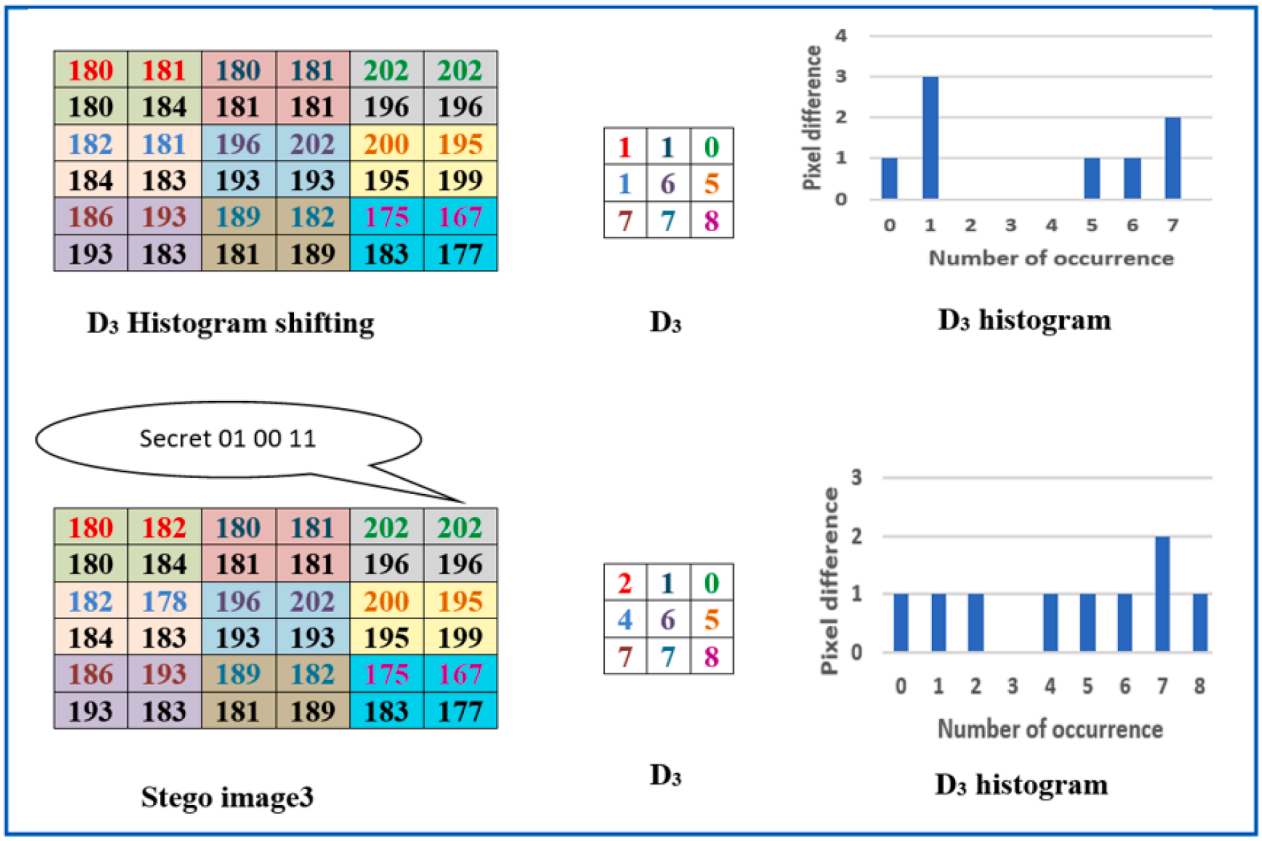

Fig. 7. Secret Embedding in D_3 histogram.

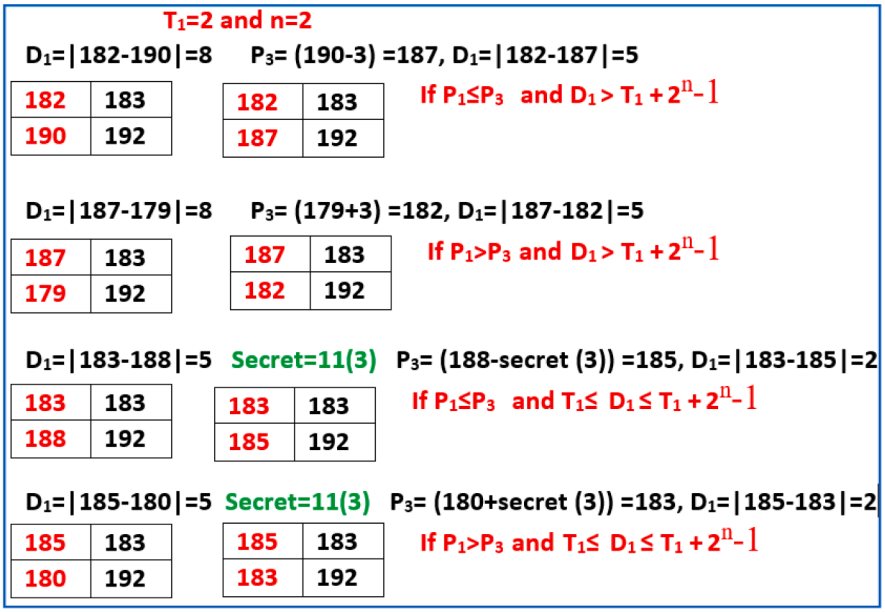

Fig. 8. Example of secret extraction from D_1 and histogram shifting.

Similarly, for D_2, pixel P_2 is fixed, and pixel P_4 has to be updated by following Eq. (9)

$$P_4 = \begin{cases} P_4 + (2^n - 1) & if P_2 \le P_4 \quad and \quad D_2 > T_2 \\ P_4 - (2^n - 1) & if P_2 > P_4 \quad and \quad D_2 > T_2 \\ (P_4 + Secret) & if P_2 \le P_4 \quad and \quad D_2 = T_2 \\ (P_4 - Secret) & if P_2 > P_4 \quad and \quad D_2 = T_2 \end{cases} \tag{9}$$

Similarly, for D_3, pixel P_1 is fixed, and pixel P_2 has to be updated by following Eq. (10)

$$P_2 = \begin{cases} P_2 + (2^n - 1) & if P_1 \le P_2 \quad and \quad D_3 > T_3 \\ P_2 - (2^n - 1) & if P_1 > P_2 \quad and \quad D_3 > T_3 \\ (P_2 + Secret) & if P_1 \le P_2 \quad and \quad D_3 = T_3 \\ (P_2 - Secret) & if P_1 > P_2 \quad and \quad D_3 = T_3 \end{cases} \tag{10}$$

The algorithm of data embedding is depicted in Algorithm 1, and an example of data embedding is shown in Figs. 5–7. The location map

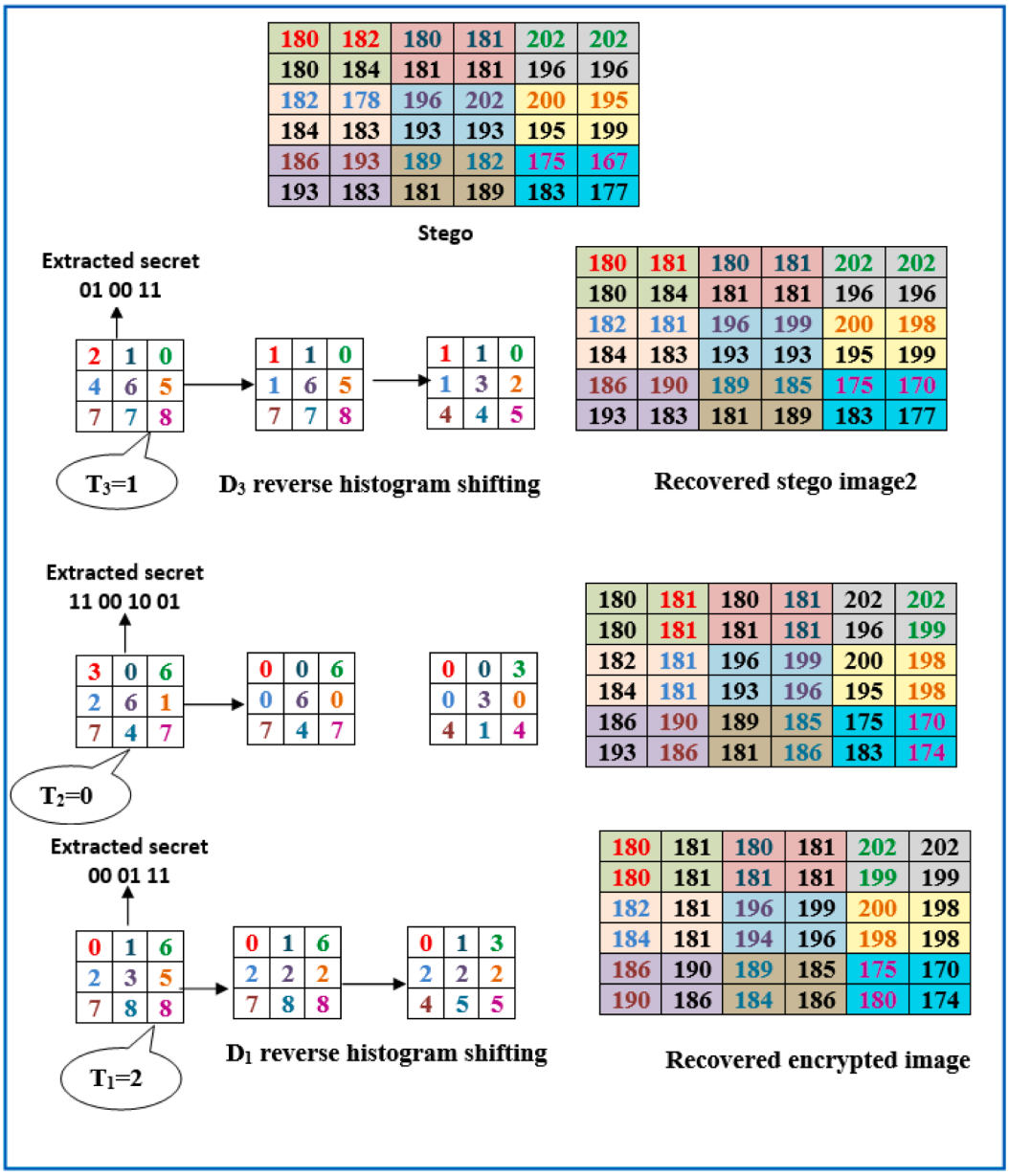

Fig. 9. Secret extraction and reverse histogram shifting.

identifies the overflow pixels.

2.3. Phase III - secret extraction and image recovery

Secret extraction and image recovery are carried out in two ways.

2.3.1. Way-I

Firstly, the D_3 is calculated for each block. If its value is equal to T_3 to T_3+2^n-1, then secrets are extracted from those D_3. The D_3 value is equal to T_3, then 0 is extracted, if it is equal to T_3+1, then secret 1 is extracted like it is T_3+2, then secret 2 is extracted, if it is T_3+2^n-1then secret 2^n-1 is extracted. After extracting the secret from D_3, make all those D_3 as T_3. If D_3 value is greater than T_3+2^n-1, then all D_3 are shifted back by 2^n-1. Secret extraction and histogram shifting are shown in Eqs. (11) and (12). The example of secret extraction and histogram shifting is shown in Fig. 8.

$$Secret = D_3 - T_3, if T_3 \le D_3 \le (T_3 + 2^n - 1) \tag{11}$$

$$P_2 = \begin{cases} P_2 - (2^n - 1) & if P_1 \le P_2 \quad and \quad D_3 > T_3 + 2^n - 1 \\ P_2 + (2^n - 1) & if P_1 > P_2 \quad and \quad D_3 > T_3 + 2^n - 1 \\ (P_2 - Secret) & if P_1 \le P_2 \quad and \quad T_3 \le D_3 \le T_3 + 2^n - 1 \\ (P_2 + Secret) & if P_1 > P_2 \quad and \quad T_3 \le D_3 \le T_3 + 2^n - 1 \end{cases} \tag{12}$$

$$Secret = D_2 - T_2, if T_2 \le D_2 \le (T_2 + 2^n - 1) \tag{13}$$

$$P_4 = \begin{cases} P_4 - (2^n - 1) & if P_2 \le P_4 \quad and \quad D_2 > T_2 + 2^n - 1 \\ P_4 + (2^n - 1) & if P_2 > P_4 \quad and \quad D_2 > T_2 + 2^n - 1 \\ (P_4 - Secret) & if P_2 \le P_4 \quad and \quad T_2 \le D_2 \le T_2 + 2^n - 1 \\ (P_4 + Secret) & if P_2 > P_4 \quad and \quad T_2 \le D_2 \le T_2 + 2^n - 1 \end{cases} \tag{14}$$

Similarly, the secrets are extracted from D_2 and histograms are shifted back, as shown in Eqs. (13) and (14), respectively. Similarly, the secrets are extracted from D_1 and histograms are shifted back, as shown in Eqs. (15) and (16), respectively.

$$Secret = D_1 - T_1, if T_1 \le D_1 \le (T_1 + 2^n - 1) \tag{15}$$

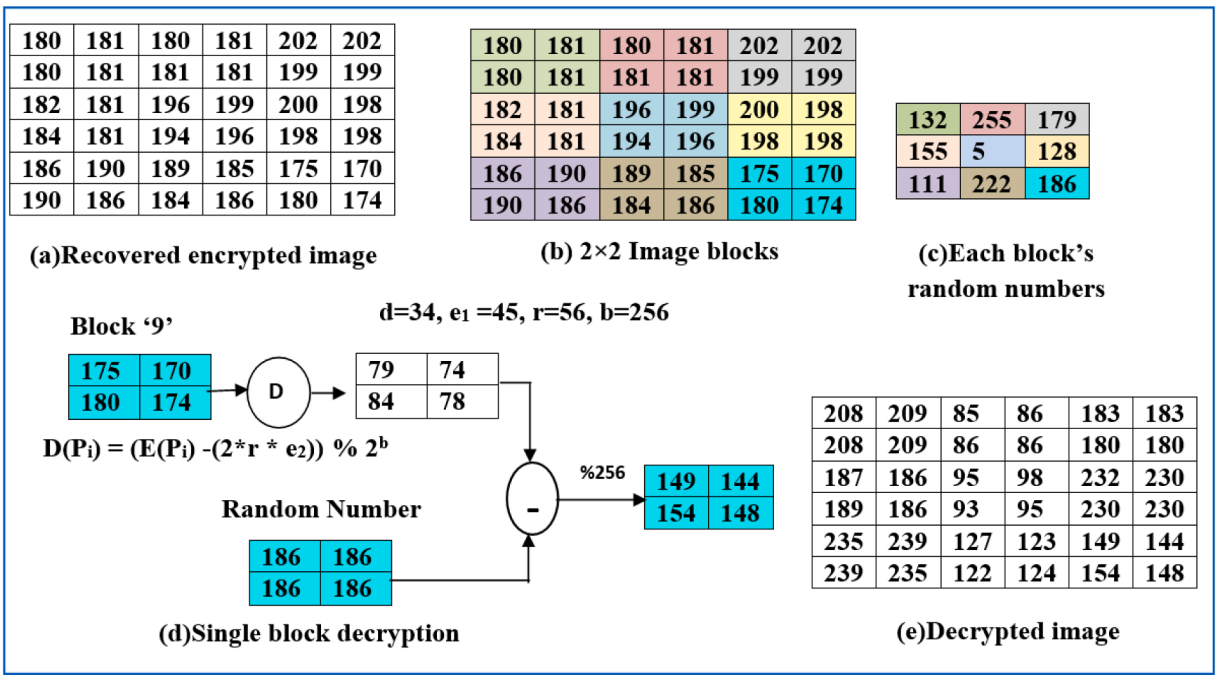

Fig. 10. Image decryption.

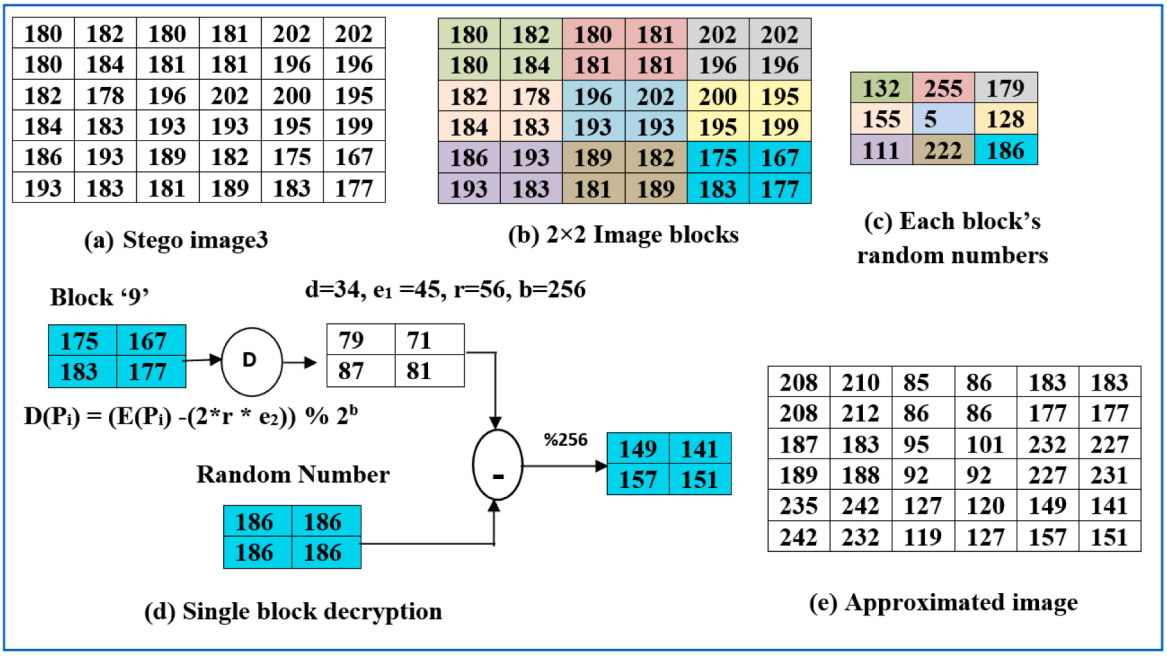

Fig. 11. Approximated image creation.

$$P_3 = \begin{cases} P_3 - (2^n - 1) & if P_1 \le P_3 \quad and \quad D_1 > T_1 + 2^n - 1 \\ P_3 + (2^n - 1) & if P_1 > P_3 \quad and \quad D_1 > T_1 + 2^n - 1 \\ (P_3 - Secret) & if P_1 \le P_3 \quad and \quad T_1 \le D_1 \le T_1 + 2^n - 1 \\ (P_3 + Secret) & if P_1 > P_3 \quad and \quad T_1 \le D_1 \le T_1 + 2^n - 1 \end{cases} \tag{16}$$

The data extraction algorithm is depicted in Algorithm 2; an example is shown in Fig. 9.

After extracting the data, each pixel value(P_i) of the encrypted image can be decrypted by Eqs. (17) and (18), and it is shown in Fig. 10.

$$D(P_i) = (E(P_i) - (2 \times r \times e_2))\%s2^b \tag{17}$$

$$P_i = D(P_i) - R_X \tag{18}$$

2.3.2. Way-II

An approximated image can be obtained from an original image without extracting the secret. Firstly, the encrypted image is decrypted to obtain the approximated image, as indicated in Fig. 11; then, the secret is extracted from an approximated image, as shown in Fig. 12.

3. Experimental results and discussions

The proposed method is implemented in MATLAB 2016b and tested

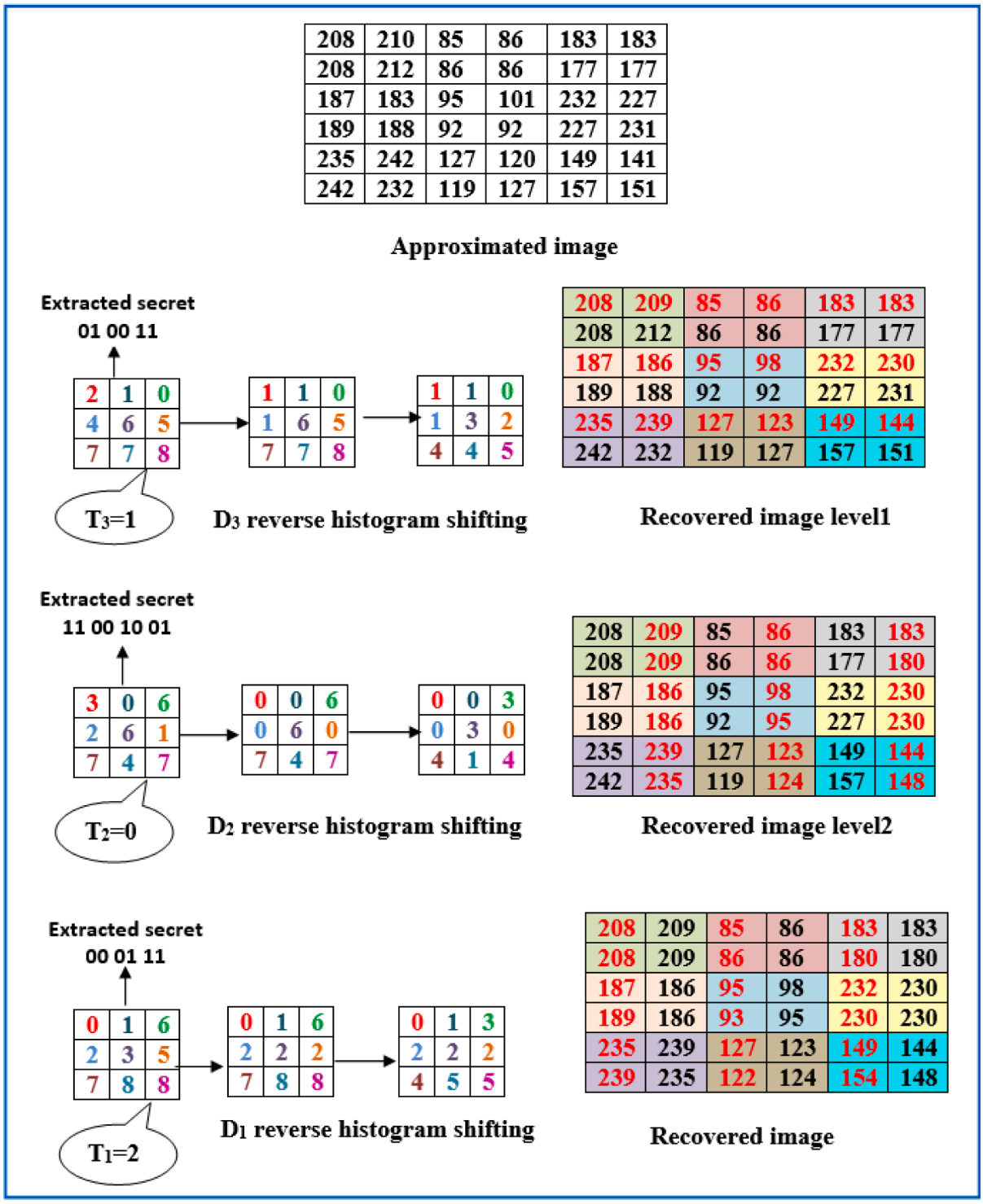

Fig. 12. Secret extraction.

with images of size 512 × 512 grayscale, military and medical images. Fig. 13 shows some of the images that have been tested.

3.1. Statistical analysis

The encrypted image security level is analysed using histogram, correlation coefficient, Entropy, and deviation from the encrypted images' ideality.

3.1.1. Histogram

Fig. 14 shows the histogram of the encrypted images, which is evenly distributed, so it is challenging to reveal its original content.

3.1.2. Correlation coefficient

The original image can be easily identified if the correlation among the neighbouring pixels is high. The correlation between the adjacent encrypted pixel value is low in the proposed method, indicating that the original image information is difficult to find. The correlation values of the encrypted images are indicated in Table 1, and this value is

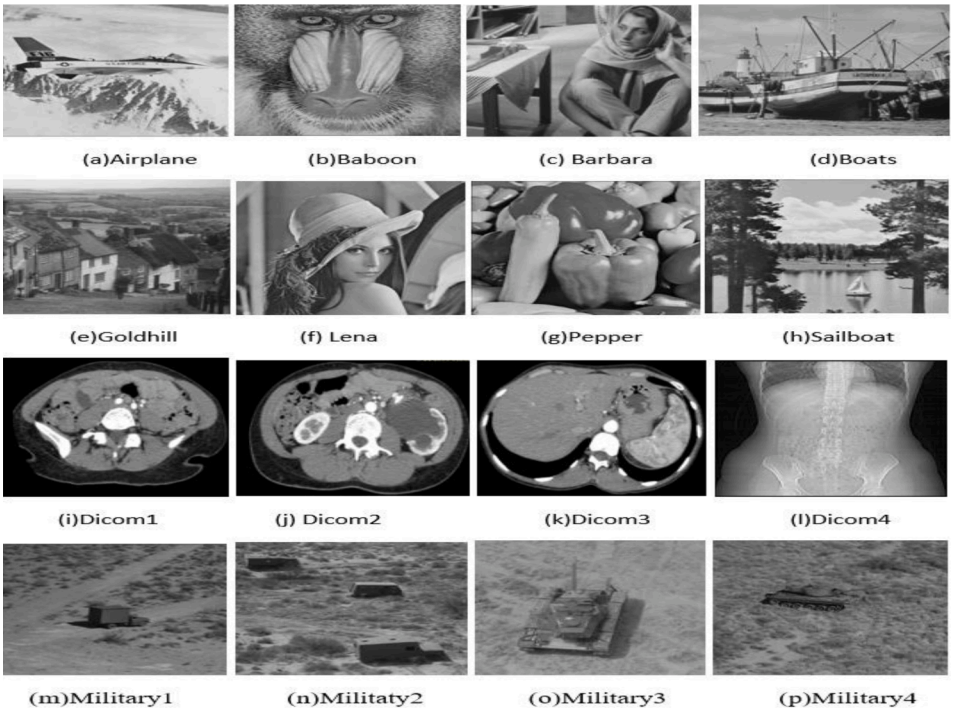

Fig. 13. Test images.

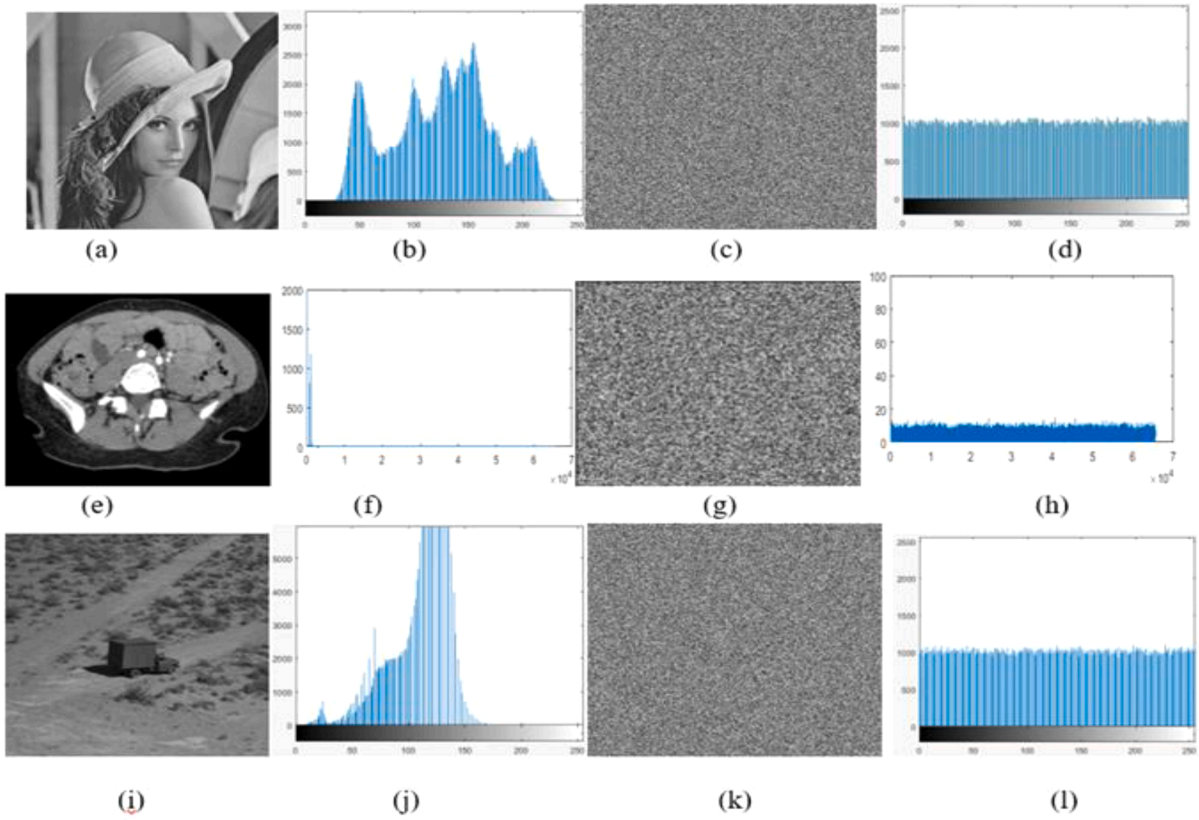

Fig. 14. Histogram analysis.

Table 1
Diagonal, horizontal and vertical correlation, Entropy and deviation from ideality of the encrypted images.

Image	Diagonal correlation	Horizontal correlation	Vertical correlation	Entropy	Deviation from Ideality
Airplane	0.0039	−0.0012	−0.0012	7.9993	0.0243
Baboon	0.0091	0.0008	−0.0015	7.9992	0.0241
Barbara	0.0047	0.0033	0.0004	7.9993	0.0243
Boats	0.0098	0.0011	0.0002	7.9994	0.0234
Goldhill	0.0053	−0.0038	0.0042	7.9993	0.0253
Lena	0.0093	−0.0012	0.0013	7.9992	0.0262
Pepper	0.0094	−0.0005	−0.0003	7.9993	0.0233
Sailboat	0.0079	0.0021	0.0009	7.9994	0.0235
Dicom1	0.0094	−0.0017	0.0028	15.8083	0.3892
Dicom2	0.0093	−0.0019	0.0034	15.8091	0.3897
Dicom3	0.0096	−0.0019	0.0031	15.8072	0.3920
Dicom4	0.0083	−0.0008	0.0042	15.8082	0.3906
Military1	0.0048	0.0009	0.0019	7.9991	0.0259
Military2	0.0048	−0.0034	0.0007	7.9992	0.0248
Military3	0.0074	−0.0025	0.0039	7.9993	0.0251
Military4	0.0051	−0.0032	0.0008	7.9991	0.0248

Table 2
PSNR, SSIM values for its corresponding EC, of directly decrypted images.

Images	EC (bpp)	PSNR(dB)	SSIM
Airplane	0.1595	50.83	0.9991
Baboon	0.0050	54.45	0.9992
Barbara	0.0570	50.05	0.9991
Boats	0.0989	50.34	0.9992
Goldhill	0.0509	50.02	0.9990
Lena	0.0911	50.33	0.9993
Pepper	0.1409	50.85	0.9987
Sailboat	0.0425	49.96	0.9998
Dicom1	0.2647	98.43	1.0000
Dicom2	0.2673	98.44	1.0000
Dicom3	0.2267	98.30	1.0000
Dicom4	0.2080	98.23	1.0000
Military1	0.07245	50.79	0.9992
Military2	0.02428	50.22	0.9995
Military3	0.0763	50.52	0.9992
Military4	0.0273	50.02	0.9995

calculated by Eq. (19)

$$CR = \frac{\sum_j (OI_j - OI_m)(EI_j - EI_m)}{\sqrt{\sum_j (OI_j - OI_m)^2}\sqrt{\sum_j (EI_j - EI_m)^2}} \tag{19}$$

whereOI_jandOI_m are *j*th pixel intensiveness of original image and mean intensity of the original image. EI_jandEI_m are *j*th pixel intensity of the encrypted image and mean intensity of the encrypted image

3.1.3. Entropy

Suppose the encrypted image entropy value is closer to the image's bit depth. In that case, finding the original image information is very difficult, and the Entropy of the proposed encrypted images is shown in Table 1. These values depict that the original image content of the encrypted image is difficult to detect. This entropy value is calculated by Eq. (20).

$$H(Sy) = -\sum_{j=1}^{MaxI} P(Sy_j)\log_2 P(Sy_j) \tag{20}$$

where 'MaxI' is the maximum intensity value of an image and $P(Sy_i)$is the probability of occurrence of the symbol Sy_i.

3.1.4. Deviation from ideality

The image histogram, which is evenly distributed to 100%, is called the ideal image. This metric decides how much the encrypted image deviates from the ideal image. If this value is low, it is difficult to find the original image contents. This value is calculated by Eqs. (21) and (22), and the deviation from ideality of the proposed encrypted images are shown in the Table 1.

$$H(Ideal) = \frac{512 \times 512}{2^b} \tag{21}$$

$$DI = \frac{\sum_{i=1}^{2^b} |H(EI_i) - H(Ideal)|}{512 \times 512} \tag{22}$$

where H(Ideal)and H(EI) are the histograms of an ideal and encrypted image

3.2. Pure embedding capacity

Secret bits are embedded by shifting the histogram of the image. After shifting the histogram and embedding the secret, some pixels may be out of range, i.e. less than zero or more than 2^b where b is the image's bit depth. These pixels are identified by the location map (LM). The pure embedding capacity (EC) is calculated by Eq. (23). Some of the sample test image's pure embedding capacities are shown in Table 2

$$EC = \frac{Totalnumberofbitsembeddedintheimage}{Totalpixelsintheimage} - LMsize(bits) \tag{23}$$

3.3. Directly decrypted image visual quality

The directly decrypted image visual quality is measured by using peak-signal-to-noise ratio and Structural Similarity Index Metric (SSIM).

3.3.1. Peak-signal-to-noise-ratio

This peak signal to noise ratio (PSNR) is applied to compare the

Table 3
EC and PSNR comparison with existing methods.

Images		[13]	[14]	[22]	[23] block size = 5	[23] blocksize = 9	Proposed
Airplane	EC	0.0039	0.0039	0.0039	0.1584	0.1692	0.1595
	PSNR	37.981	37.981	38.851	49.860	49.330	50.830
Baboon	EC	0.0039	0.0039	0.0039	–	0.0330	0.0050
	PSNR	37.934	37.934	37.913	–	50.190	54.450
Barbara	EC	0.0039	0.0039	0.0039	0.0550	0.0768	0.0570
	PSNR	37.925	37.925	38.706	50.530	49.670	50.050
Boats	EC	–	–	–	0.0475	0.0744	0.0989
	PSNR	–	–	–	46.730	45.710	50.340
Goldhill	EC	0.0039	0.0039	0.0039	0.0478	0.0738	0.0509
	PSNR	37.919	37.919	38.919	50.090	49.260	50.020
Lena	EC	0.0039	0.0039	0.0039	0.0900	0.1110	0.0911
	PSNR	37.904	37.904	39.863	49.800	49.210	50.330

Algorithm 1
Data Embedding.

1. Divide the image into 2 × 2 blocks
2. Loop in each block
2.1. $D_1 = P_1\text{-}P_3$
2.2. $D_2 = P_2\text{-}P_4$
2.3. $D_3 = P_1\text{-}P_2$
3. End Loop
4. Plot the histogram for individual differences(D_1,D_2,D_3)
5. Find peak bin in each histogram and store it as T_1, T_2, T_3 respectively
6. Loop in each block
6.1. If(D_1>T_1)
6.1.1. If (P_1<=P_3)
6.1.1.1. $P_3 = P_3+2^n\text{-}1$
6.1.2. Else
6.1.2.1. $P_3 = P_3\text{-}2^n\text{-}1$
6.1.3. End if
6.2. Else If(D_1==T_1)
6.2.1. If (P_1<=P_3)
6.2.1.1. $P_3 = P_3$+Binary_To_Decimal(n bits of secret)
6.2.2. Else
6.2.2.1. $P_3 = P_3$-Binary_To_Decimal(n bits of secret)
6.2.3. End If
6.3. End if
7. End Loop
8. Loop in each block
8.1. If(D_2>T_2)
8.1.1. If (P_2<=P_4)
8.1.1.1. $P_4 = P_4+2^n\text{-}1$
8.1.2. Else
8.1.2.1. $P_4 = P_4–2^n\text{-}1$
8.1.3. End if
8.2. Else If(D_2==T_2)
8.2.1. If (P_2<=P_4)
8.2.1.1. $P_4 = P_4$+Binary_To_Decimal(n bits of secret)
8.2.2. Else
8.2.2.1. $P_4 = P_4$-Binary_To_Decimal(n bits of secret)
8.2.3. End If
8.3. End if
9. End Loop
10. Loop in each block
10.1. If(D_3>T_3)
10.1.1. If (P_1<=P_2)
10.1.1.1. $P_2 = P_2+2^n\text{-}1$
10.1.2. Else
10.1.2.1. $P_2 = P_2–2^n\text{-}1$
10.1.3. End if
10.2. Else If(D_3==T_3)
10.2.1. If (P_1<=P_2)
10.2.1.1. $P_2 = P_2$+Binary_To_Decimal(n bits of secret)
10.2.2. Else
10.2.2.1. $P_2 = P_2$-Binary_To_Decimal(n bits of secret)
10.2.3. End If
10.3. End if
11. End Loop
12. End

image to the original image for quality measurement. If this ratio is greater than 30 dB, then the human eye cannot realize the changes in the image. This value can be calculated by Eqs. (24) and (25). Some of the directly decrypted image's PSNR values are shown in Table 2

$$PSNR = 10log_{10}\left(\frac{MaxI^2}{MSE}\right) \tag{24}$$

$$MSE = \frac{1}{512 \times 512}\sum_{i=1}^{512}\sum_{j=1}^{512}(OI(i,j) - DDI(i,j))^2 \tag{25}$$

3.3.2. Structural similarity index metric

This SSIM metric is used to evaluate the similarity between the original and the decrypted image; if this value is high, it is challenging to distinguish them. Therefore, this value is calculated by using Eq. (26). Some of the directly decrypted image's PSNR values are shown in Table 2.

Algorithm 2
Data extraction.

1. Divide the image into 2 × 2 blocks
2. Loop in each block
2.1. $D_1 = P_1\text{-}P_3$
2.2. $D_2 = P_2\text{-}P_4$
2.3. $D_3 = P_1\text{-}P_2$
3. End Loop
4. Loop in each block
4.1. Secret3= Secret3+D_3-T_3,if $T_3 \le D_3 \le (T_3+2^n\text{-}1)$
5. End Loop
6. Loop in each block
6.1. Secret2= Secret2+D_2-T_2,if $T_2 \le D_2 \le (T_2+2^n\text{-}1)$
7. End Loop
8. Loop in each block
8.1. Secret1= Secret1+D_1-T_1,if $T_1 \le D_1 \le (T_1+2^n\text{-}1)$
9. End Loop
10. Secret = Secret1+Secret2+Secret3
11. End

$$SSIM = \frac{(2\mu_{OI}\mu_{DDI} + cons_1)2\sigma_{OIDDI} + cons_2}{(\mu_{OI}^2 + \mu_{DDI}^2 + cons_1)\sigma_{OI}^2 + \sigma_{DDI}^2 + cons_2} \tag{26}$$

where μ_{OI}, μ_{DDI} are the average of the original and directly decrypted image, σ_{OI},σ_{DDI}arethe variance of the original and directly decrypted image, σ_{OIDDI}isthecovariance between original and directly decrypted image and $cons_1$,$cons_2$ are predefined constants

3.4. Performance comparison

The proposed method is compared with several existing methods of reversible data hiding in an encrypted image. This comparison is shown in Table 3. The pure embedding capacity and corresponding PSNR values of the methods [13,14,22] are much lower compared to the proposed method. The proposed method's pure embedding capacity and the method proposed in [23] with block size 5 are the same, but their PSNR values are low compared to the proposed method. And with block size 9, it has a high embedding capacity compared to the proposed method, but their PSNR values are low.

4. Conclusion

This paper proposes the separable reversible data hiding in an encrypted image using a histogram of the adjacent pixel difference and additive homomorphism. The proposed method was carried out in three phases: image encryption, Secret Embedding, Secret Extraction and Image Recovery. In the first phase, the image is encrypted using additively homomorphic encryption. In the second phase, the secret is embedded in the encrypted image by shifting the image histogram. In the third phase, the secret data and images are recovered. The statistical attack on the encrypted image is callous because the encrypted image's histogram is uniformly distributed, correlation values are all significantly less, entropy values are equal to the bit representation of the image and deviation from ideality is closer to zero. The data hider can embed the secret without knowing the original cover. The PSNR value of the directly decrypted image with an embedding capacity of 0.0807 bpp is 50.84 dB. The secret and cover images are retrieved without errors with SSIM above 0.9.

Funding

DST FIST funding (SR/FST/ET-I/2018/221(C)).

Ethical approval

This article does not contain any studies with human participants or animals performed by any authors.

CRediT authorship contribution statement

R. Anushiadevi: Conceptualization, Methodology, Software, Visualization, Validation, Investigation, Writing – original draft. **Rengarajan Amirtharajan:** Data curation, Methodology, Software, Supervision, Writing – review & editing.

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.

Data availability

Data will be made available on request.

Acknowledgments

Authors thank the Department of Science & Technology, New Delhi, for the FIST funding (SR/FST/ET-I/2018/221(C)). The authors wish to thank Intrusion LAB at the School of Electrical & Electronics Engineering, SASTRA Deemed University, for providing infrastructural support to carry out this research work.

References

[1] Honsinger, C.W., Jones, P.W., Rabbani, M., Stoffel, J.C., Eastman Kodak Co, 2001. *Lossless recovery of an original image containing embedded data*. U.S. Patent 6,278,791. https://www.google.co.in/patents/US6278791.
[2] Fridrich J, Goljan M, Du R. Invertible authentication. in. In: Proceedings of the security and watermarking of multimedia contents III; 2001. p. 197–209.
[3] Goljan M, Fridrich JJ, Du R. Distortion-free data embedding for images. in:. In: Proceedings of the international workshop on information hiding; 2001. p. 27–41.
[4] Tian J. Reversible data embedding using a difference expansion. IEEE Trans Circuits Syst Video Technol 2003;13:890–6. https://doi.org/10.1109/TCSVT.2003.815962.
[5] Ni Z, Shi YQ, Ansari N, Su W. Reversible data hiding. IEEE Trans Circuits Syst Video Technol 2006;16:354–62.
[6] Chang CC, Tai WL, Chen KN. Lossless data hiding based on histogram modification for image authentication. in:. In: EUC '08: proceedings of the 2008 IEEE/IFIP international conference on embedded and ubiquitous computing; 2008. p. 506–11.
[7] Li X, Li J, Li B, Yang B. High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. Signal Process 2013;93:198–205. https://doi.org/10.1016/j.sigpro.2012.07.025.
[8] Ou B, Li X, Zhao Y, Ni R. Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion. Signal Process Image Commun 2014;29:760–72.
[9] Gao E, Pan Z, Gao X. Reversible data hiding based on novel pairwise PVO and annular merging strategy. Inf Sci 2019;505:549–61.
[10] Wu H, Li X, Zhao Y, Ni R. Improved reversible data hiding based on PVO and adaptive pairwise embedding. J Real-Time Image Process 2019;16(3):685–95.
[11] Ying Q, Qian Z, Zhang X, Ye D. Reversible data hiding with image enhancement using histogram shifting. IEEE Access 2019;7:46506–21.
[12] Anushiadevi R, Praveenkumar P, Rayappan JBB, Amirtharajan R. Reversible data hiding method based on pixel expansion and homomorphic encryption. J Intell Fuzzy Syst 2020;39(3):2977–90.
[13] Zhang X. Reversible data hiding in encrypted image. IEEE Signal Process Lett 2011;18:255–8.
[14] Hong W, Chen TS, Wu HY. An improved reversible data hiding in encrypted images using side match. IEEE Signal Process Lett 2012;19:199–202.
[15] Liao X, Shu C. Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels. J Vis Commun Image Represent 2015;28:21–7.
[16] Zhang X, Qian Z, Feng G, Ren Y. Efficient reversible data hiding in encrypted images. J Vis Commun Image Represent 2014;25:322–8.
[17] Zhang X. Separable reversible data hiding in encrypted image. IEEE Trans Inf Forensics Secur 2012;7:826–32.
[18] Ma K, Zhang W, Zhao X, Yu N, Li F. Reversible data hiding in encrypted images by reserving room before encryption. IEEE Trans Inf Forensics Secur 2013;8:553–62.
[19] Shiu CW, Chen YC, Hong W. Encrypted image-based reversible data hiding with public key cryptography from difference expansion. Signal Process Image Commun 2015;39:226–33.
[20] Xiao D, Xiang Y, Zheng H, Wang Y. Separable reversible data hiding in encrypted image based on pixel value ordering and additive homomorphism. J Vis Commun Image Represent 2017;45:1–10. https://doi.org/10.1016/j.jvcir.2017.02.001.
[21] Xiong L, Xu Z, Shi YQ. An integer wavelet transform based scheme for reversible data hiding in encrypted images. Multidimens Syst Signal Process 2018;29:1191–202.
[22] Agrawal S, Kumar M. Mean value based reversible data hiding in encrypted images. Opt Int J Light Electron Opt 2017;30:922–34.
[23] Tang Z, Xu S, Ye D, Wang J, Zhang X, Yu C. Real-time reversible data hiding with shifting block histogram of pixel differences in encrypted image. J Real-Time Image Proc 2018. https://doi.org/10.1007/s11554-018-0838-0.
[24] Yu M., Liu Y., Sun H., Yao H., Qiao T.Adaptive and separable multiary reversible data hiding in encryption domain. J Image Video Proc. 2020, 16 (2020). https://doi.org/10.1186/s13640-020-00502-w.
[25] Bhardwaj R, Aggarwal A. Hiding clinical information in medical images: an enhanced encrypted reversible data hiding algorithm grounded on hierarchical absolute moment block truncation coding. Multidimens Syst Signal Process 2020;31(3):1051–74.
[26] Liu J, Zhao K, Zhang R. A fully reversible data hiding scheme in encrypted images based on homomorphic encryption and pixel prediction. Circuits Syst Signal Process 2020;39(7):3532–52.
[27] Malik A, Wang HX, Chen Y, Khan AN. A reversible data hiding in encrypted image based on prediction-error estimation and location map. Multimed Tools Appl 2020;79(17–18):11591–614.
[28] Anushiadevi R, Pravinkumar P, Rayappan JBB, Amirtharajan R. A high payload separable reversible data hiding in cipher image with good decipher image quality. J Intell Fuzzy Syst 2020;38(5):6403–14.
[29] Wu F, Zhou X, Chen Z, Yang B. A reversible data hiding scheme for encrypted images with pixel difference encoding. Knowl Based Syst 2021;234:107583.
[30] Long M, Zhao Y, Zhang X, Peng F. A separable reversible data hiding scheme for encrypted images based on Tromino scrambling and adaptive pixel value ordering. Signal Process 2020;176:107703.
[31] Liu ZL, Pun CM. Reversible image reconstruction for reversible data hiding in encrypted images. Signal Process 2019;161:50–62.
[32] Wu X, Qiao T, Xu M, Zheng N. Secure reversible data hiding in encrypted images based on adaptive prediction-error labeling. Signal Process 2021;188:108200.
[33] Chen CC, Chang CC, Chen K. High-capacity reversible data hiding in encrypted image based on Huffman coding and differences of high nibbles of pixels. J Vis Commun Image Represent 2021;76:103060.
[34] Ren H, Lu W, Chen B. Reversible data hiding in encrypted binary images by pixel prediction. Signal Process 2019;165:268–77.
[35] Wang X, Chang CC, Lin CC. Reversible data hiding in encrypted images with block-based adaptive MSB encoding. Inf Sci 2021;567:375–94 (Ny).