SECURITY AND PRIVACY

# Multi secret sharing based reversible data hiding using quadruple difference expansion

| | |
|---|---|
| Journal: | *Security and Privacy* |
| Manuscript ID | SPY-2023-0070 |
| Wiley - Manuscript type: | Research Article (Direct Via EEO) |
| Date Submitted by the Author: | 22-Mar-2023 |
| Complete List of Authors: | Agarwal, Ruchi; Babasaheb Bhimrao Ambedkar University<br>Kumar, Manoj; Babasaheb Bhimrao Ambedkar University |
| Keywords: | Communication Security, Information Security, Security in Business, Healthcare and Society |
| Abstract: | An important subcategory of reversible data hiding in images is reversible data hiding in encrypted images (RDH-EI). Numerous applications of this technology, including cloud computing, medical imaging, and military image transmission, have attracted much attention in recent years. In this paper, a recent RDH-EI algorithm termed shared one key (SOK) is improved by introducing modified difference expansion for encrypted images. Here, the data hider independently hides the data in an encrypted image without any need for a key. By using SOK properties, secure communication between the owner and receiver has been established. Also, in the proposed method, the key is shared between the owner and receiver utilizing the Diffie-Hellman key exchange algorithm, unlike the conventional SOK system where a private channel is required to share the key. The proposed modified difference expansion scheme improves the embedding capacity. When compared to other recent existing works, the proposed work's effectiveness can be seen. |
| | |

Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

# Multi secret sharing based reversible data hiding using quadruple difference expansion

Ruchi Agarwal | Manoj Kumar*

[1]Department of Computer Science, Babasaheb Bhimrao Ambedkar University, Uttar Pradesh, India

**Correspondence**
*Manoj Kumar. Email: mkjnuiitr@gmail.com

**Present Address**
Babasaheb Bhimrao Ambedkar University, Lucknow, (U.P.), India, (226025)

**Abstract**

An important subcategory of reversible data hiding in images is reversible data hiding in encrypted images (RDH-EI). Numerous applications of this technology, including cloud computing, medical imaging, and military image transmission, have attracted much attention in recent years. In this paper, a recent RDH-EI algorithm termed shared one key (SOK) is improved by introducing modified difference expansion for encrypted images. Here, the data hider independently hides the data in an encrypted image without any need for a key. By using SOK properties, secure communication between the owner and receiver has been established. Also, in the proposed method, the key is shared between the owner and receiver utilizing the Diffie-Hellman key exchange algorithm, unlike the conventional SOK system where a private channel is required to share the key. The proposed modified difference expansion scheme improves the embedding capacity. When compared to other recent existing works, the proposed work's effectiveness can be seen.

**KEYWORDS:**
Encryption, Difference expansion, Reversible data hiding, Multi secret sharing, Security

## 1 | INTRODUCTION

Advancements in multimedia technology offers many benefits in terms of storage and processing capabilities, meantime it also possess privacy and security concerns. Access to an infinite amount of information and entertainment, the exchange of various types of digital media, and other examples demonstrate the internet's growing popularity. With the advantage of smooth, procurable, and transferable communication, a huge breeding ground for illegal activities has also been spotted, resulting in a hefty issue of privacy. In the digital world, images are the most preferable source of information exchange, making them susceptible to attack by unintended and malicious users. Fortifying the issue, many solutions were proposed by researchers and scientists, including data hiding, cryptography, and steganography. All three terms mentioned are highly interconnected: steganography places special emphasis on the secret data, data hiding gives importance to the cover image, and in cryptography, the whole process appears in encoded form.

Data hiding[1] refers to the act of protecting the cover image while communicating it over the cloud by embedding secret data such that it is visually imperceptible. For the authentication and integrity of the image, when the secret data is embedded, sometimes the image gets slightly distorted. Military imaging, medical diagnosis, and law enforcement are the fields where a slight change or distortion in the original image is not admissible. To solve this problem, the concept of reversible data hiding has been introduced.

Further, reversible data hiding allows for the insertion of additional data, such as encryption keys or authentication information, into an image without altering its visual appearance. This additional data can be used to prevent unauthorized access to the sensor network, ensure the authenticity of the image data, and enable real-time monitoring and analysis. Additionally, the original image can be restored without any loss of quality or information, which helps to maintain the integrity of the image data. Overall, the use of reversible data hiding in visual sensor networks can improve the security and surveillance capabilities of the sensor network while also protecting the integrity of the image data.

Reversible data hiding (RDH) can be seen as a combination of data hiding and steganography where the hidden data and cover image both carry equal significance. In RDH, both the hidden data and the host image must be recovered losslessly. In[2], Barton proposed the first RDH technique. After this, many salient algorithms were proposed, among which difference expansion, histogram shifting, and lossless compression have vital importance. The ideology behind the concept of difference expansion[3] is to double the calculated difference of a pixel pair, such that the space for embedding a secret bit at the LSB position can be created. Ni et al.[4] proposed histogram shifting based on the fact that it creates space adjacent to the peak point for embedding secret data by slightly modifying other pixel values. In lossless compression[5], by modifying some of the pixels of the image, compressed secret data is embedded. Following these, many algorithms were proposed[6,7,8,9,10,11] which extended the basic idea of the schemes[3,4,5] with the motive to upgrade the embedding capacity with less distortion.

In recent years, RDH has been combined with encryption to give rise to a new direction of research in the encrypted domain, popularly known as reversible data hiding in encrypted images (RDH-EI). The main aim of RDH-EI is not limited to the perfect recovery of cover images and secret data; it also considers the privacy of cover images. There are three parties involved in RDH-EI: the owner of the content, the data hider, and the receiver. The data hider receives the encrypted image from the data owner and hides the secret data in it. Further, the data hider sends the embedded encrypted image to the receiver. After receiving the embedded encrypted image, the receiver can extract the data and reconstruct the image to its original state based on the availability of data hiding and encryption keys. RDH-EI is classified into two main scenarios: vacating the room after encryption (VRAE)[12,13,14,15,16] and vacating the room before encryption (VRBE)[17,18,19,20].

This paper presents a modified multi-secret sharing-based reversible data hiding scheme using a quadruple difference expansion. The proposed method creates the space before encryption, which is further used by the data hider to hide the data. The data owner doesn't share any keys with the data hider, unlike traditional schemes. The proposed work lowers the order of a polynomial by using one-fourth of the random values according to the available pixels in an image. Also, the embedding capacity is enhanced while maintaining security. The following are the primary contributions of the proposed work.

The proposed work utilizes the Diffie-Hellman key exchange algorithm for secure communication of random values between owner and receiver through a public network. The proposed work uses a multi-secret sharing scheme to encrypt the image using one random value over four pixels, which reduces the computational overhead. A modified difference expansion scheme is proposed for encrypted images, which raises embedding capacity from 0.5 bits per pixel in the traditional difference expansion scheme to 0.75 bits per pixel. The size of the location map is reduced by one-fourth, and the embedding capacity has maximised to three-fourths of the total image size. The secret data embedded is also randomly obtained using a pseudorandom number generator to verify the embedding capacity.

The Diffie-Hellman key exchange algorithm and multi-secret sharing are the most secure cryptographic methods, making the proposed technique more reliable than existing schemes.

The remainder of the paper is structured as follows: Related work is given in Section 2. Section 3 explores the proposed scheme along with a working example. Section 4 includes experimental results and analysis. Finally, Section 5 concludes the paper.

## 2 | RELATED WORK

The first novel RDH-EI algorithm in the field of VRAE was put forward by Zhang[12] in 2011. The algorithm utilizes the XOR operation for encryption; further, for embedding in images, data is segmented into blocks, and in each block, a bit is added in the form of secret data by flipping the three least significant bits (LSBs). In[13], Hong et al. proposed a modified version of Zhang's scheme by providing a better fluctuation function using the side match property with the aim of reducing the error rate. In[14], Zhang proposed a separable RDH in the encrypted domain. The key feature of this algorithm was its property of separability, i.e., that data extraction and recovery could be done independently. A novel technique[15] for calculating the complexity of an image by dividing it into non-overlapping blocks has been proposed; it also accounts for multiple neighboring pixels instead of only two adjacent pixels to increase the rate of embedding. Qian et al.[16] presented an RDH-EI scheme with the help of a distributed source encoding operation that enhances the payload.

Ma et al.[17] proposed the first algorithm in the category of VRBE. The algorithm first creates room for the embedding of secret data by embedding LSBs of some pixels into others with the help of existing algorithms, and then encrypts the image, followed by this secret data being embedded into the space created. Zhang et al.[18] discussed an improved reversible data hiding method based on the prediction error (PE) technique. Here, instead of encrypting the original pixel values, the predicted errors were encrypted by the encryption algorithm, and shifting of the histogram of predicted errors has been performed for embedding the data. Cao et al.[19] presented a patch-level sparse representation scheme for creating extra space for the embedding of secret data.

Later, many algorithms[20,21,22,23,24,25,26,27,28,33,34] were proposed using VRAE and VRBE concepts with the aim of securing the image with the maximum capacity. Nowadays, a new field[29,30,31] has been introduced by researchers based on the key setting and secret sharing for RDH-EI. The scheme has two aspects: sharing of an independent key (SIK) and sharing no key (SNK). Shiu et al.[29] presented a technique by applying Paillier homomorphic encryption to generate keys and encrypt the image; further, they utilized traditional difference expansion for embedding the data. Singh et al.[30] proposed a reversible data hiding scheme using Shamir secret sharing for image encryption and a combination of DWT and SVD for data embedding.

Most of the above discussed algorithms follow the concept of sharing an independent key, but some of them follow sharing no key. Chen et al.[31] recently introduced the mediator term sharing one key (SOK), which shares only one key with the receiver and no key with the data hider. The work in[31] can be summarised as follows: A secret sharing scheme is followed instead of homomorphic encryption; further, the concept of SOK provides the platform to directly embed data with the use of a well-known difference expansion technique without any knowledge of the key. Besides all these facts, the scheme needs an equal share of random variables as the number of pixels; however, they mentioned that the key size and random variables are compressed using lightweight cryptographic algorithms, but these can be further reduced. Two data hiding techniques were proposed in[32]. The first technique, homomorphic difference expansion in the encrypted domain (HDE-ED), performs the extraction of data from the retrieved image by utilizing the homomorphism property of the Chinese remainder theorem for secret sharing. The second technique, difference expansion in image shares (DE-IS), enables the extraction of data from the encrypted shares before image reconstruction as well as from the reconstructed image separately. The maximum embedding rates of the first and second methods could reach up to 0.500 and 0.4652 bits per pixel, respectively. In[35], the secret message was embedded using a novel adaptive pixel mapping technique. The encrypted image's pixel sub-arrays were analyzed to determine a new correlation measure between nearby pixels to restore the image.This framework ensures that the information in the original image is difficult to access by the data hider. However, it does not assure complete image recovery. Intending to eliminate the shortcomings i.e., low embedding capacity, less security, high order of polynomial and the size of the location map being equal to half the image size, which increase the computational cost of the scheme mentioned in[31], the authors proposed a more efficient version by applying the secret sharing concept.

## 3 | PROPOSED SCHEME

The proposed scheme needs some pre-processing before image encryption using the multi-secret sharing scheme at the owner's end. The embedding of the secret data in the encrypted image is performed at the data hider end, while message retrieval and image reconstruction are performed at the receiver end. In the proposed RDH-EI scheme, the data owner and receiver share a secret key (with the help of the Diffie-Hellman key exchange algorithm shown in Figure 1) that is only used to encrypt and decrypt the image, and they do not share any secrets with the data hider. A general sketch of the proposed scheme is shown in Figure 2.

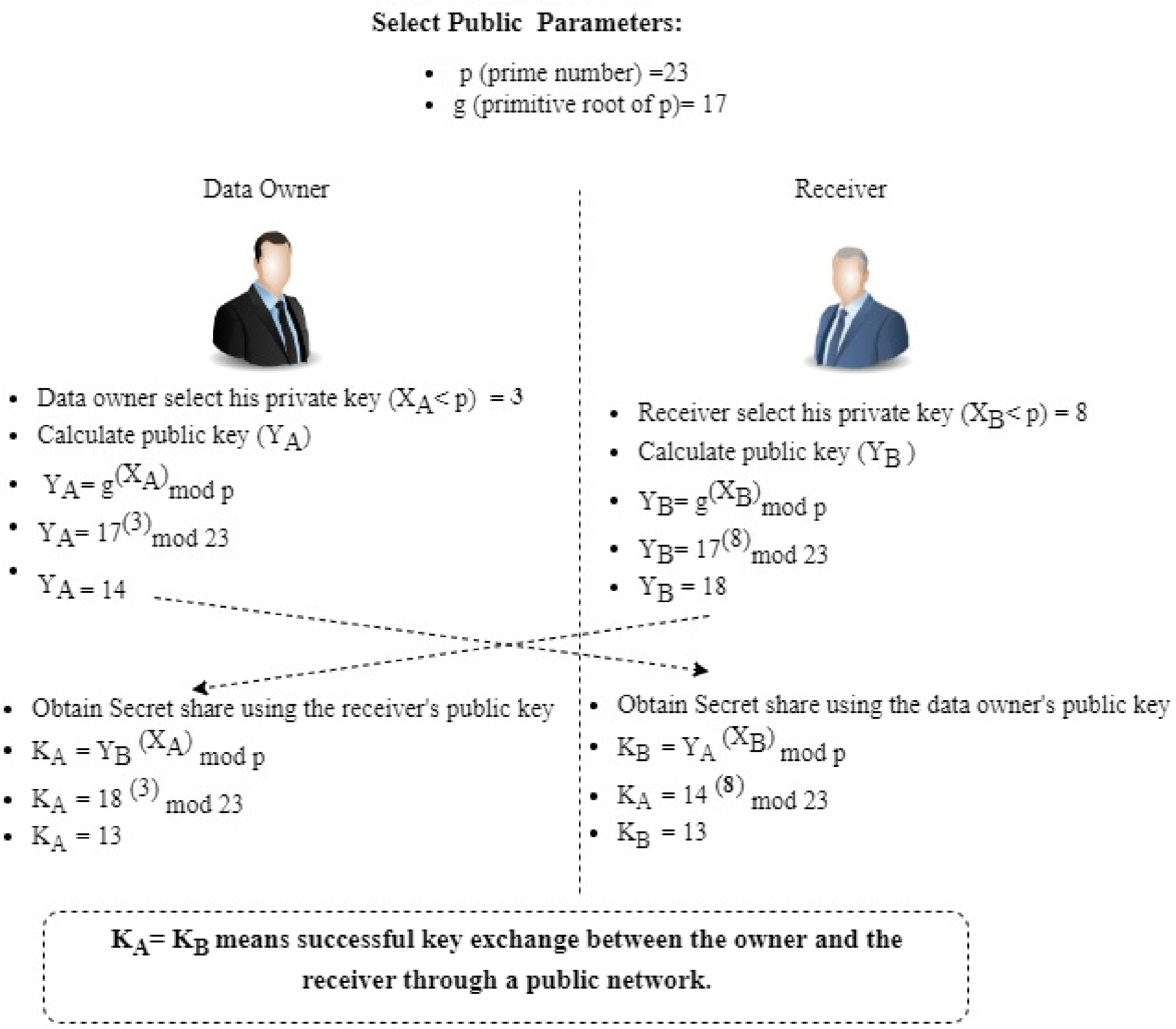

Select Public Parameters:
p (prime number) =23
g (primitive root of p)= 17
Data Owner
Receiver
Data owner select his private key ($X_A$< p) = 3
Calculate public key ($Y_A$)
$Y_A$= $g^{(X_A)}$mod p
$Y_A$= $17^{(3)}$mod 23
$Y_A$ = 14
Receiver select his private key ($X_B$< p) = 8
Calculate public key ($Y_B$ )
$Y_B$= $g^{(X_B)}$mod p
$Y_B$= $17^{(8)}$mod 23
$Y_B$ = 18
Obtain Secret share using the receiver's public key
$K_A$ = $Y_B$ $^{(X_A)}$ mod p
$K_A$ = 18 $^{(3)}$ mod 23
$K_A$ = 13
Obtain Secret share using the data owner's public key
$K_B$ = $Y_A$ $^{(X_B)}$ mod p
$K_A$ = 14 $^{(8)}$ mod 23
$K_B$ = 13
$K_A$= $K_B$ means successful key exchange between the owner and the receiver through a public network.

**Figure 1** An example of Diffie Hellman key exchange algorithm.

## 3.1 | Data Owner

The following steps are performed at data owner side:

**Step 1 Pre-processing:** To begin, all of the pixels of the image $I$ (say, $s$ pixels) are divided into quadruples, i.e., the $s$ pixels of the image are divided into $s/4$ quadruples. Calculate the differences ($d_i$) (where $i$ ranges from 1 to 3) of each pixel from the fourth pixel (the last pixel of a quadruple) while keeping the fourth pixel intact in each quadruple. Note that the differences ($d_i$) can be positive or negative. In a similar manner, differences for all the quadruples of the image are calculated. For pre-processed image ($I_p$) double the differences and add them to the value of the last pixel of each quadruple so that the first, second, and third pixel of each quadruple can be formed from pre-processed image. Similarly, process all the quadruples of the image to form a pre-processed image. The fourth pixel of each quadruple in the pre-processed image will remain the same as in the original image for the reference needed at the time of message extraction and image restoration.

**Step 2 Encryption:** The multi-secret scheme is used to encrypt $I_p$. But unlike[31], only one random value is generated per quadruple, reducing space and time complexities. The random values are generated through a pseudo-random number generator (PRNG) based on the working of Mersenne Twister (MT)[45] algorithm which is one of

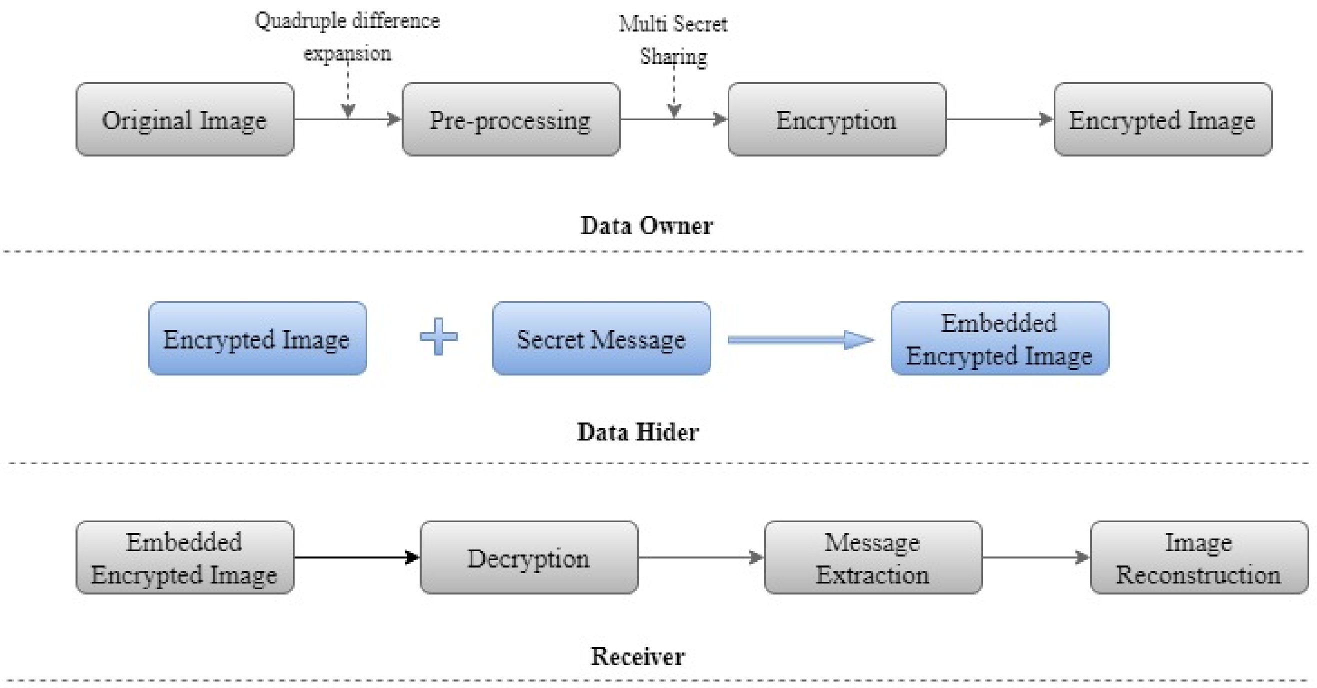


**Figure 2** Generalized schema of proposed work.

the most secure, attack resistant and powerful PRNG algorithms. In the case of the Mersenne Twister, the period is $2^{19937} - 1$, which is a very large number and is considered to be sufficiently long for many practical applications. These random values are shared with the receiver using the Diffie-Hellman key exchange algorithm over the field $\mathbb{F}$ in order to exchange the key over a public network as in[31] it was required to share the key through private channel. Random values $\mathcal{R}$ $(r_1, r_2, ......, r_{s/4})$ and distinct identity values $\mathcal{DI}$ for every pixel which are used to calculate the encrypted values belongs to $\mathbb{F}$.

Furthermore, we require that the field $\mathbb{F}$ be of prime numbers in order to maintain secret sharing security, so we choose 251 as the size of $\mathbb{F}$ that can be encoded to 8 bits.

To encrypt $t$ pixels of an image, where $t$ is a multiple of 4, $t/4$ random values are required. Thus, the generalised polynomial equation $Q(x)$ will have the following form:

$$Q(x) \;=\; s_1x^{(t+t/4-1)} + s_2x^{(t+t/4-2)} ... + s_tx^{(t+t/4-t} + R_1x^{t/4-1} + R_2x^{t/4-2} ... + R_{t/4}x^{t/4-t/4}(mod \quad \mathbb{F}) \quad (1)$$

The generation of polynomial for each quadruple is described below.

$$Q(x) = s_1x^4 + s_2x^3 + s_3x^2 + s_4x^1 + r_1x^0(mod\ \ 251) \quad (2)$$

Above equation make sure that $Q : \mathbb{F} \leftrightarrow \mathbb{F}$. Finally, the encrypted values are calculated as $(\mathcal{DI}_i, x)$, where $i$ varies from 1 to $t$. $Q(x) = Q(DI_i)$ is the encrypted value obtained after solving the polynomial, $Q(DI_i)$ for each quadruple are calculated as below :

$$Q(DI_1) \;=\; s_1(DI_1)^4 + s_2(DI_1)^3 + s_3(DI_1)^2 + s_4(DI_1)^1 + r_1(DI_1)^0(mod\ \ 251) \quad (3)$$
$$Q(DI_2) \;=\; s_1(DI_2)^4 + s_2(DI_2)^3 + s_3(DI_2)^2 + s_4(DI_2)^1 + r_1(DI_2)^0(mod\ \ 251) \quad (4)$$
$$Q(DI_3) \;=\; s_1(DI_3)^4 + s_2(DI_3)^3 + s_3(DI_3)^2 + s_4(DI_3)^1 + r_1(DI_3)^0(mod\ \ 251) \quad (5)$$
$$Q(DI_4) \;=\; s_1(DI_4)^4 + s_2(DI_4)^3 + s_3(DI_4)^2 + s_4(DI_4)^1 + r_1(DI_4)^0(mod\ \ 251) \quad (6)$$

### 3.2 | Data Hider

The data hider, upon receiving the encrypted image, performs the following steps:
**Step 1** Generate a random message of a binary number of three bits, with the fourth bit set to 0 for each quadruple. As a result, we have a secret message in the form $m_1, m_2, m_3, 0$. Now, again generate the polynomial using these secret message values and the distinct identity values used in encryption phase.
To encrypt $t$ secret bits, where $t$ is a multiple of 4, the generalised polynomial equation $M(x)$ would be in the following form:

$$M(x) = m_1 x^{(t+t/4-1)} + m_2 x^{(t+t/4-2)} ... + m_t x^{(t+t/4-t)} \tag{7}$$

**Step 2** Calculate the encrypted secret message values as $(DI_i, x)$, where $i$ varies from 1 to $t$ and $M(x) = M(DI_i)$. For each quadruple the encrypted message $M(x)$ is calculated below:

$$M(x) = m_1 x^4 + m_2 x^3 + m_3 x^2 + m_4 x^1 (mod\ \ \mathbb{F}) \tag{8}$$

The elaboration of above equation is mentioned below:

$$M(DI_1) = m_1(DI_1)^4 + m_2(DI_1)^3 + m_3(DI_1)^2 + m_4(DI_1)^1(mod\ \ 251) \tag{9}$$
$$M(DI_2) = m_1(DI_2)^4 + m_2(DI_2)^3 + m_3(DI_2)^2 + m_4(DI_2)^1(mod\ \ 251) \tag{10}$$
$$M(DI_3) = m_1(DI_3)^4 + m_2(DI_3)^3 + m_3(DI_3)^2 + m_4(DI_3)^1(mod\ \ 251) \tag{11}$$
$$M(DI_4) = m_1(DI_4)^4 + m_2(DI_4)^3 + m_3(DI_4)^2 + m_4(DI_4)^1(mod\ \ 251) \tag{12}$$

**Step 3** Now, for embedded encrypted image add the values generated from eq.1 and eq. 7 as:

$$E(x) = Q(x) + M(x)(mod\ \ \mathbb{F}) \tag{13}$$

where, $E$ is the embedded encrypted image.

### 3.3 | Receiver

On receiving the embedded encrypted image, the receiver first decrypts the image using a multi secret sharing scheme and the key shared through a public network, then extracts the message from it. The process includes the following steps:
**Step 1** Generate a four degree polynomial using the random values $\mathcal{R}$ and distinct identities $DI$. Let $E_1, E_2 ...... E_t$ be the pixels of embedded encrypted image $E$ having $t$ pixels. Then, for a quadruple the polynomial is generated as given below:

$$y_i(DI_i)^4 + y_{i+1}(DI_i)^3 + y_{i+2}(DI_i)^2 + y_{i+3}(DI_i)^1 + R\ \ (mod\ \ \mathbb{F}) = E_i \tag{14}$$

where, $i$ varies from 1 to 3 for a quadruple and $y_i$ are the pixels to be obtained from the eq. 14 using system of linear equations.
**Step 2** Extract the message, after calculating all the $y_i$, by solving the following equations:

$$m_i = y_{i+3} - y_i mod\ \ 2 \tag{15}$$
$$m_{i+1} = y_{i+3} - y_{i+1} mod\ \ 2 \tag{16}$$
$$m_{i+2} = y_{i+3} - y_{i+2} mod\ \ 2 \tag{17}$$
$$m_{i+3} = y_{i+3} - y_{i+3} mod\ \ 2 \tag{18}$$

where, $m_i$ are the bits of extracted message.
**Step 3** Reconstruct the original pixels $Rec_i$ of quadruple by performing the following:

$$Rec_i = (y_{i+3} - y_i + m_i)/2 \tag{19}$$
$$Rec_{i+1} = (y_{i+3} - y_{i+1} + m_{i+1})/2 \tag{20}$$
$$Rec_{i+2} = (y_{i+3} - y_{i+2} + m_{i+2})/2 \tag{21}$$
$$Rec_{i+3} = (y_{i+3} - y_{i+3} + m_{i+3})/2 \tag{22}$$

Put $i = 1$, in above eqs. to get all the recovered pixels from a quadruple. Perform the same step with all the quadruples to recover the whole image.

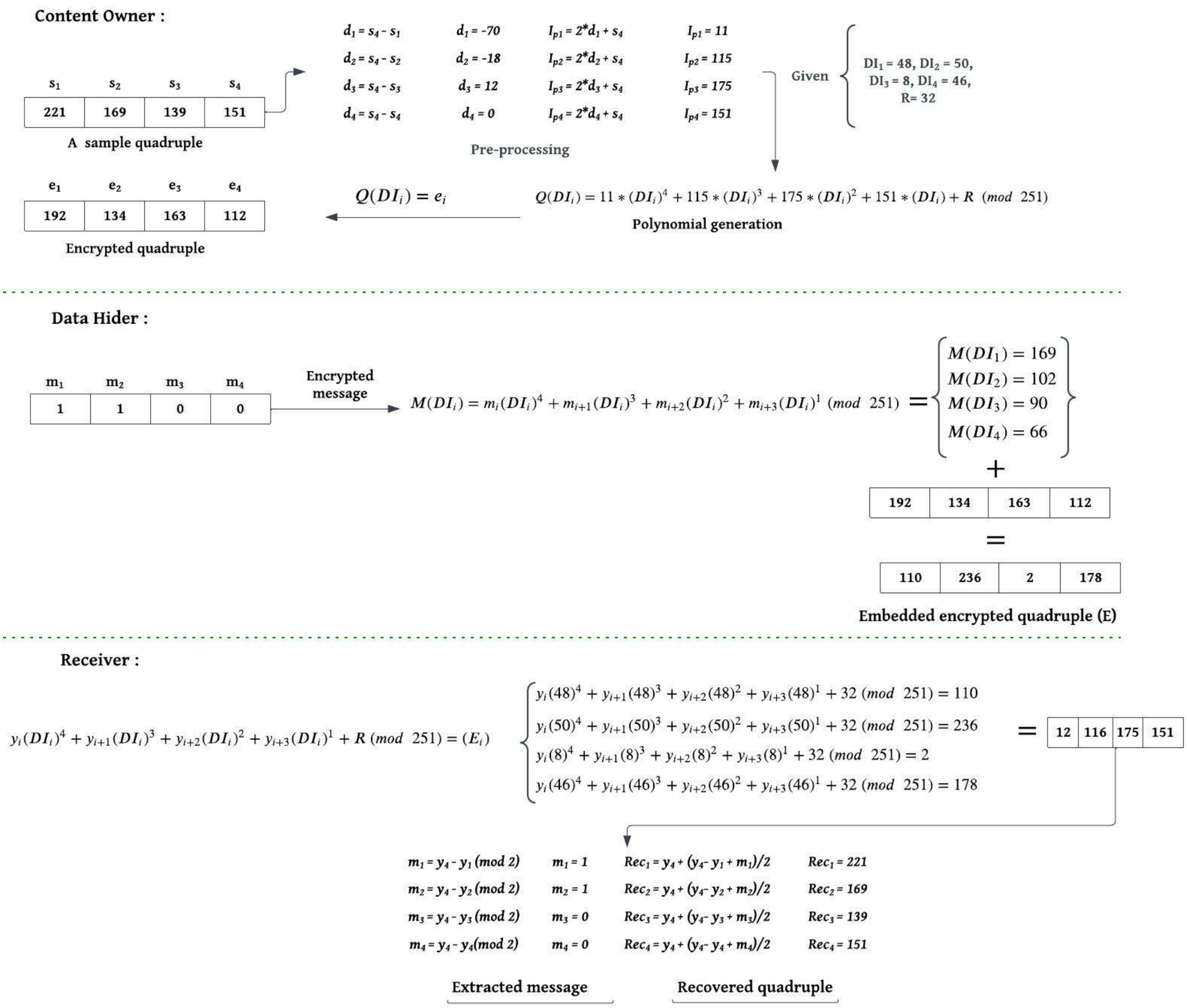


**Figure 3** Demonstration of proposed scheme.

A working example of the proposed scheme is demonstrated in Figure 3. The example below employs a random quadruple with values $s = [221, 169, 139, 151]$. The pre-processed values obtained after pre-processing are $I_p = [11, 115, 175, 151]$. To encrypt these values, a polynomial is generated using a random value ($\mathcal{R}$) and distinct identities ($DI$) (generated by a pseudo-random number generator in $\mathbb{F}$). Furthermore, as shown in Figure 3, the encrypted pixels ($e_i$) of each quadruple are calculated using $DI$ from the generated polynomial.

The data hider applies the same procedure for encrypting the secret message (without using a random value) and then performs the addition of these encrypted message bits and encrypted pixels to get embedded encrypted pixels ($E$). Finally, to obtain the decrypted values ($y$), the receiver generates a polynomial and solves this system of linear

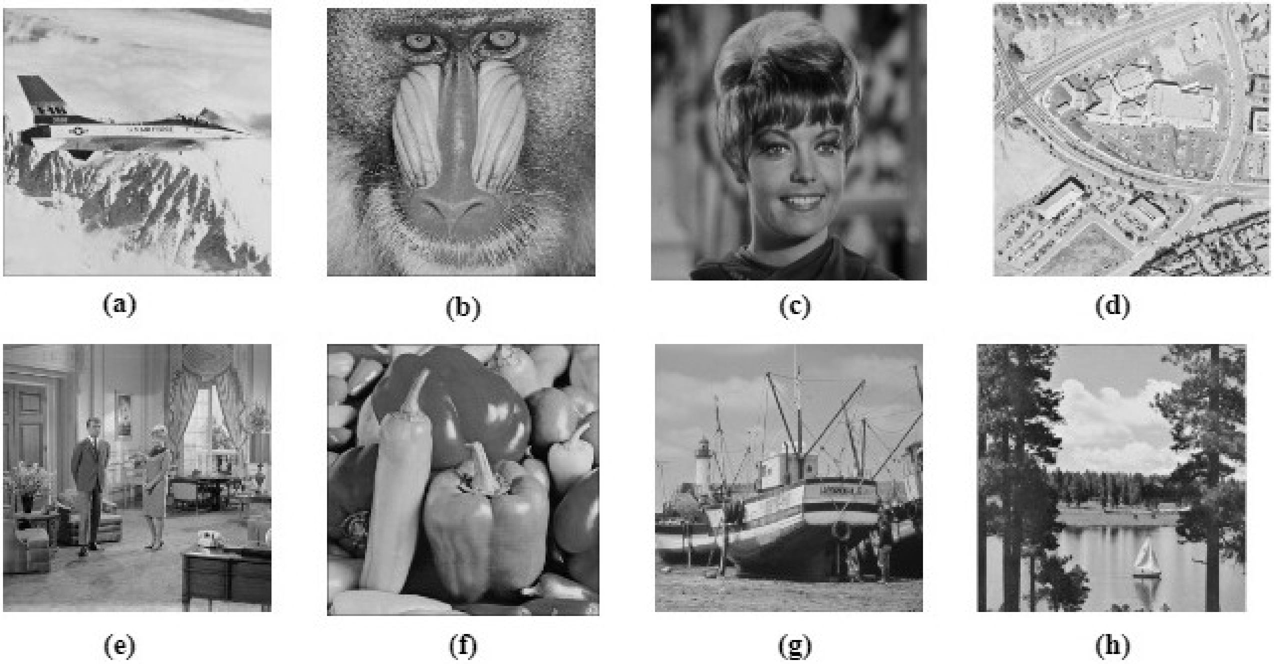


**Figure 4** Standard test images (a) Airplane, (b) Baboon, (c) Zelda, (d) Aerial, (e) Couple, (f) Peppers, (g) Boat and (h) Sail.

equations using the distinct identities and random values (shared via the Diffie-Hellman key exchange algorithm). It is important to note that the result of solving equations may be in the form of $\frac{a}{b}$. Here, $\frac{a}{b}$ is not a simple division because all operations are performed on the field (there is no division operation in the field). Instead, it is a multiplicative inverse calculated as $a.b^{-1}$ over the field $\mathbb{F}$. Now, to extract the message, simply subtract fourth pixel from all other pixels, followed by a modulo 2 operation. Finally, $Rec_i = ((y_4 - y_i + m_i)/2)$ is used to reconstruct the original pixels, where $i$ varies from 1 to 3 for a quadruple.

## 4 | RESULTS AND ANALYSIS

Testing of the proposed work is done using a variety of standard test images. The proposed work is performed on Windows 11 with 8GB of RAM and an Intel(R) Core(TM) i5-8265U CPU with MATLAB 18. The efficiency of any RDH-EI can be determined by its embedding rate, security parameters, and computational complexity. Some of these parameters are discussed below to evaluate the performance of the proposed work.

### 4.1 | Embedding capacity Analysis

Embedding capacity ($EC$) is defined as the number of bits embedded per pixel. The calculation of the embedding capacity is performed using the following formula:

$$EC = \frac{M}{N} \tag{23}$$

where, $M$ is the total number of embedded bits and $N$ is the total number of pixels in an image.
In the proposed work, the maximum possible embedding rate is up to 0.75 bits per pixel as embedding is done in three out of four pixels using the proposed modified difference expansion method. The pixels are not embedded in

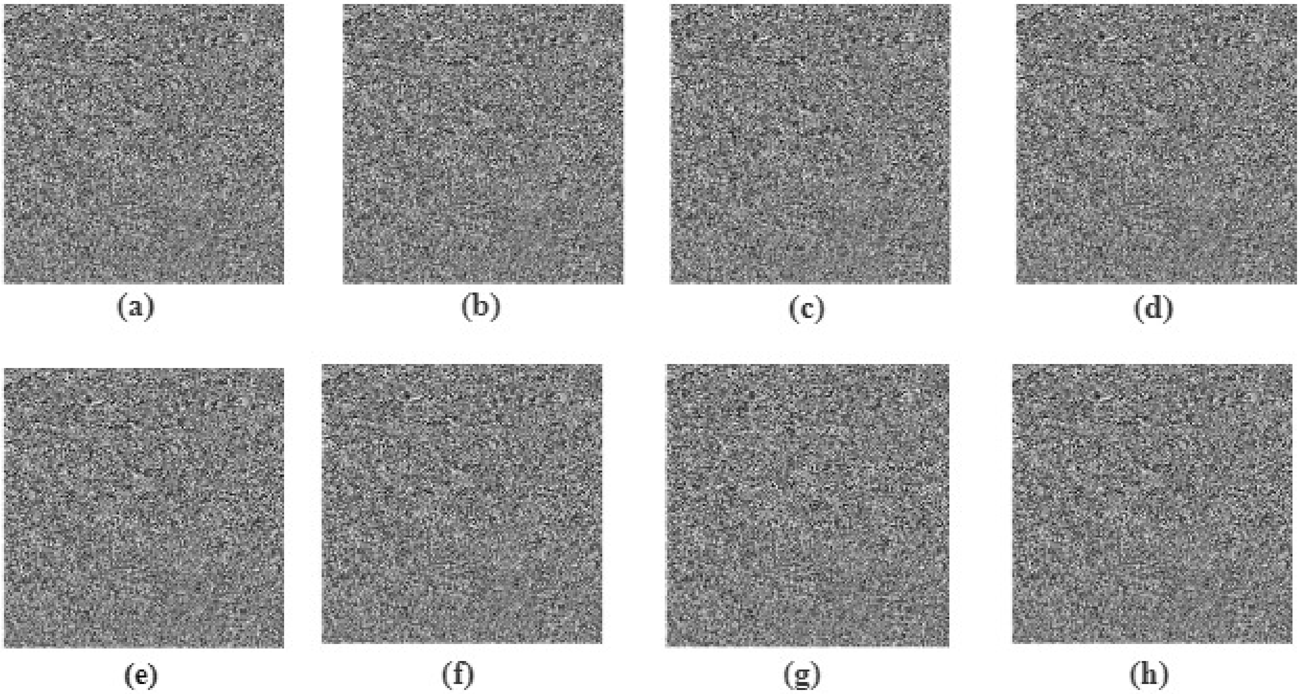


**Figure 5** Corresponding encrypted images of Figure 4(a-h).

cases of overflow or underflow, and they are also not embedded because they were chosen to be 251 pixels above or below this range. Let $q$ represent the pixels that are not embedded in all of these cases, then, the total embedding rate ($TER$) would be:

$$TER = \frac{(N - q) * 0.75}{N} \tag{24}$$

The embedding capacity of the proposed scheme is shown in Table 1, along with the embedding capacity of some other recent works. It can be clearly analysed from Table 1 that the proposed work achieves a higher embedding rate when compared to recent works. For instance, the embedding rate for the standard test image *Airplane* in[35] was around 0.125 bpp, in[36] was about 0.250 bpp, and in[31] almost equal to 0.500 bpp, whereas in the proposed work it is approximately 0.700 bpp, which is significantly better among all the other recent works, reflecting the potency of the proposed work.

## 4.2 | Security Analysis

In the proposed work, encryption is performed at the data owner stage. Also, during data hiding, in quadruple of the image, three out of four pixels are embedded. It should be noted that during the process of data hiding, a bit value of 0 or 1 from the secret message can be embedded in the first three pixels of each quadruple. The message is embedded by using a polynomial, and the values obtained after solving the polynomial are added to encrypted values generated at the data owner stage. All pixel values in an embedded encrypted image, including those embedded with 0 get changed during data hiding. Thus, data hiding adds more security to the final embedded encrypted image.

**Table 1** Comparison of embedding capacity.

| Methods | Airplane | Baboon | Zelda | Aerial | Couple | Peppers | Boat | Sail |
|---|---|---|---|---|---|---|---|---|
| Tsai et al.[37] | 0.2612 | - | 0.1924 | 0.3127 | 0.2636 | - | - | 0.1227 |
| Yu et al.[36] | 0.2477 | 0.2426 | 0.2679 | 0.2479 | 0.2479 | 0.2477 | 0.2477 | 0.2469 |
| Manikandan et al.[35] | 0.1250 | 0.0625 | 0.1250 | 0.1250 | 0.1250 | 0.1250 | 0.1250 | 0.1250 |
| Ke et al.[32] (HDE-ED) | 0.4381 | 0.2642 | 0.4402 | - | - | 0.4217 | - | - |
| Ke et al.[32] (DE-IS) | 0.4044 | 0.4956 | 0.5562 | - | - | 0.5090 | - | - |
| Chen et al.[31] | 0.4948 | 0.4756 | 0.4948 | 0.4948 | 0.4948 | 0.4948 | 0.4948 | 0.4948 |
| Proposed | 0.6861 | 0.5029 | 0.7278 | 0.5734 | 0.6821 | 0.6960 | 0.6910 | 0.6506 |

### 4.2.1 | Statistical Analysis

Several measures can be used to evaluate the encryption efficiency of images, including the peak signal-to-noise ratio (PSNR)[38], the structural similarity index metric (SSIM)[39], entropy[40], and histograms. Analyzing the image encryption efficiency measures between the original and encrypted image enables us to ascertain the security of the scheme. Table 2 shows the experimental values of PSNR and SSIM between the original and the encrypted image. The third parameter, entropy, of the encrypted image is also shown in Table 2. Figures 6 and 7 show the histograms of the original and encrypted test images.

As described by Thu et al.[38], PSNR is used to estimate the distortion and visual quality between the original and the encrypted image. Better visual quality is indicated by high PSNR value. Lower PSNR value reflects poor visual quality. It can be seen from Table 2, that the PSNRs of all encrypted images do not exceed 10 dB, thus, no visual information about the original image can be revealed from the encrypted images. This demonstrates the perceptual security of the encryption scheme.

Similarly, in order to analyze the degradation in the reconstructed image, SSIM[39] is calculated whose values ranges between -1 to 1. Here, value 1 indicates the identical reconstruction to the original image. The SSIM values in Table 2 clearly shows all encrypted images have values near to 0 which reflect almost no similarity between original and encrypted images.

A measure of randomness or average uncertainty in an encrypted image is interpreted as entropy[40]. A larger entropy implies a better statistical security. Table 2 shows that the results of the entropy values for all the encrypted images are nearly equal to 8, which can be the maximum for 8 bit representation for gray scale images. The entropy values in Table 2 illustrates that the encryption scheme is also secure from the viewpoint of statistical results.

For instance, Table 2 shows that the PSNR, SSIM, and entropy values of the encrypted image *Airplane* are 8.2036, 0.0113, and 7.9675, respectively. These experimental values clearly specify that from the encrypted image, nothing can be guessed about the original content of the image. Furthermore, Figures 6 and 7 clearly show that in the encrypted version, no information is disclosed about the original image.

**Table 2** Security metrics for the encrypted image.

| Security Measures | Airplane | Baboon | Zelda | Aerial | Couple | Peppers | Boat | Sail |
|---|---|---|---|---|---|---|---|---|
| PSNR (db) | 8.3500 | 10.9986 | 9.2790 | 9.2285 | 10.0951 | 9.4394 | 9.8263 | 8.8593 |
| SSIM | 0.0493 | 0.2203 | 0.0135 | 0.1117 | 0.0529 | 0.0480 | 0.0445 | 0.0642 |
| Entropy | 7.9675 | 7.9318 | 7.9698 | 7.9541 | 7.9674 | 7.9673 | 7.9699 | 7.9674 |

### 4.2.2 | Edge Distortion

Edge distortion refers to the distortion or blurring of edges in an image. This can occur due to a variety of factors such as camera lens imperfections, poor lighting conditions, or image compression. Edge distortion can negatively impact the performance of image processing applications, as it makes it more difficult to accurately detect and locate edges in the image.

To attain perceptual security in image encryption, it is important to evaluate the edge ratio and edge distortion ratio[42,44]. The Edge Ratio (ER) is used to evaluate the preservation of edges in the encrypted image. It compares the number of edges present in the original image to the number of edges present in the encrypted image. A lower value of ER indicates that the edges in the original image are not well preserved in the encrypted image, which improves the perceptual security. From table 3, it can be deduced that the average edge ratio value of the images is approximately 0.3221 that is sufficiently low. Thus, the proposed scheme attains a high perceptual security.

The Edge Distortion Ratio (EDR) is used to evaluate the distortion of edges in the encrypted image. It compares the difference in the edge intensity along the edges between the original and encrypted images. A high EDR indicates that the edges in the original image are highly distorted in the encrypted image, which improves the perceptual security. From Table 3, it can be seen that all the values for edge distortion ratio are near to 1, that indicates the efficacy of proposed scheme.

Evaluating these parameters can provide a measure of how well the encryption algorithm distorts the edges in the image, which is important for maintaining the perceptual security of the encrypted image. ER and EDR are defined as:

$$ER = \frac{\sum_{i,j=0}^{N-1} \hat{A}_{i,j}}{\sum_{i,j=0}^{N-1} A_{i,j}} \tag{25}$$

$$EDR = \frac{\sum_{i,j=0}^{N-1} |A_{i,j} - \hat{A}_{i,j}|}{\sum_{i,j=0}^{N-1} |A_{i,j} + \hat{A}_{i,j}|} \tag{26}$$

where, $A_{i,j}$ and $\hat{A}_{i,j}$ denote the pixel values in an edge detected binary matrix for the original and encrypted image respectively.

**Table 3** Edge ratio (ER), edge distortion ratio (EDR), NPCR ratio and UACI ratio for the encrypted images.

| Different Ratios | Airplane | Baboon | Zelda | Aerial | Couple | Peppers | Boat | Sail |
|---|---|---|---|---|---|---|---|---|
| ER | 0.3067 | 0.2953 | 0.3775 | 0.2643 | 0.2573 | 0.5076 | 0.2895 | 0.2790 |
| EDR | 0.9923 | 0.9578 | 0.9843 | 0.9792 | 0.9814 | 0.9019 | 0.9816 | 0.9920 |
| NPCR | 0.9112 | 0.6675 | 0.9666 | 0.7613 | 0.9057 | 0.9242 | 0.9176 | 0.8641 |
| UACI | 0.2986 | 0.2191 | 0.3183 | 0.2501 | 0.2979 | 0.3033 | 0.3020 | 0.2846 |

### 4.2.3 | Differential Attack Analysis

A differential attack is a chosen-plaintext attack in which a cryptanalyst has ciphertext of slightly different plaintext and tries to deduce a statistical relationship between them by computing the difference between the chosen plaintexts and their corresponding ciphertexts[43].

Number of changing pixel rate (NPCR) and unified average changed intensity (UACI) are two statistical evaluation parameters that can be used to measure the robustness of a multimedia encryption technique against a differential attack. NPCR is the ratio of the number of pixels that are different between the original and encrypted images to the total number of pixels in the image. UACI is the average intensity difference between the original and encrypted images.

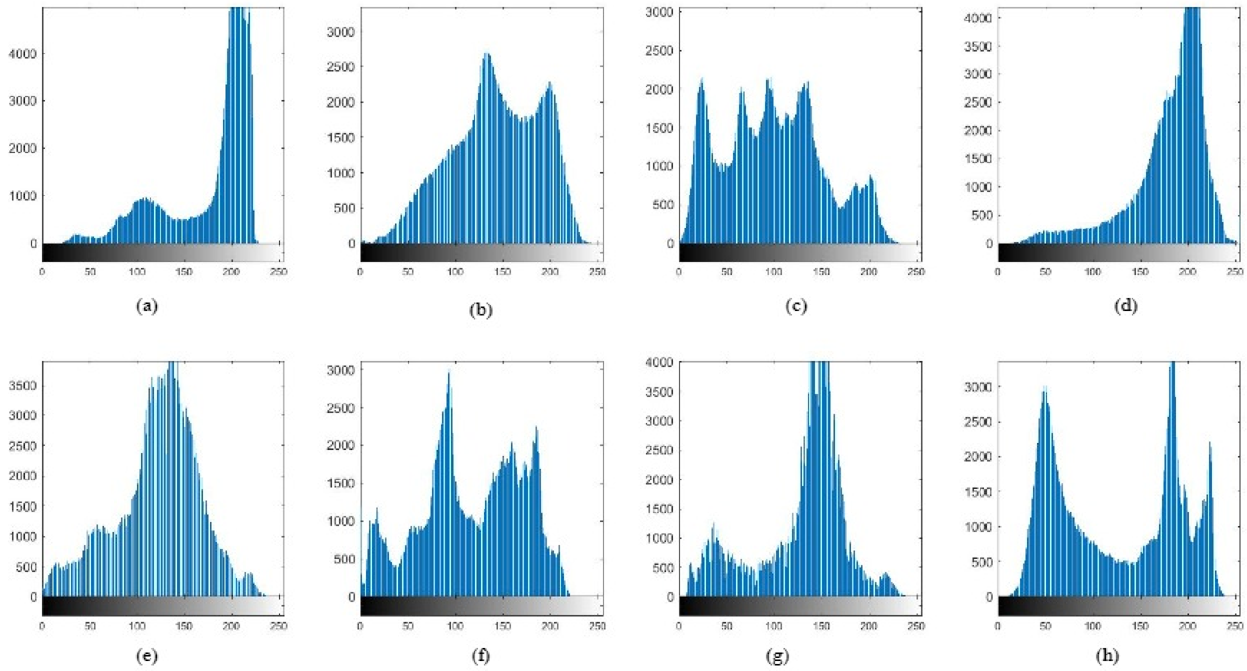


**Figure 6** Histogram of original images of Figure 4(a-h).

Both NPCR and UACI are used to evaluate the level of similarity between the original and the encrypted images. These metrics also test the encryption strength[41]. The values of NPCR and UACI of a secure encrypted image are expected near to be the maximum limit, i.e., 1 and 0.33, respectively. Table 3 shows that the NPCR and UACI values are near to benchmark values, thus, the proposed work could resist differential attacks.

$$NPCR\,(E_1, E_2) = \sum_{i,j} \frac{B(i,j)}{P} \tag{27}$$

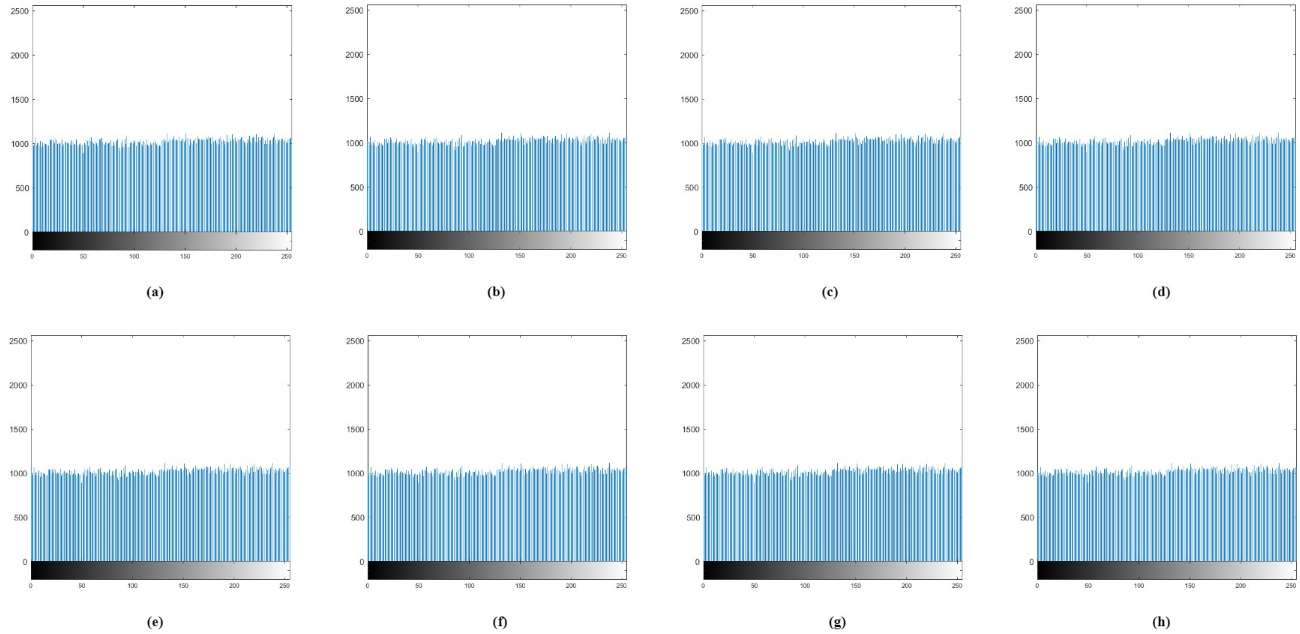
$$UACI\,(E_1, E_2) = \sum_{i,j} \frac{|E_1\,(i,j) - E_2(i,j)|}{S.P} \tag{28}$$

where, $E_1$ is encrypted image of original image and $E_2$ is the encrypted image corresponding to the original image differing with the value at one pixel only. $P$ is the number of pixels in the encrypted image, $|.|$ denotes the absolute value function, $B$ is bipolar array as in[41] and $S$ is the largest pixel value; $i, j$ are the index variables.

## 4.3 | Theoretical Analysis

A time complexity analysis is presented here. The multi-secret sharing operation takes a consistent amount of time to access all $N$ pixels in the original image. In the same way, data hiding is performed once over all the pixels, and for every pixel in an encrypted image, there will be a pixel in an embedded encrypted image. This procedure will also be completed in constant time. As a result, the proposed data hiding scheme has linear time complexity. Thus, when the encrypted image size is $N$, the proposed data hiding scheme has a time complexity of $O(N)$. Each embedded encrypted image must be accessed exactly twice in order to extract the message and restore the image. Consequently, message extraction and image restoration have an overall time complexity of $O(N)$.

RDH schemes that process all pixels in an image have a theoretical time complexity of $O(N)$, allowing the proposed scheme to be compared to or even better than existing schemes. Also, the proposed scheme reduces the order of the polynomial, as the number of random values is one-fourth when compared to the random values taken in reference[31].



(a) (b) (c) (d)

(e) (f) (g) (h)

**Figure 7** Histogram of encrypted images of Figure 5(a-h).

In the proposed work, the polynomial generated has an order of $(N + N/4 - 1)$ when there were a total of $N$ pixels in the image, which is significantly less than the order of the polynomial $(2N - 1)$ reported in[31]. Thus, the proposed scheme takes less computational time than that discussed in[31].

## 5 | CONCLUSION

The present work introduces an enhanced method for reversible data hiding in encrypted images with a high embedding rate. According to the proposed approach, no key is shared to encrypt or decrypt the original or encrypted image with the data hider. For enhanced security, only the Diffie-Hellman key exchange algorithm is used to transfer the key between the owner and the sender. The original image is pre-processed first and then encrypted using a multi-secret sharing scheme by the data owner. The encrypted image is sent to the data hider to hide the secret message. The secret message is also encrypted using a multi-secret property. During data extraction and image recovery, with the help of the same key, secret data is extracted, followed by image reconstruction. The results of the experimental analysis utilizing various standard test images indicate that the embedding rate is satisfactory. Future research can focus on improving the embedding rate in images with numerous sharp edges or high amounts of texture and also, compressing the size of the location map using compression algorithms.

## Author contributions

Conception/ design of the work - Ruchi Agarwal and Manoj Kumar.
Data analysis and interpretation - Ruchi Agarwal and Manoj Kumar.
Drafting the article - Ruchi Agarwal.
Critical revision of the article - Manoj Kumar and Ruchi Agarwal.

## Financial disclosure

None reported.

## Conflict of interest

The authors declare no potential conflict of interests.

## References

1. Cox, I., Miller, M., Bloom, J., Fridrich, J. and Kalker, T., 2007. Digital watermarking and steganography. Morgan Kaufmann.

2. Barton, J.M., 1997. Method and apparatus for embedding authentication information within digital data. United States Patent, 5 646 997.

3. Tian, J., 2003. Reversible data embedding using a difference expansion. IEEE transactions on circuits and systems for video technology, 13(8), pp.890-896.

4. Ni, Z., Shi, Y.Q., Ansari, N. and Su, W., 2006. Reversible data hiding. IEEE Transactions on circuits and systems for video technology, 16(3), pp.354-362.

5. Celik, M.U., Sharma, G., Tekalp, A.M. and Saber, E., 2005. Lossless generalized-LSB data embedding. IEEE transactions on image processing, 14(2), pp.253-266.

6. Hong, W., 2012. Adaptive reversible data hiding method based on error energy control and histogram shifting. Optics Communications, 285(2), pp.101-108.

7. Hong, W. and Chen, T.S., 2010. A local variance-controlled reversible data hiding method using prediction and histogram-shifting. Journal of Systems and Software, 83(12), pp.2653-2663.

8. Jung, S.W. and Ko, S.J., 2010. A new histogram modification based reversible data hiding algorithm considering the human visual system. IEEE Signal Processing Letters, 18(2), pp.95-98.

9. Qin, C., Chang, C.C. and Chen, Y.C., 2013. Efficient reversible data hiding for VQ-compressed images based on index mapping mechanism. Signal Processing, 93(9), pp.2687-2695.

10. Qiu, Y., Qian, Z. and Yu, L., 2015. Adaptive reversible data hiding by extending the generalized integer transformation. IEEE Signal Processing Letters, 23(1), pp.130-134.

11. Wang, W., Ye, J., Wang, T. and Wang, W., 2017. Reversible data hiding scheme based on significant bit difference expansion. IET image processing, 11(11), pp.1002-1014.

12. Zhang, X., 2011. Reversible data hiding in encrypted image. IEEE signal processing letters, 18(4), pp.255-258.

13. Hong, W., Chen, T.S. and Wu, H.Y., 2012. An improved reversible data hiding in encrypted images using side match. IEEE signal processing letters, 19(4), pp.199-202.

14. Zhang, X., 2011. Separable reversible data hiding in encrypted image. IEEE transactions on information forensics and security, 7(2), pp.826-832.

15. Liao, X. and Shu, C., 2015. Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels. Journal of Visual Communication and Image Representation, 28, pp.21-27.

16. Qian, Z. and Zhang, X., 2015. Reversible data hiding in encrypted images with distributed source encoding. IEEE Transactions on Circuits and Systems for Video Technology, 26(4), pp.636-646.

17. Ma, K., Zhang, W., Zhao, X., Yu, N. and Li, F., 2013. Reversible data hiding in encrypted images by reserving room before encryption. IEEE Transactions on information forensics and security, 8(3), pp.553-562.

18. Zhang, W., Ma, K. and Yu, N., 2014. Reversibility improved data hiding in encrypted images. Signal Processing, 94, pp.118-127.

19. Cao, X., Du, L., Wei, X., Meng, D. and Guo, X., 2015. High capacity reversible data hiding in encrypted images by patch-level sparse representation. IEEE transactions on cybernetics, 46(5), pp.1132-1143.

20. Agrawal, S. and Kumar, M., 2017. Mean value based reversible data hiding in encrypted images. Optik, 130, pp.922-934.

21. Li, M. and Li, Y., 2017. Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding. Signal Processing, 130, pp.190-196.

22. Puteaux, P. and Puech, W., 2018. An efficient MSB prediction-based method for high-capacity reversible data hiding in encrypted images. IEEE transactions on information forensics and security, 13(7), pp.1670-1681.

23. Malik, A., Wang, H., Chen, T., Yang, T., Khan, A.N., Wu, H., Chen, Y. and Hu, Y., 2019. Reversible data hiding in homomorphically encrypted image using interpolation technique. Journal of Information Security and Applications, 48, p.102374.

24. Xu, D. and Wang, R., 2016. Separable and error-free reversible data hiding in encrypted images. Signal Processing, 123, pp.9-21.

25. Yi, S. and Zhou, Y., 2018. Parametric reversible data hiding in encrypted images using adaptive bit-level data embedding and checkerboard based prediction. Signal Processing, 150, pp.171-182.

26. Qin, C., Zhang, W., Cao, F., Zhang, X. and Chang, C.C., 2018. Separable reversible data hiding in encrypted images via adaptive embedding strategy with block selection. Signal Processing, 153, pp.109-122.

27. Yi, S., Zhou, Y. and Hua, Z., 2018. Reversible data hiding in encrypted images using adaptive block-level prediction-error expansion. Signal Processing: Image Communication, 64, pp.78-88.

28. Qin, C., He, Z., Luo, X. and Dong, J., 2018. Reversible data hiding in encrypted image with separable capability and high embedding capacity. Information Sciences, 465, pp.285-304.

29. Shiu, C.W., Chen, Y.C. and Hong, W., 2015. Encrypted image-based reversible data hiding with public key cryptography from difference expansion. Signal Processing: Image Communication, 39, pp.226-233.

30. Singh, P. and Raman, B., 2018. Reversible data hiding based on Shamir's secret sharing for color images over cloud. Information Sciences, 422, pp.77-97.

31. Chen, Y.C., Hung, T.H., Hsieh, S.H. and Shiu, C.W., 2019. A new reversible data hiding in encrypted image based on multi-secret sharing and lightweight cryptographic algorithms. IEEE Transactions on Information Forensics and Security, 14(12), pp.3332-3343.

32. Ke, Y., Zhang, M., Zhang, X., Liu, J., Su, T. and Yang, X., 2021. A reversible data hiding scheme in encrypted domain for secret image sharing based on Chinese remainder theorem. IEEE Transactions on Circuits and Systems for Video Technology, 32(4), pp.2469-2481.

33. Shi, Y.Q., Li, X., Zhang, X., Wu, H.T. and Ma, B., 2016. Reversible data hiding: advances in the past two decades. IEEE access, 4, pp.3210-3237.

34. Puteaux, P., Ong, S., Wong, K. and Puech, W., 2021. A survey of reversible data hiding in encrypted images - The first 12 years. Journal of Visual Communication and Image Representation, 77, p.103085.

35. Manikandan, V.M. and Zhang, Y.D., 2022. An adaptive pixel mapping based approach for reversible data hiding in encrypted images. Signal Processing: Image Communication, 105, p.116690.

36. Yu, M., Yao, H. and Qin, C., 2022. Reversible data hiding in encrypted images without additional information transmission. Signal Processing: Image Communication, 105, p.116696.

37. Tsai, C.S., Zhang, Y.S. and Weng, C.Y., 2022. Separable reversible data hiding in encrypted images based on Paillier cryptosystem. Multimedia Tools and Applications, 81(13), pp.18807-18827.

38. Huynh-Thu, Q. and Ghanbari, M., 2008. Scope of validity of PSNR in image/video quality assessment. Electronics letters, 44(13), pp.800-801.

39. Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4), pp.600-612.

40. Thum, C., 1984. Measurement of the entropy of an image with application to image focusing. Optica Acta: International Journal of Optics, 31(2), pp.203-211.

41. Wu, Y., Noonan, J.P. and Agaian, S., 2011. NPCR and UACI randomness tests for image encryption. Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT), 1(2), pp.31-38.

42. Taneja, N., Raman, B. and Gupta, I., 2011. Chaos based partial encryption of spiht compressed images. International Journal of Wavelets, Multiresolution and Information Processing, 9(02), pp.317-331.

43. Taneja, N., Raman, B. and Gupta, I., 2012. Combinational domain encryption for still visual data. Multimedia Tools and Applications, 59(3), pp.775-793.

44. Taneja, N., Raman, B. and Gupta, I., 2011. Selective image encryption in fractional wavelet domain. AEU-International Journal of Electronics and Communications, 65(4), pp.338-344.

45. Makoto Matsumoto and Takuji Nishimura. 1998. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo random number generator. ACM Trans. Model. Comput. Simul. 8(1), pp.3-30.

□