Multimedia Tools and Applications (2020) 79:10825–10849
https://doi.org/10.1007/s11042-019-08476-2

# A secure multimedia steganography scheme using hybrid transform and support vector machine for cloud-based storage

Arunkumar Sukumar[1] · V. Subramaniyaswamy[1] · V. Vijayakumar[2] · Logesh Ravi[3]

Received: 22 February 2019 / Revised: 8 October 2019 Accepted: 12 November 2019 /
Published online: 4 January 2020


**Abstract**
Cloud computing is widely accepted by both individuals and enterprises alike for the storage of multimedia contents. It is due to the introduction of a new architecture where the cost of computation, storage, and services needed for maintenance for storage of multimedia are less. Cloud computing addresses the scarcity of resources for clients by offering options to pay for services only as they are used. But once the organization's multimedia contents are uploaded into cloud space, the user loses control over their contents which may no longer be safe. The cloud user has to take some measure to avoid privacy issues. Steganography is preferred over encryption for providing multimedia security as content concealed in a cover image is not revealed. The multimedia content is transformed using Discrete Rajan Transform (DRT) and embedded into a chosen cover image which is created by Integer Wavelet using Diamond Encoding Scheme. Generated stego images are stored in the cloud. When the multimedia content is required, stego images are downloaded from the cloud and are subjected to inverse transform of IWT. SVM provides Good learning ability to our extraction process which makes our algorithm more robust to various attacks, viz., salt and pepper noise, Gaussian noise, cropping, compression, etc. Experimental values for Peak Signal to Noise Ratio (PSNR) for two secret images are 53 and 50 respectively which is better over the available schemes in the literature. Similarly for robustness and security evaluation, our scheme provides a better result.

**Keywords** Multimedia security · Image steganography · Discrete Rajan transform · Support vector machine · Diamond embedding

## 1 Introduction

When an enterprise is started, more investments are made to safely maintain its secret data, which range from buying and maintaining a server to paying salary to network administrator

✉ V. Subramaniyaswamy
vsubramaniyaswamy@gmail.com

Extended author information available on the last page of the article

[1]. Nowadays, instead of maintaining an independent server to store an organization's confidential data, server space in the public cloud can be hired and used [2]. Cloud computing, an emerging technology in recent times, represents a host of advanced technologies, services, marketing techniques and exciting business opportunities. Cloud computing services can be broadly viewed as three different categories: Infrastructure as a Service(IaaS), Software as a Service(SaaS) and Platform as a Service(PaaS) [3]. Incoherence with another emerging technology called the Internet of Things (IoT), Cloud Computing offers seamless and consistent connection, intelligent algorithms, applications and social networks to a wide range of devices and people [4]. The marriage of the Internet of Things (IoT) with multiple technologies creates a wide range of exciting applications. One such core application, Software Defined Networking(SDN), is a solution for offering centralized control architecture in a manner that is both flexible and powerful [5]. In the midst of this dynamic progress, security and privacy concerns pose integral challenges. As the world expands to consume multimedia through IoT services, providing a basic form of enterprise security and authentication is mandatory. Though a considerable amount of flaws exist with present-day web services, simple and straightforward measures go a long way in safeguarding the service [6]. To improve multimedia security issues with the transmission of multimedia, countermeasures are taken to respond to attacks and specify the ways to recover from attacks [7].

Information security is a broad branch of study, where methods to keep user multimedia contents safe in personal computers, in transit across a network, and in storage in cloud space are discussed. It is broadly classified into two types: encryption and information hiding. Encryption satisfies the confidentiality property of cryptography. Through encryption, the produced cipher is of meaningless or unintelligible form, which may invite suspicion [8]. But if multimedia content is subjected to information hiding using some cover media, mere knowledge of the existence of multimedia itself is suppressed which may evade suspicion. Information hiding provides confidentiality and at the same time suppress suspicion. So, information hiding is preferred over encryption. Information hiding is of two types: steganography and watermarking. Steganography is the art of sending a secret to a destination without other knowledge. Image, audio or video can be used as a cover for transmitting secret data in steganography. Based on the quantum of secret data to hide in the cover medium and based on the operational simplicity, the cover medium is chosen. If an image is used as a cover for transmitting multimedia contents, then the process is called image steganography [9].

Image steganography is done in both spatial and frequency domains. The Least Significant Bit (LSB) based method in spatial domain steganography embeds the payload of any size into a cover medium [10]. LSB embedding may introduce some distortion in the resultant image. To avoid introducing errors, embedding is done in the frequency domain which provides better imperceptibility and robustness [11]. Once the image is transformed, the image is divided into four sub-bands: LL, HL, LH and HH. LL is low-frequency sub-bands and HH, LH, and HL are high-frequency sub-bands. Embedding can be done in any of the sub-bands, but if embedding is done in HH sub-band, the output Stego image is not robust enough against attacks, so embedding in HH sub-band is avoided [12].

### 1.1 SVM based steganography

Recently efforts are made to exploit advantages of machine learning techniques like SVM for embedding and extraction of secret images. Support vector machine is utilized for the extraction of secret images from the tampered stego images. Probability of extraction of the

original secret image from attacked stego image is more while SVM is utilized in the extraction process [13]. SVM can also be used in the wavelet domain. A relation is established between the statistical characteristics of the wavelet coefficient for successive data using SVM and further used in the extraction of the original secret from the tampered stego image [14].

### 1.2 Salient features of our approach

The proposed steganographic scheme is an enhanced variant of the existing approach by following aspects:

1. The secret image is transformed using a Discrete Rajan Transform (DRT) before embedding. DRT is used to increase the security of the secret image. DRT performs many rounds of shifting and does some mathematical operations on the secret image. So steganalysis cannot be done based on pattern matching or using any other image processing techniques.
2. DRT transformed images are subjected to Base b ($b = 2k^2 + 2k + 1$, where $k \geq 1$ is embedding parameter) conversion. It increases the embedding rate to some extent.
3. In DWT, integer pixel values of the secret image are converted into double values while transforming. While the inverse of DWT is applied, this double value may not be mapped into an exact integer value. Because of wrong mapping, some amount of round of error is introduced in the stego image. To overcome the roundoff error, Integer Wavelet Transform is used for transformation.
4. SVM classifier is introduced in the extraction phase which increases the probability of recovering the original secret image even after some image processing attacks are done on the stego images.

As all the above four techniques are combined, the resultant approach provides better security, enhanced embedding rate, less roundoff error and robust to any kind of attacks.

We propose an adaptive steganographic scheme for providing multimedia security in the cloud. Multimedia refers to audio, image, video or text. In our proposed work, image is considered as a multimedia content throughout the article, but the proposed method can easily be used with other types of multimedia. Prior to sending a secret image into cloud space, it is first subjected to DRT transformation and embedded using Diamond Encoding method into an IWT transformed cover image. The resulting stego image is stored in the cloud and whenever the multimedia content is required, the stego image is downloaded from the cloud and the multimedia content is extracted using a similar process. Usage of a Support Vector Machine increases the robustness of extraction of stego images which makes the extraction of the original secret image from tampered stego image possible.

Remainder of the article is organized as follows: Section 2 summaries similar contributions, Section 3 describes the basic concepts used in our proposed work, Section 4 explains proposed methods, Section 5 gives the detailed experimental results and analysis. Section 6 provides a conclusion and future works.

## 2 Literature survey

In earlier stages of spatial steganography, LSB of cover image pixels is replaced in such a way that multimedia content can be embedded into it to produce stego images. The stego images

thus produced are of low distortion and high quality [15]. All image pixels may not be used for embedding a multimedia content; the cover image may be divided into blocks and LSB embedding can be done in some of the pixels within a block [16]. Even if a multimedia content is of very large size which can not be embedded into a single cover image, multimedia content can be divided into many segments and are LSB embedded separately into a many cover images [17]. LSB replacements are not changing the odd values of intensities of the image, but change only even values of an image. In LSB replacement, statistical changes happen which may pave the way for steganalysis [18]. To overcome this problem, LSB matching is used. In this approach, before embedding a secret bit in a pixel, matching is done and accordingly either one is added or subtracted from the pixel value. Because of this modification, statistical changes are minimized [19]. But, both LSB replacement and LSB matching methods may modify 50 % of pixel intensity values. LSB matching the revised approach, which is based on a block of pixels, is proposed to mitigate this problem. Using some specific functions and tables, embedding is done so that it utilizes almost all possible modifications of pixels optimally to preserve the quality of stego images [20].

Steganography is done in the transform domain as well. A given cover image is transformed from its time-domain into its equivalent frequency domain using a mathematical function. At the earlier stage of usage of transform domain, Fourier Transform (FT) and Discrete Cosine Transform (DCT) are used to transform a multimedia content [21]. But nowadays, Discrete Wavelet Transform (DWT) is used in transformation because of its high quality and flexibility. But wavelet transform converts the pixel intensity values of integer values of time domain into floating-point value coefficients of the frequency domain. In the extraction stage, the reverse of conversion is done from floating values of coefficients into integer values of pixels. Some amount of error is introduced because of this conversion [22]. To overcome this limitation, Integer Discrete Wavelet Transform (IDWT) is proposed [23]. Usually, the cover image alone is transformed and multimedia content is embedded into cover image without doing any transform. To increase the security of multimedia in the embedding algorithm, the multimedia content is also transformed and then embedded into the transformed coefficients of the cover image [24].

Enterprise Security in hardware remains a major challenge for the successful implementation of IoT in the industry. Solutions have been explored by researchers and organizations to prevent unauthenticated attacks on a device commonly termed as outside device attack [25]. Successful solutions are a critical requirement for the deployment of the fog computing environment and the Internet of Things (IoT) in the industry of Information Technology. Data science, an interdisciplinary track, allows various experts to work in cohesion and obtain meaningful information through investigation of data [26].

The multimedia content is compressed using Huffman encoding. Generated bitstreams are divided into three bits each using 3-bit block preparation and are converted into decimal notation. A cover image is chosen, which is transformed using Haar-Wavelet. Embedding of the multimedia content is done in the high-frequency components of the transformed image as it may not reveal the existence of the secret image within it [27]. Cover images can be represented in YCbCr color space. Both the Cr component and the multimedia content are transformed using DWT. Coefficients of the multimedia content are embedded into coefficients of the Cr component of the cover image by dividing both multimedia content and cover image into pixel pairs of equal size and comparing both before embedding a bit in it [28]. The cover image can be transformed using DWT twice and embedding is done in subband LL2. 5 MSB bits of each pixel in the multimedia content is embedded into 5 LSB of each coefficient

of LL2 [29]. The cover image can be transformed using DWT. The multimedia content is subjected to Base b ($b = 2k^2 + 2k + 1$, where k is embedding parameter) conversion. Then each Base b digits are embedded into coefficients of subbands using Diamond Encoding Scheme which treats a pair of pixels as embedding units [30].

Discrete Rajan Transform does coding morphism. In DRT, the number sequence of the length of the power of two is mapped into a highly related same length sequence. It is most useful in data compression and pattern recognition [31]. If a stego image is attacked by image processing attack, such as the addition of noise, clipping or blurring of a portion of an image, extraction may not be accurate. But due to the usage of support vector machine, even if a stego image is subjected to some image processing attacks, extraction is done accurately without any trace of attacks [32].

## 3 Technical background

### 3.1 Integer wavelet transform(IWT)

In Integer Wavelet Transform (IWT), a cover image undergoes frequency domain transform to obtain detailed coefficients and that coefficients are used for classification based on most significant bits (MSBs). As IWT prevents modifications in MSBs, it helps to extract the secret image precisely without any errors from the stego image. IWT maps integer value of each pixel into integer coefficients in the forward transformations. It reduces the error occurs in mapping integer coefficients into its corresponding integer pixel values during inverse transformations. The forward and inverse transformations of IWT is given in the following equations

$$\begin{aligned} d_{1,n} &= s_{0,2n+1} - s_{0,2n} \\ s_{1,n} &= s_{0,2n} + \left\lfloor {}^{d_{1,n}}/_{2} \right\rfloor \end{aligned} \tag{1}$$

$$\begin{aligned} s_{0,2n+1} &= d_{1,n} + s_{0,2n} \\ s_{0,2n} &= s_{1,n} - \left\lfloor {}^{d_{1,n}}/_{2} \right\rfloor \end{aligned} \tag{2}$$

where $s_{i,n}$ and $d_{i,n}$ are the nth low and high-frequency coefficients of the ith level wavelet, respectively. After a successful transformation, one approximation coefficient matrix and three detailed coefficient matrixes were obtained. Finally, the stego image is generated by concealing bits of the confidential image within the frequency coefficients of cover media. By reversing the embedding algorithm, original confidential image bits are recovered from the stego image by employing inverse IWT in the extraction phase. HH matrix is first selected for embedding to provide reduced distortion, and then other detailed matrices are used for embedding subsequently if required.

### 3.2 Discrete Rajan transform (DRT)

Discrete Rajan Transform is an efficient and fast algorithm derived from Decimation-In-Frequency (DIF) algorithm and Fast Fourier Transform (FFT). Let us consider that x(n) be a

number with length N and also a power of 2, then x(n) is decomposed into two equal parts as follows

$$g(j) = x(i) + x\left(i + \left(\frac{N}{2}\right)\right), 0 \le j \le \frac{N}{2}, 0 \le i \le \frac{N}{2} \tag{3}$$

$$h(j) = x(i) - x\left(i - \left(\frac{N}{2}\right)\right), 0 \le j \le \frac{N}{2}, \frac{N}{2} \le i \le N \tag{4}$$

The division process continues and terminates when there is no option for further division. Thus the level of division is log2N. The operators + and –denotes the function addition and subtraction respectively. For a sequence of number with length 8, the one-dimensional signal flow of DRT is illustrated in Fig. 1.

Suppose that the length of the sequence of a number is 2 k, where k is greater than 0, then DRT is denoted by X(k). DRT supports isomorphism which means it maps a domain set containing dyadic and cyclic permutations onto the range set which is in the form of X(k)E(r), where E(r) indicates encryption code and x(k) indicates permutation invariant. DRT is treated as transform as it is one to one and onto correspondence and also inverse facilitates inverse DRT. The forward transform is performed by recursively dividing the input sequence and partitioning the matrix operator R matrix as follows

$$A_{N\times 1} = R_N \times X_{N\times 1} \tag{5}$$

$$R_N = \begin{bmatrix} I_{N/2} & I_{N/2} \\ -e_k I_{N/2} & e_k I_{N/2} \end{bmatrix} \tag{6}$$

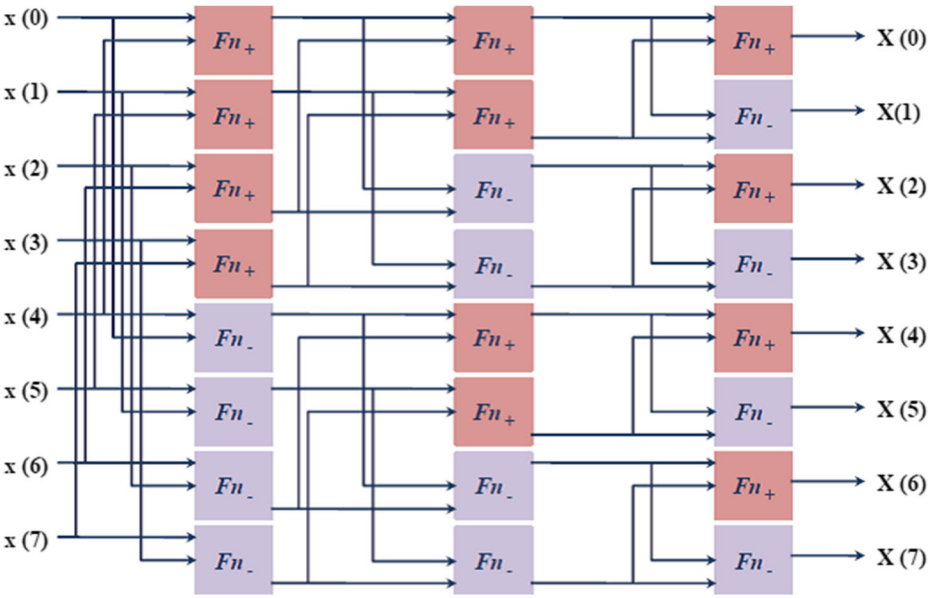


**Fig. 1** One dimensional signal flow of DRT of length 8

where $X_{N\times 1}$ denotes column matrix of an input whose sequence length is N, $R_N$ is the N × N sized R matrix, $I_{N/2}$ is the identity matrix and $e_k$ represents encryption function. The value of k is determined as follows

$$e_k = (-1)^k \text{ such that k} = \begin{cases} 1 & \text{for } x(i+N/2) < x(i); 0 \le i \le N/2 \\ 0 & \text{Otherwise} \end{cases} \tag{7}$$

The signal x(n) can be recovered by the inverse DRT with the help of DRT coefficients and its corresponding encryption values used in the forward DRT [31].

### 3.3 Support vector machine (SVM)

A binary classification algorithm, SVM is utilized for classification and error detection. Based on the statistical learning hypothesis, SVM is trained to detect the errors by applying a hypothetical upper limit for error generalization. Theoretically, error predictions are determined for unknown or new instances by the binary classification. Figure 2 depicts the classes and hyperplane of the SVM classifier. Let us assume that there are m data points represented by Xi, where i = 1, 2, 3 … m and Yi = ±1 be the two labelled class and decision function f(x) is given as follows

$$f(s) = sign(w \times x_i + b) \tag{8}$$

Various classification algorithms are available for datasets with linear classes. But for non-linear class datasets, SVM is best employed with kernel substitutions. Image processing techniques provide better results when combined with SVM classifiers [32].

### 3.4 Diamond encoding

The Diamond Encoding (DE) is a spatial domain-based transform used in the proposed hybrid transform-based image steganography to conceal secret bits in the cover media. The embedding units of DE are neighbourhood pairs (p, q) of cover pixels. An image which is to be protected is first transformed into digits of base b, where b = 2 k2 + 2 k + 1 and k is embedding parameter(or embedding rate), which is always greater than or equal to 1. The lower limit of k is estimated as follows

$$\left\lfloor \frac{M \times N}{2} \log_2\left(2k^2 + 2k + 1\right) \right\rfloor \ge |S| \tag{9}$$

where M × N indicates cover media size and |S| denotes the size of the medical image. After estimating the embedding parameter, for every pixel pair (a, b) defined in neighbourhood set φ(p, q), DCV (Diamond Characteristic Value) is computed by the following equation

$$\varphi(p,q) = \left\{(a,b) \| a-p| + |b-q| \le k\right\} \tag{10}$$

$$DCV(a,b) = ((2k+1)a + b) \bmod b \tag{11}$$

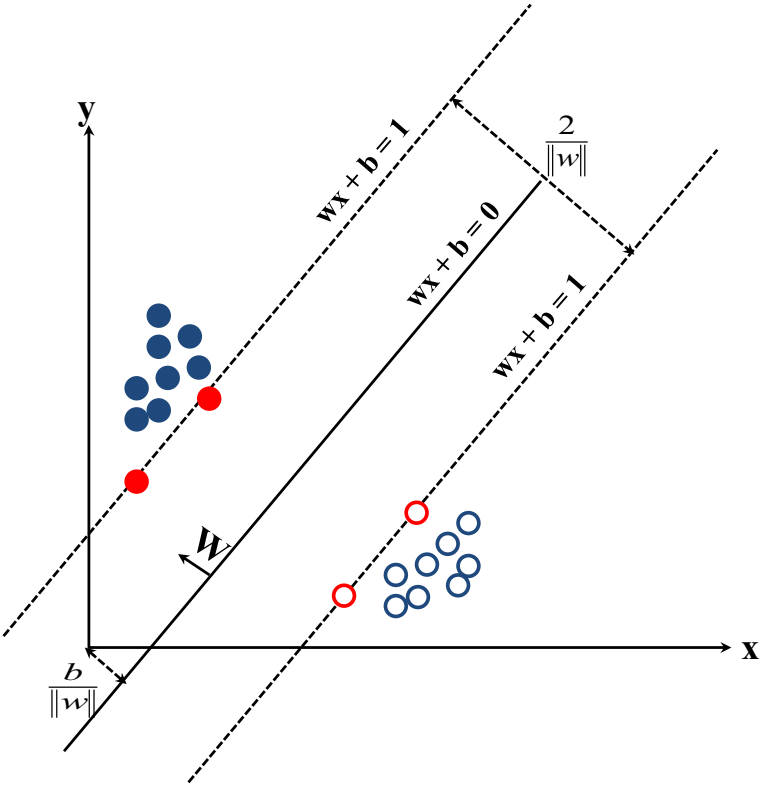


**Fig. 2** SVM classifier with hyperplane and classes

The important features of φ(p, q) are every DCV should be unique and a member of φ(p, q). The embedding is performed by changing the difference values of pixel pairs. As each pixel pair of stego image holds $\log_2 b$ bits, the maximum payload embedding capacity of the DE is $(1/2)\log_2 b$ bits per pixel. To conceal the digit $S_b$ of a secret message within the pixel pair (p, q), DCVs are examined to discover the coordinates (p′, q′), where DCV(p′, q′) = $S_b$. Then pixel pair (p′, q′) is replaced in place of (p, q). This type of substitution causes underflow or overflow problem and to overcome such problem, the values of p′ and q′ are modified into p″ and q″ as follows

$$p'' = \begin{cases} p' - b, & \text{if } p' > 255; \\ p' + b, & \text{if } p' < 0; \end{cases} \tag{12}$$

$$q'' = \begin{cases} q' - b, & \text{if } q' > 255; \\ q' + b, & \text{if } q' < 0; \end{cases} \tag{13}$$

In the extraction phase, both DCV(p′, q′) and $S_b$ from DCV(p′, q′) is computed. In DE method, the distortion occurred in the cover image during the embedding process does not exceed embedding parameter k. If the embedding parameter increases while embedding large payloads, then the distortion occurs is also large which can be detected easily by steganalysis. So the embedding parameter should be as small as possible.

### 3.5 Cloud computing

Flexibility and cost-effectiveness are provided through cloud computing architecture for many services on the Internet. Cloud computing has gained momentum because of the following factors: elimination of planning overhead for the novice user by providing facilities which are available on-demand, lower initial investment cost by using the infrastructure of the cloud and pay as they use, etc. The cloud gives a major boost to Information Technology as it facilitates a new model of competing over computing resources i.e., bandwidth, processing power, storage capacity and so on. Prior to submitting content into the cloud, content is encrypted using an encryption key and then uploaded into the cloud. All cloud users can access the encrypted data so long as they possess the valid key to decrypt; at the same time, unauthorized users cannot access the information, as shown in Fig. 3. According to the National Institute of Standards and Technology (NIST), there are three cloud models. Through Software as a Service (SaaS), users can make use of some software or application deployed in the cloud. The Platform as a Service (PaaS) provides computing environment for the user to deploy their application or programs. Infrastructure as a Service (IaaS) provides infrastructure to run different operating systems or sophisticated applications.

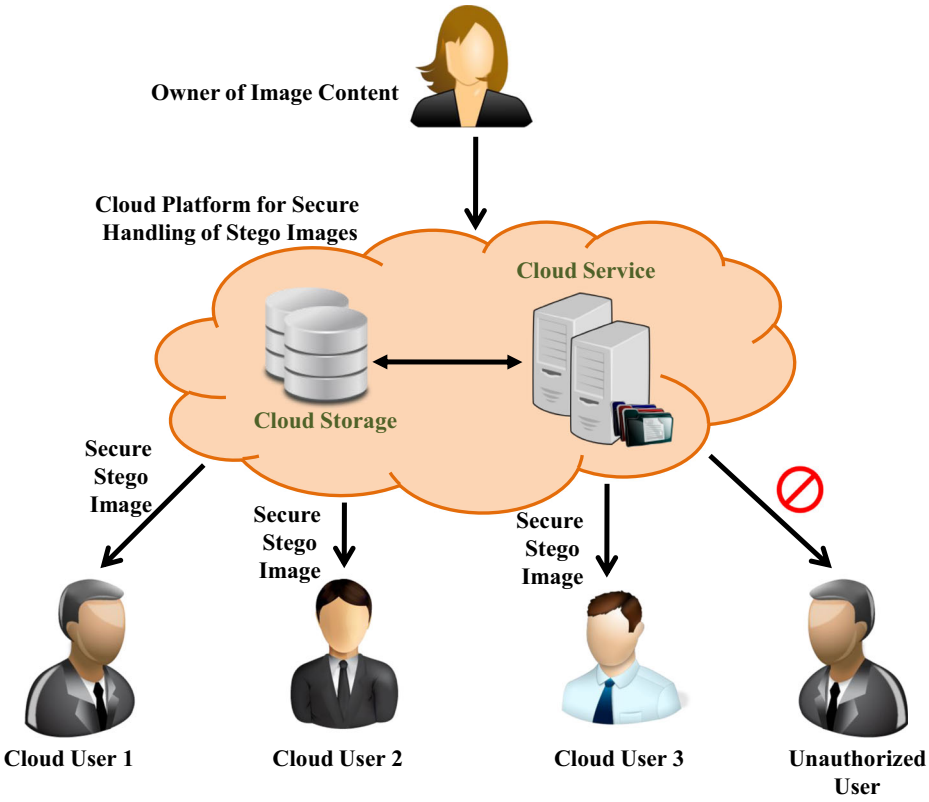


**Fig. 3** Cloud architecture for secure image sharing

Even though the cloud provides many benefits, some security issues are also present. Once data is stored in the cloud, Cloud Service Provider (CSP) has all the control over the data. Either intentionally or accidentally, data may be deleted or shared by any unauthorized user. Sometime the CSP himself may steal confidential data like credit card numbers, aadhaar numbers, and mobile numbers, etc. So, the CSP cannot always be trusted.

The user of the cloud can take some measures to protect the confidentiality of essential information. Before uploading data to a cloud server, encryption can be done over the data. But if the CSP wants to take secret information from the user data and if the user data are encrypted, the CSP may become suspicious due to the encryption. To avoid this issue, image steganography can be employed where multimedia contents are embedded into cover images to produce stego images. These stego images can be uploaded to the cloud server. The cloud user can download these stego images back from the server and then do an extraction operation from the stego image and produce the original multimedia contents.

## 4 Proposed method

In this section, a novel steganographic scheme based on diamond encoding(DE) scheme and Discrete Rajan Transform (DRT) is presented in Fig. 4. The proposed scheme embeds the coefficient of the secret image which is transformed using DRT into coefficients of the cover image which is transformed using Integer Wavelet Transform. Then inversion of the Integer wavelet transform is done. Stego images thus produced are stored in the cloud. When the secret image is needed, the stego image is downloaded from the cloud. It is first subjected to Integer wavelet transform. The transformed image goes through DE extraction. The image thus produced is subjected to the inverse of DRT which reveals the secret image. The proposed scheme reduces the distortion created in the stego image by exploiting DE.

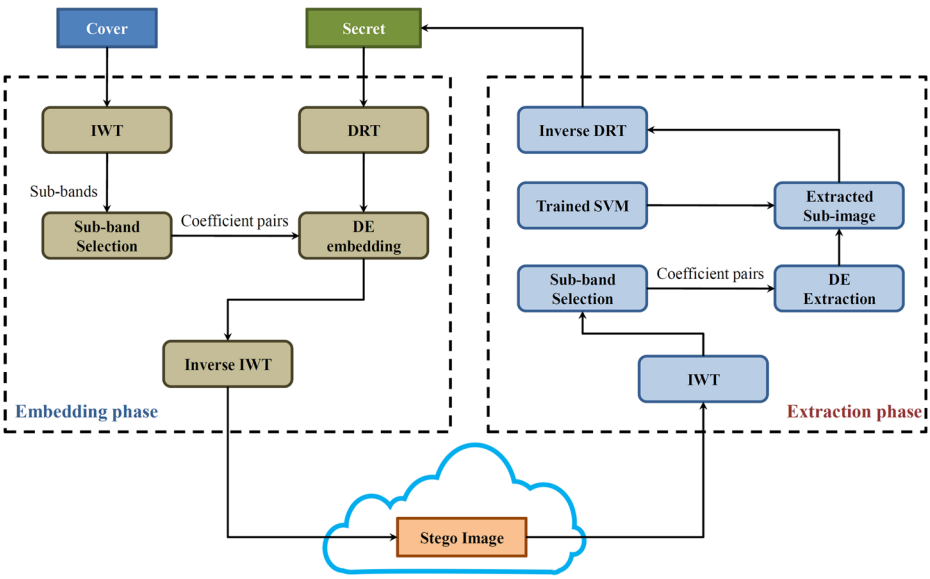


**Fig. 4** Architecture for embedding and extraction operations of our proposed steganographic approach

### 4.1 Embedding secret image

Let C be a cover image of size M × N and S be the secret image which needs to be embedded of size m × n. Following are steps for the embedding phase:

Pseudo code for embedding secret image in the cover image

**Input:** $M \times N$ sized Cover image $C$, $m \times n$ sized secret image $S$, and $r_1$, $r_2$, and $r_3$ are random seeds
**Output:** Stego-image $C'$

| | |
|---|---|
| Step 1: | Secret image $S$ is transformed using Discrete Rajan Transform. The transformed image is subjected to base $b$ conversion based on the embedding parameter $k$. |
| Step 2: | At first, cover image $C$ undergoes integer wavelet transform to divided it into three detailed sub-bands LH, HL, HH and one approximation sub-band LL. Then LL, LH, and HL sub-bands are preferred for embedding because embedding on these sub-bands produce less distortion. |
| Step 3: | $C$ is split into nonoverlapping blocks of two pixels each (Block size is 2 × 1 pixel). |
| Step 4: | Diamond Encoding is utilized to conceal medical image pixels within the pixels of cover image based on DCV value of neighbourhood sets. Location of coefficients in LL, LH, and HL are determined by random seeds r1, r2 and r3 respectively. |
| Step 5: | Finally, the inverse Integer Wavelet Transform is executed to generate stego image C′. |

The problem of extraction of the secret image from the stego image is examined as a type of binary classification. Embedded secret image components are associated with some of the particular patterns of singular values. There is a possibility of a non-linear functional relationship occurring between two singular values of the cover image and secret image bits $W_i \in \{0,1\}$. As SVM efficiently applies non-linear manipulation relationships, it is a good choice to understand the relationship. Moreover, many kinds of image processing attacks upon stego images are possible. So, the design of extraction of the secret image must include a solution to address the problem of extraction of corrupted stego images. As the usage of SVM has a good ability to generalize, our scheme uses SVM for extracting secret images from stego images.

### 4.2 Retrieving secret image

Once the stego-image C′ and the embedding parameters k, and the random seeds r1, r2, and r3 are obtained, the secret image can be readily extracted from C′. The extraction steps are listed as follows:

Pseudo code for reconstructing confidential image from stego image

**Input:** A stego-image $C'$, random seeds $r_1$, $r_2$, and $r_3$ and embedding parameters $k$.
**Output:** Secret image $S$.

| | |
|---|---|
| Step 1: | Stego-image $C'$ is initially subjected to integer wavelet transform. |
| Step 2: | The transformed image is partitioned into four sub-bands, namely, LL, LH, HL, and HH sub-band. The sub-bands LH, HL and LL are used for the extraction process. |
| Step 3: | Pixel values are extracted from subbands LL, LH, and HL according to random seeds r1, r2, and r3 respectively from the image generated in step 2. The extracted pixel values are subjected to reverse base b conversion. |
| Step 4: | SVM training pattern Ψ is created by manipulating feature sets and singular values where embedding is performed previously. |

$\psi = \{(x_i, y_i) \in R^N \times R |\ i = 1, 2, \cdots, L_r\}$

where $x_i$ indicates feature set and $y_i$ indicates singular values and $R$ characterizes the preferred output. An optimal solution can be modelled as follows:

$$\max \sum_{i=1}^{L_r} \alpha_i - \frac{1}{2} \sum_{i=1}^{L_r} \sum_{j=1}^{L_r} \alpha_i \alpha_j y_i y_j K(x_i x_j)$$

$$\text{exposed to } \sum_{i=1} \alpha_i \alpha_j 0, \quad 0 \le \alpha_i \le C, \quad i = 1, 2, \cdots, L_r$$

Pseudo code for reconstructing confidential image from stego image

Here, $\alpha_i$(i = 1,2,…,$L_r$) denotes the training parameter, $C$ represents the control parameter, and $y$ indicates decision function

$y = \sum_{i=1}^{L_r} \alpha_i y_i K(x_i, x) + b$ where b ϵ R is the bias.

Step 5: Required factors are calculated for each block

$x_u^{'} = (x_i^{'}, y_i^{'}) \in \psi^{'}$, where $u = 1, 2, \cdots, L_r$ and $y_u^{'} = f(x_u^{'})$

And stego image is computed as

$$W_u^{'} = \begin{cases} 1, & if\ y_u^{'} = 1 \\ 0, & f\ y_u^{'} = -1 \end{cases}$$

Step 6: The generated image is subjected to inverse DRT to produce secret image S.

## 5 Experimental results and analysis

In this section, the performance of our proposed method is compared with the similar schemes like Nag et al. [27], Hemalatha et al. [28], Verma et al. [29] and Atawneh et al. [30]. To assess

**Fig. 5** Cover images of size 512 × 512


**Fig. 6** Original Secret images

the performance of our scheme with other schemes, sixteen cover images of size 512 × 512, four from benchmark images from the UCID dataset and twelve non-benchmark images are considered as shown in Fig. 5 and two secret images of size 128 × 128 and 256 × 256 as

**Fig. 7** Stego images


**Fig. 8** Extracted Secret images

shown in Fig. 6. Figure 7 shows the produced stego images, Fig. 8 shows the extracted secret images. The proposed scheme is applied to practical images like military and bio-medical images as well. As the research community is accustomed to utilizing images provided in benchmark databases, Effectiveness of our approach is tested with standard images than practical images. The effectiveness of the steganographic system is determined by the imperceptibility, structural similarity, strength and robustness of the stego images produced by our proposed scheme. These four characteristics are discussed in section 5.1 through 5.4.

### 5.1 Imperceptibility

Imperceptibility is the measure of the quality of produced stego images. The amount of distortion occurring in stego images is measured using PSNR. It is computed using the following formula:

**Table 1** PSNR of our scheme for sixteen test images

| Cover Image | Secret image1 | Secret Image2 |
|---|---|---|
| #1 | 53.25 | 50.11 |
| #2 | 53.29 | 50.21 |
| #3 | 53.67 | 50.53 |
| #4 | 53.42 | 50.52 |
| #5 | 53.41 | 50.36 |
| #6 | 53.56 | 50.57 |
| #7 | 53.62 | 50.59 |
| #8 | 53.58 | 50.50 |
| #9 | 53.49 | 50.44 |
| #10 | 53.69 | 50.59 |
| #11 | 53.72 | 50.28 |
| #12 | 53.63 | 50.38 |
| #13 | 53.43 | 50.48 |
| #14 | 53.49 | 50.41 |
| #15 | 53.51 | 50.35 |
| #16 | 53.64 | 50.53 |
| Average | 53.52 | 50.43 |

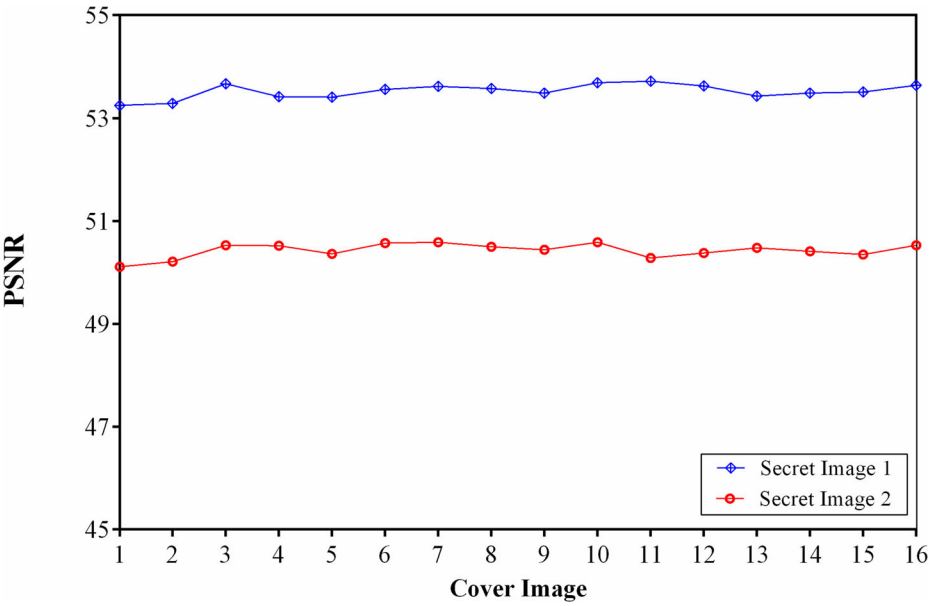


**Fig. 9** PSNR values of sixteen test images

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{14}$$

Where MSE denotes the mean squared error between cover and stego images determined by the following formula:

$$\mathrm{MSE} = \frac{\sum_{p=0}^{m-1}\sum_{q=0}^{n-1}\left(x(p,q)-\left(x'(p,q)\right)^2\right.}{mXn} \tag{15}$$

**Table 2** Comparison of PSNR of Secret image1 of our scheme with other schemes

| Cover Image | Nag et al. | Hemalath et al. | Verma et al. | Atawneh et al. | Ours |
|---|---|---|---|---|---|
| #1 | 48.10 | 46.09 | 37.60 | 52.12 | 53.25 |
| #2 | 47.91 | 45.96 | 37.87 | 52.09 | 53.29 |
| #3 | 48.05 | 46.02 | 37.94 | 52.11 | 53.67 |
| #4 | 48.05 | 46.06 | 37.83 | 52.31 | 53.42 |
| #5 | 48.29 | 46.18 | 39.04 | 52.10 | 53.41 |
| #6 | 48.06 | 46.01 | 37.81 | 52.62 | 53.56 |
| #7 | 48.14 | 46.10 | 38.06 | 52.12 | 53.62 |
| #8 | 48.28 | 46.21 | 38.29 | 52.10 | 53.58 |
| #9 | 48.11 | 46.08 | 38.05 | 52.78 | 53.49 |
| #10 | 48.98 | 46.92 | 38.16 | 52.43 | 53.69 |
| #11 | 48.49 | 46.18 | 38.23 | 52.25 | 53.72 |
| #12 | 48.28 | 46.28 | 38.43 | 52.87 | 53.63 |
| #13 | 48.17 | 46.85 | 38.91 | 52.38 | 53.43 |
| #14 | 48.21 | 46.28 | 38.69 | 52.21 | 53.49 |
| #15 | 48.93 | 46.27 | 38.72 | 52.29 | 53.51 |
| #16 | 48.95 | 46.92 | 38.24 | 52.98 | 53.64 |
| Average | 48.31 | 46.27 | 38.24 | 52.37 | 53.52 |

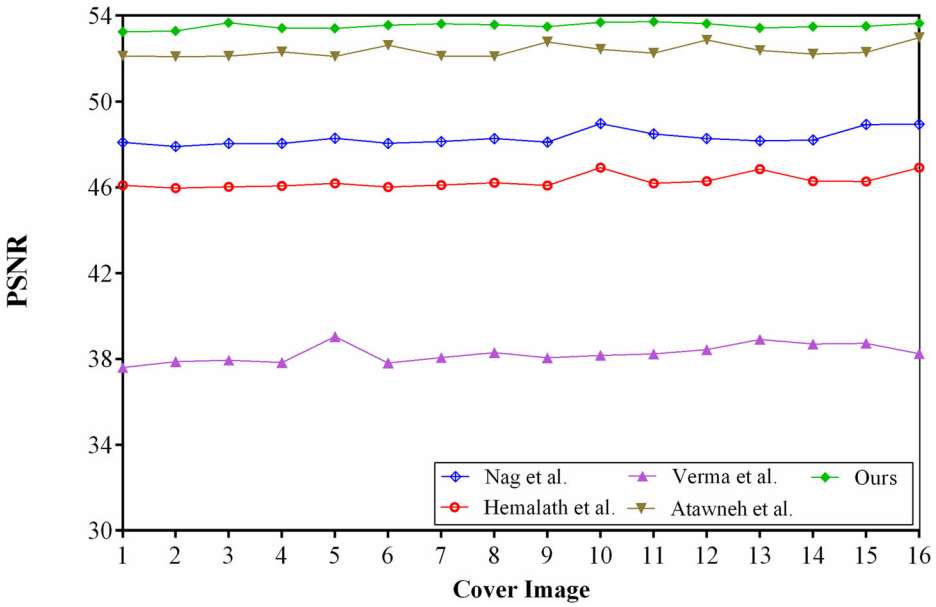


**Fig. 10** Comparison of PSNR values for secret image1

Table 1 lists PSNR values produced by our scheme for stego images of 128 × 128 and 256 × 256. Average PSNR values for the image of size 128 × 128 and 256 × 256 are 53.52 and 50.43 respectively. Figure 9 shows the PSNR values chart produced by our scheme for sixteen images. From the figure, it is shown that our scheme produces stego images of good quality, higher than the acceptable PSNR value of similar schemes.

Table 2 shows PSNR of the stego images for sixteen cover images produced by ours as well as by the other schemes with secret images of size 128 × 128. Figure 10 shows the comparison of PSNR for secret image1. Average PSNR produced by Nag et al. is 48.31, Hemalatha et al.

**Table 3** Comparison of PSNR of Secret image2 of our scheme with other scheme

| Cover Image | Nag et al. | Hemalath et al. | Verma et al. | Atawneh et al. | Ours |
|---|---|---|---|---|---|
| #1 | 44.36 | 42.22 | 34.16 | 49.14 | 50.11 |
| #2 | 44.35 | 42.19 | 34.18 | 49.15 | 50.21 |
| #3 | 44.34 | 42.22 | 34.25 | 49.13 | 50.53 |
| #4 | 44.32 | 42.18 | 34.16 | 49.11 | 50.52 |
| #5 | 44.34 | 42.22 | 34.68 | 49.25 | 50.36 |
| #6 | 44.34 | 42.20 | 34.14 | 49.16 | 50.57 |
| #7 | 44.33 | 42.25 | 34.83 | 49.15 | 50.59 |
| #8 | 44.31 | 42.20 | 34.22 | 49.12 | 50.50 |
| #9 | 44.34 | 42.21 | 34.33 | 49.15 | 50.44 |
| #10 | 44.23 | 42.33 | 35.47 | 49.31 | 50.59 |
| #11 | 44.36 | 42.87 | 35.29 | 49.91 | 50.28 |
| #12 | 44.27 | 42.35 | 35.39 | 49.19 | 50.38 |
| #13 | 44.34 | 42.44 | 35.41 | 49.40 | 50.48 |
| #14 | 44.56 | 42.48 | 35.57 | 49.63 | 50.41 |
| #15 | 44.85 | 43.21 | 35.30 | 49.20 | 50.35 |
| #16 | 44.12 | 42.29 | 35.40 | 48.80 | 50.53 |
| Average | 44.36 | 42.35 | 34.80 | 49.23 | 50.43 |

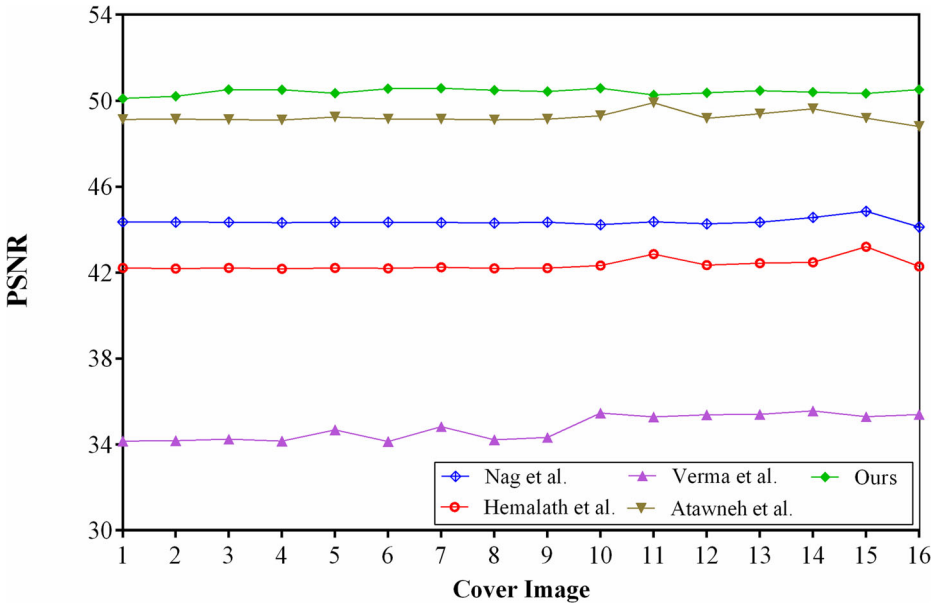


**Fig. 11** Comparison of PSNR values for secret image2

are 46.27 and Verma et al. are 38.24. These three schemes produce images with less quality, but the average PSNR for Atawneh et al. scheme is 52.37 which is better than the previous three. However, ours is 53.52, which is better than the other four schemes. Table 3 shows PSNR of the stego images for sixteen cover images produced by ours as well as by the other schemes with secret images of size 256 × 256. Figure 11 shows the comparison of PSNR for secret image2. Average PSNR produced by Nag et al. is 44.36, Hemalatha et al. are 42.35 and Verma et al. are 34.80. These three schemes produce images with less quality, but Average PSNR for Atawneh et al. scheme is 49.23 which is better than the previous three. However,

**Table 4** SSIM of our scheme for sixteen test images

| Image | Secret image 1 (65,536 bits) | Secret image 2 (131,072 bits) |
|---|---|---|
| #1 | 0.9851 | 0.9837 |
| #2 | 0.9832 | 0.9821 |
| #3 | 0.9848 | 0.9817 |
| #4 | 0.9858 | 0.9831 |
| #5 | 0.9846 | 0.9824 |
| #6 | 0.9853 | 0.9827 |
| #7 | 0.9874 | 0.9849 |
| #8 | 0.9865 | 0.9857 |
| #9 | 0.9853 | 0.9829 |
| #10 | 0.9824 | 0.9821 |
| #11 | 0.9848 | 0.9842 |
| #12 | 0.9869 | 0.9861 |
| #13 | 0.9852 | 0.9847 |
| #14 | 0.9855 | 0.9841 |
| #15 | 0.9895 | 0.9891 |
| #16 | 0.9853 | 0.9848 |
| Average | 0.9855 | 0.9841 |

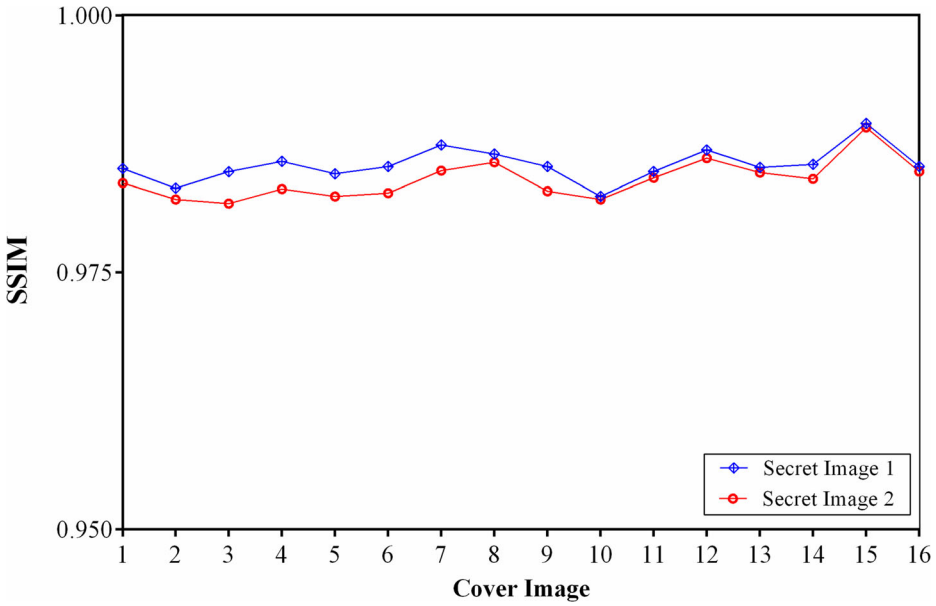


**Fig. 12** SSIM values for all sixteen stego images for both secret images

ours is 50.43, which is better than the other four schemes. For both secret images of size 128 × 128 and 256 × 256, PSNR values for our scheme is greater than the other schemes. Thus, it proves that our scheme produces stego images of high quality.

### 5.2 Structural similarity

Structural similarity index measure (SSIM) is a human visual system-based quality measure to assess image distortion in structural information. This is measured by the following formula:

$$\mathrm{SSIM}\ (\mathrm{X},\mathrm{Y}) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)} \tag{16}$$

A hundred pairs of input cover images and output stego images are considered to test similarity for our scheme as well as the other four schemes. If two images are structurally equal, SSIM value is equal to 1, otherwise, it may vary between 0 and 1. Table 4 depicts the SSIM value produced by our scheme for sixteen images of size 128 × 128 and 256 × 256. SSIM for all sixteen stego images for both secret images is shown diagrammatically in Fig. 12. Stego

**Table 5** Comparison of SSIM of our scheme with other four schemes

| Scheme | Secret image1 | Secret Image2 |
|---|---|---|
| Nag et al. | 0.9726 | 0.9646 |
| Hemalatha et al. | 0.9789 | 0.9733 |
| Verma et al. | 0.9684 | 0.9694 |
| Atawneh et al. | 0.9862 | 0.9848 |
| Ours | 0.9889 | 0.9878 |

**Table 6** Comparison of Steganalysis result (Test error) of ours and other schemes

| Scheme | Secret image1 | Secret Image2 |
|---|---|---|
| Nag et al. | 0.1120 | 0.0578 |
| Hemalatha et al. | 0.0820 | 0.0455 |
| Verma et al. | 0.0323 | 0.0035 |
| Atawneh et al. | 0.3013 | 0.2965 |
| Ours | 0.4218 | 0.3042 |

images produced for the secret image of size 128 × 128 are much closer to 1 than those of the secret image of size 256 × 256. Table 5 shows the average SSIM produced by our scheme as well as by the other four schemes. From a human visual system point of view, our scheme produces similarity of around 98.89% for the image of size 128 × 128 and 98.78% for the image of size 256 × 256, which is slightly higher than the other schemes: for Nag et al. is 97.26% and 96.46%, for Hemalath et al. is 97.89% and 97.33%, for Verma et al. is 96.44% and 96.94% and for Atawneh et al. is 98.62% and 98.48%. Thus, it proves that our scheme is producing visually similar stego images.

### 5.3 Resistance strength to steganalysis

This section proves the strength of produced stego images. The classifier is used to assess our scheme. For classifier, thousand arbitrary images of size 512 × 512 were used as cover images. All the cover images are subjected to embedding using our scheme and by the other four schemes, producing 5 sets of stego images. Stego images from each set are steganalysed separately. The output of this test is depicted in Table 6. The failure rate of the set of stego

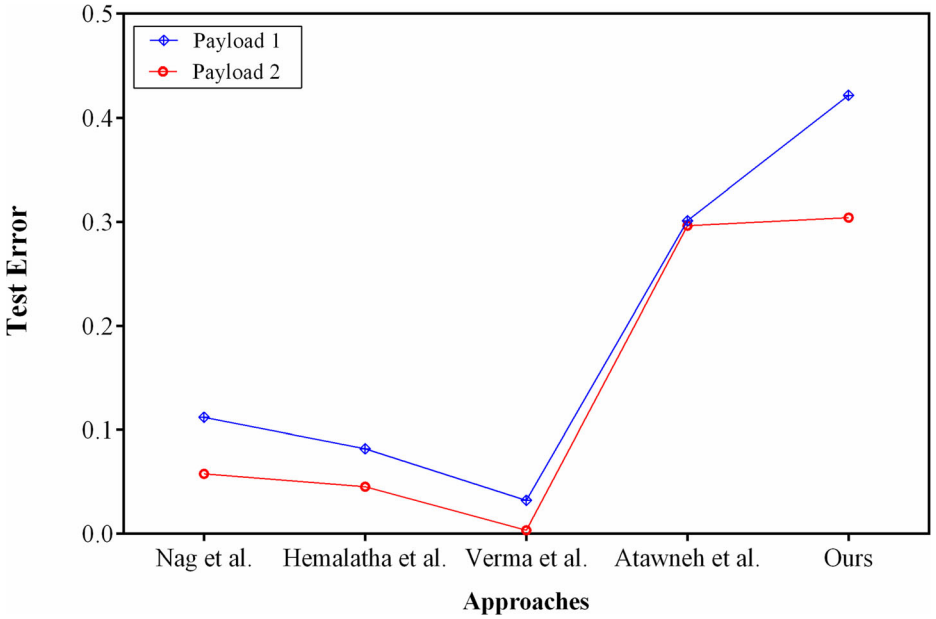


**Fig. 13** Comparison of Failure rate of ours with other scheme

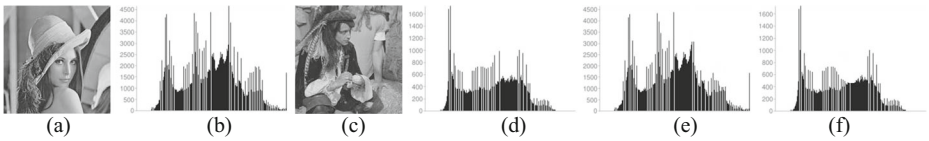


**Fig. 14** Comparison of Histogram of cover images and its stego image

images produced by our scheme is 42.18% for 128 × 128 and 30.42% for 256 × 256 sized images. This is superior over the other four schemes. Usage of Discrete Rajan Transform increases the strength of stego images. This is shown diagrammatically in Fig. 13. Thus, it proves that our scheme is strong enough to produce stego images with more strength which can withstand steganalytic attacks.

Steganlyser may also compare the histogram of the stego image with its original cover images(in the case cover image is taken from the benchmark data set). If the difference in the histogram is witnessed, then steganalyser would learn that some kind of operation is performed on the secret image which may induce them to steganalyse. Fig. 14 gives a comparison of the histogram. (a) and (b) are the actual cover image and its histogram respectively, (c) and (d) is the original secret image and its histogram respectively, (e) is the histogram of stego image and (f) is the histogram of the extracted secret image. From the Fig. 14b and e, it is evident that histogram of stego image and its corresponding cover image are almost identical which makes Histogram attack impractical.

### 5.4 Robustness

This section proves the robustness of our proposed scheme against all possible attacks. These attacks are salt and pepper noise, Gaussian noise, cropping and compression. Each secret image is first embedded into all other cover images, and the resultant image is subjected to noise and other image processing attacks. For salt and pepper noise, noise density of 0.05 is added to the stego image. For Gaussian noise, noise with a mean of 0 and variance of 0.000001 is added to the stego images. In a cropping attack, 30 × 30-pixel values are cropped from the stego image. In compression, the stego image is compressed in a lossless manner. Then in all the above cases, original images are recovered almost exactly with a meager trace of the presence of noise or effects of image processing attacks. This is evident from the stego images which were subjected to different image processing attacks as shown in Fig. 15 and from MSSIM, NCC and BER values between the extracted secret image and the original secret images as shown in Table 7. MSSIM and NCC values have to close to be 1 to judge that no


**Fig. 15** Stego images under different image processing attacks

**Table 7** MSSIM, NCC and BER values between the extracted secret image and the original secret images

| Attacks | MSSIM | NCC | BER |
|---|---|---|---|
| salt and pepper noise | 0.7467 | 0.9754 | 0.0124 |
| Gaussian noise | 0.7173 | 0.9556 | 0.0134 |
| cropping | 0.6784 | 0.9431 | 0.0161 |
| Compression | 0.7047 | 0.9532 | 0.0147 |

error is introduced in the extracted secret images, Its average values are 0.7118 and 09568 respectively in our experimental result. BER value have to be close to being 0 to judge the amount of error is introduced in the extracted secret images, the average value is 0.0141 in our experimental result. Usage of a Support Vector Machine increases the robustness of the extraction of stego images. Thus, this proves that our scheme is robust enough to withstand different kinds of attacks on the stego images.

## 6 Conclusion and future works

A new scheme for providing multimedia security in cloud space is presented in this work. In the proposed scheme, the multimedia content was first subjected to DRT transformation which was done to increase the strength of our proposed scheme. Then it was converted to Base B, which was used to increase the embedding rate. The cover image was then subjected to Integer Wavelet Transform. Base B converted coefficients of the transformed multimedia content were embedded into the coefficients of the transformed cover image using diamond embedding method. The produced image was subjected to the inverse of IWT to produce the stego image which was later stored in cloud space. When in-demand, these stego images were downloaded from the cloud, and the original multimedia content was extracted from them, using a similar procedure. The experimental results reveal that the produced stego image was structurally similar to the cover image. It also shows that our scheme is robust against many attacks. The experimental results prove the superior performance of our proposed approach over the existing methods in the literature. In the future, we plan to enhance our work with a color image steganographic technique for cloud storage.

## References

1. Hon WK, Millard C (2018) Banking in the cloud: part 1–banks' use of cloud services. Comput Law Secur Rev 34(1):4–24
2. Zafar F, Khan A, Malik SUR, Ahmed M, Anjum A, Khan MI, Jamil F (2017) A survey of cloud computing data integrity schemes. Design challenges, taxonomy and future trends. Comput Secur 65:29–49
3. Chang V (2018) An overview, examples, and impacts offered by emerging services and analytics in cloud computing virtual reality. Neural Comput & Applic 29(5):1243–1256
4. Hallman R, Rohloff K, & Chang V (2017) Workshop on multimedia privacy and security. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, pp 2645–2646
5. Kuo CT, Chi PW, Chang V, Lei CL (2018) SFaaS: keeping an eye on IoT fusion environment with security fusion as a service. Futur Gener Comput Syst 86:1424–1436
6. Duncan B, Whittington M, & Chang V (2017) Enterprise security and privacy: why adding IoT and big data makes it so much more difficult. In: 2017 International conference on engineering and technology (ICET). IEEE, pp 1–7

7. Hong JB, Nhlabatsi A, Kim DS, Hussein A, Fetais N, Khan KM (2019) Systematic identification of threats in the cloud: a survey. Comput Netw 150:46–69
8. Thanki Rohit, Vedvyas Dwivedi, and Komal Borisagar (2017) A hybrid watermarking scheme with CS theory for security of multimedia data. Journal of King Saud University-Computer and Information Sciences
9. Kadhim IJ, Premaratne P, Vial PJ, Halloran B (2019) Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research. Neurocomputing 335:299–326
10. Subhedar MS, Mankar VH (2014) Current status and key issues in image steganography: a survey. Comput Sci Rev 13:95–113
11. Atawneh S, Almomani A, Sumari P (2013) Steganography in digital images: common approaches and tools. IETE Tech Rev 30(4):344–358
12. Barve S, Nagaraj U, Gulabani R (2011) Efficient and secure biometric image stegnography using discrete wavelet transform. Int J Comput Sci Commun Netw 1(1):96–99
13. Fridrich J (2008) Detection of double-compression in JPEG images for applications in steganography. IEEE Trans Inf Forensics Secur 3(2):247–258
14. Li C-H, He-fei L, Zheng-ding L, (2007) Semi-fragile watermarking based on SVM for image authentication, IEEE International Conference on Multimedia and Expo, Beijing, China, 1255–1258.
15. Fan L, Gao T, Cao Y (2013) Improving the embedding efficiency of weight matrix-based steganography for grayscale images. Comput Electr Eng 39(3):873–881
16. Srinivasan B, Arunkumar S, Rajesh K (2015) A novel approach for color image, steganography using nubasi and randomized, secret sharing algorithm. Indian J Sci Technol 8:228
17. Arunkumar S, Subramaniyaswamy V, Karthikeyan B, Saravanan P, Logesh R (2018) Meta-data based secret image sharing application for different sized biomedical images. Biomed Res:29
18. Dumitrescu S, Wu X, Wang Z (2002) Detection of LSB steganography via sample pair analysis. In: International workshop on information hiding. Springer, Berlin/Heidelberg, pp 55–372
19. Ker AD (2005) Steganalysis of LSB matching in grayscale images. IEEE Signal Process Lett 12(6):441–444
20. Mielikainen J (2006) LSB matching revisited. IEEE Signal Process Lett 13(5):285–287
21. Randa A, Ghanbari M (2018) A high payload steganography mechanism based on wavelet packet transformation and neutrosophic set. J Vis Commun Image Represent 53:42–54
22. Lin YK (2014) A data hiding scheme based upon DCT coefficient modification. Comput Stand Interfaces 36(5):855–862
23. Subhedar MS, Mankar VH (2016) Image steganography using redundant discrete wavelet transform and QR factorization. Comput Electr Eng 54:406–422
24. Houssein EH, Ali MA, Hassanien AE (2016) An image steganography algorithm using haar discrete wavelet transform with advanced encryption system. In: 2016 federated conference on computer science and information systems (FedCSIS). IEEE, pp 641–644
25. Sohal AS, Sandhu R, Sood SK, Chang V (2018) A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments. Comput Secur 74:340–354
26. Chang V, Ramachandran M, Yen N, Walters RJ, Wills G (2016) The second international workshop on enterprise security. In: Proceedings-IEEE 7th International Conference on Cloud Computing Technology and Science, CloudCom. pp 16–17
27. Nag A, Biswas S, Sarkar D, Sarkar PP (2011) A novel technique for image steganography based on DWT and Huffman encoding. Int J Comput Sci Secur 4(6):497–610
28. Hemalatha S, Acharya UD, Kamath PR (2013) A secure color image steganography in transform domain. Int J Cryptography Inf Secur 3(1):17–24
29. Verma A, Nolkha R, Singh A, Jaiswal G (2013) Implementation of image steganography using 2-level DWT technique. Int J Comput Sci Bus Inf 1(1)
30. Atawneh S, Almomani A, Al Bazar H, Sumari P, Gupta B (2017) Secure and imperceptible digital image steganographic algorithm based on diamond encoding in DWT domain. Multimed Tools Appl 76(18): 18451–18472
31. Mandalapu EN, Rajan EG (2009) Rajan transform and its uses in pattern recognition. Informatica 33(2): 205–213
32. Tsai HH, Sun DW (2007) Color image watermark extraction based on support vector machines. Inf Sci 177(2):550–569

**Arunkumar Sukumar** is currently working as Assistant Professor in School of Computing, SASTRA Deemed University, India. He received his Bachelor of Engineering and Master of Engineering from Bharathidasan University, Tiruchirappalli and Anna University, Chennai respectively. He is currently pursuing Ph.D. in the School of Computing, SASTRA Deemed University, Thanjavur, India. He has more than 10 years of teaching experience and his area of research includes image processing, cloud, network security and steganography. He is also the author/co-author of many papers in conferences and journals of international repute.

**Subramaniyaswamy V** received the B.E. degree in Computer Science & Engineering and M.Tech. degree in Information Technology from Bharathidasan University, India and Sathyabama University, India respectively. He received his Ph.D degree from Anna University, India and currently working as an Associate Professor in the School of Computing, SASTRA University, India. He has published more than 75 papers in reputed international journals and conferences. His research interests include Data Mining, Recommender Systems, and Big Data Analytics.

**Vijayakumar V** is currently working as Professor in VIT University Chennai Campus; He has more than 18 years of experience which includes 10 years in teaching and 8 years in the Industry. He is also a coordinator of Cloud Computing Research Group and Coordinator for Internship in India and Worldwide. In VIT, he involved in many Research and Development Activities, he has also organized many National/International Seminars/ Workshops/ Symposiums/ Conferences/Special Sessions in the area of Cloud computing and Big Data which includes ISBCC'14, ISBCC'15, ISBCC'16, ISBCC'16, and ICBCC'18 in India, CCCA'14 in Vietnam and CCNC'14 in USA. His area of research includes Grid Computing, Cloud computing, Big Data, Web semantics and also involved in the domain like biomedical application, Mammogram, Autism, Immune system and other areas like key management, security issues in cloud and grid computing.

**Logesh Ravi** received the B.Tech. degree in Computer Science & Engineering and M.Tech. degree in Networking from Pondicherry University, India. He was conferred Ph.D. in the area of Artificial Intelligence and Recommender Systems from the SASTRA (Deemed University), India. Currently, Dr. Logesh Ravi is associated with Sri Ramachandra Faculty of Engineering and Technology, Sri Ramachandra Institute of Higher Education and Research (formerly known as Sri Ramachandra University), Chennai, India. His research was funded and sponsored by Science and Engineering Research Board, Department of Science and Technology, New Delhi, India. He serves as Academic Coordinator and Associate Director (Research), and he is active participant in academic administration and research in computing. He has published more than 70 papers in reputed international journals and conferences. His research interests include Artificial Intelligence, Recommender Systems, Big Data, Machine Learning, and Social Computing, Information Retrieval, and Human-Computer Interaction. He also serves as a guest editor to many reputed international journals.

## Affiliations

**Arunkumar Sukumar[1] · V. Subramaniyaswamy[1] · V. Vijayakumar[2] · Logesh Ravi[3]**

Arunkumar Sukumar
vgsarun@gmail.com

V. Vijayakumar
vijayakumar.varadarajan@gmail.com

Logesh Ravi
LogeshPhD@gmail.com

[1] School of Computing, SASTRA Deemed University, Thanjavur, India

[2] School of Computing Science and Engineering, Vellore Institute of Technology, Chennai, India

[3] Sri Ramachandra faculty of Engineering and Technology, Sri Ramachandra Institute of Higher Education and Research, Chennai, India