Multimed Tools Appl
DOI 10.1007/s11042-016-3383-5

CISSKA-LSB: color image steganography using stego key-directed adaptive LSB substitution method

Khan Muhammad [1,2] · Jamil Ahmad [1,2] · Naeem Ur Rehman [2] · Zahoor Jan [2] · Muhammad Sajjad [2]

Received: 20 April 2014 / Revised: 31 October 2015 / Accepted: 23 February 2016

Abstract Information hiding is an active area of research where secret information is embedded in innocent-looking carriers such as images and videos for hiding its existence while maintaining their visual quality. Researchers have presented various image steganographic techniques since the last decade, focusing on payload and image quality. However, there is a trade-off between these two metrics and keeping a better balance between them is still a challenging issue. In addition, the existing methods fail to achieve better security due to direct embedding of secret data inside images without encryption consideration, making data extraction relatively easy for adversaries. Therefore, in this work, we propose a secure image steganographic framework based on stego key-directed adaptive least significant bit (SKA-LSB) substitution method and multi-level cryptography. In the proposed scheme, stego key is encrypted using a two-level encryption algorithm (TLEA); secret data is encrypted using a multi-level encryption algorithm (MLEA), and the encrypted information is then embedded in the host image using an adaptive LSB substitution method, depending on secret key, red channel, MLEA, and sensitive contents. The quantitative and qualitative experimental results indicate that the proposed framework maintains a better balance between image quality and security, achieving a reasonable payload with relatively less computational complexity, which confirms its effectiveness compared to other state-of-the-art techniques.

✉ Muhammad Sajjad
muhammad.sajjad@icp.edu.pk

Khan Muhammad
khan.muhammad.icp@gmail.com; khanmuhammad@sju.ac.kr

Jamil Ahmad
jamil.ahmad@icp.edu.pk

Naeem Ur Rehman
naeembcs1@gmail.com

Zahoor Jan
zahoor.jan@icp.edu.pk

[1] Digital Contents Research Institute, Sejong University, Seoul, Republic of Korea

[2] Department of Computer Science, Islamia College Peshawar, Peshawar, Pakistan

Published online: 07 April 2016

Keywords Image steganography · Information security · Image quality · LSB · Stego key · Multimedia security · Data hiding

1 Introduction

With increasing transmission of sensitive information over the public network "Internet", security of sensitive contents is becoming more challenging and have been enthusiastic area of research since last decades. Cryptography which is the process of encrypting sensitive information into scrambled messages, has been used as a solution to information security for ages [41]. However, the main problem in cryptographic methods is the meaningless form of encrypted messages, making them suspicious enough to attract adversaries' attention, which consequently can be modified or decrypted based on powerful cryptanalysis systems [2]. This problem can be resolved by employing information hiding methods such as "steganography", aiming to protect sensitive information during transmission while minimizing security breaches [24].

Steganography is a Greek origin word meaning *protected writing*. It is a special branch of information hiding and is considered as an art of science for invisible communication, aiming an imperceptible hiding of a secret message inside a cover image whose existence is known to the sender and receiver only [14]. The basic elements of steganography include a carrier object, a message, an embedding mechanism, and a stego key for better security. A carrier object can be an image, audio, video, and text. Steganography can be used for a wide range of applications such as safe circulation of secret data in military and intelligence agencies, improving mobile banking security, online voting security, and covert communication between two communicating bodies [7, 50]. Steganography has many fruitful applications, however, it can also be quite dangerous as hackers can utilize it for sending viruses and Trojans with intension of compromising sensitive systems. Further, this technology can also be used by terrorists and criminals for exchanging their secret information [6].

Different terminologies are used in reference to image steganography. *Host/cover image* is the original image with no hidden secret data; the resultant image with encoded secret information is referred as *stego image;* and, *stego key* is a secret key, utilized in the embedding process, increasing security. *Secret data* can be a simple text message, an image, audio, or video. *Payload* is the quantity of secret data that can be successfully hidden inside a cover object without producing visual artifacts in stego images. Payload is measured in terms of bits per pixel (bpp). The payload of a steganographic algorithm is 1bpp, if 1 bit of data is hidden in each pixel. The size of the payload is directly proportional to the strength of steganographic algorithm and vice versa [41]. The term robustness in the context of steganography describes the firmness of a steganographic algorithm against different types of simple and statistical attacks. A steganographic algorithm is considered to be more robust if the data embedded in the cover image is neither extracted nor modified easily by image processing operations, e.g., image rotation, noising, cropping, and scaling but robustness is usually addressed in watermarking techniques due to their concern with copyright protection [28]. The term *imperceptibility* refers to undetectability which is measured by various image quality assessment metrics (IQAMs) such as peak-signal-to-noise-ratio (PSNR), structural-similarity-index-metric (SSIM), and root-mean-square-error (RMSE). A steganographic method is highly imperceptible if it produces stego images with minimum possible distortion after intentionally concealing data such that it cannot be easily detected by the human visual system (HVS) [22, 26].

Considering the mechanism of data embedding, image steganography techniques are generally classified into spatial domain and transform domain. The former is based on direct modification of pixel intensities, having larger embedding capability with slight degradation of image quality. These methods are less robust as the embedded data cannot be fully recovered if stego images are exposed to image manipulation and simple attacks like filtering, cropping, compression, rotation, noise addition, and translation which is its limitation. Some spatial domain methods include LSB substitution methods [32, 39, 46, 54], pixel-value-differencing methods [51, 53], tri-way-pixel-value-differencing method [29], gray-level modification methods [1, 33], edges-based embedding methods [9, 19, 21, 31, 47], pixel indicator techniques (PIT) [51, 53], and pixel-pair-matching method [20]. The latter domain of techniques is based on utilization of transformed co-efficients for message concealment, having minimum vulnerability to various attacks. Some well-known techniques of this category include discrete wavelength transform method [10], discrete cosine transform method [43], discrete Fourier transform method [8], and integer contour transform method [16]. Transform domain methods are more robust compared to spatial domain, making them more suitable for watermarking purposes such as copyright protection [28]. The major drawback of such methods is their lower payload and huge computational complexity, failing to maitain a better balance between image quality, payload, efficiency, and security, hence making them not a favourite option for real-time security applications. With these drawbacks in mind, we have developed our framework based on spatial domain and are considering only spatial domain methods.

Since the last decade, researchers have presented a large number of spatial domain stegangoraphic methods. Least significant bit (LSB) replacement is the most well-known scheme in which the LSBs of the host image are replaced with message, producing relatively good quality marked images. However, its simplicity and imbalance modification of pixels make its detection relatively easy for steganaylsis methods [4]. This limitation is minimized by LSB-matching (LSBM) scheme [31] by adding/subtracting a numerical one to the host image's pixels based on the secret message, reducing the chances of detectability but still leaving some distortion on marked images. LSBM revisited (LSBMR) [32] improves the LSBM scheme by taking into consideration the relationship between a pair of two pixels for concealing two bits at a time, reducing the distortion rate up to 0.325 from 0.5 bpp for marked images. Luo et al. [31] further reduced the detectabilty by combining LSBMR with edge based data hiding mechanism, selecting regions of cover image adaptively for message concealment as per requirment. These reviewed schemes are susceptible to several problems such as: i) direct embedding of senstive information into the host image without any encryption consideration, enabling attackers to extract the secret messages relatively more easily once the embedding algorithm is cracked, ii) visual distortions in stego images are generated as a result of using ineffecient embedding algorithms, maximizing the chances of detection by human visual system, and iii) lack of maintaining an acceptable balance between image quality, payload, computational complexity, and security, making them less suitable for real-time and top-secret security applications.

In this paper, we address these problems by proposing an efficient framework for color images in spatial domain, utilizing adaptive LSB replacement mechanism with multi-level cryptography. The major contributions of this research work are summarized as follows:

1. A secure image steganographic framework combining the stregths of steganography and cryptography while maintaining a better balance between image quality, payload, and

security, making this framework more suitable for real-time and top-secret level security applications.

2. Encryption of sensitive information using MLEA prior to data hiding process, introducing an extra barrior for attackers, hence keeping secret information more secure even if the underlying stegangoraphic algorithm get cracked. In addition, the secret key utilized in MLEA is also encrypted using TLEA, providing relatively additonal security and making the extraction more challenging for adversaries.
3. Data hiding using stego key-direcred adaptive LSB substitution method (SKA-LSB), producing better visual quality stego images, which in turn minimizes the detectability by HVS. Furthermore, the method adaptively embeds data in different channels of the host image based on embedding key, red channel, and encrypted sensitive information, extending further its security and robustness.

The rest of the paper is organized as follows. Section 2 presents an overview of the state-of-the-art methods whose limitations become a base for our current proposed research work. The proposed framework is explained in Section 3. Experimental results and discussion are given in Section 4. Finally, the paper is summarized in Section 5 with its conclusion and future research outlines.

2 Related work

Literature study reveals that the most simple and popular method to hide secret data inside an image is LSB replacement method, where secret data is converted into binary bits, replacing LSBs of the host image. In case of gray-scale images where each pixel has only one value ranging from 0 to 255 with bit depth of 8 bits, secret data bits are directly replaced with LSBs of the cover image. In case of color images, having three channels (red, green, and blue) with bit depth of 24 bits, first, cover image is divided into three channels and each channel is then utilized for message hiding which are combined at the end, resulting in stego image. As our proposed framework uses a special variation of LSB replacement method, therefore, it is mathematically expressed with sufficient detail for better understanding of the core idea. Assume a cover image X with bit depth of 8 bits, having n pixels, represented as $X = X_0, X_1, X_2, \ldots\ldots X_{n-1}$ where $X_i \in X$ and $i \in \{0, 1, 2, \ldots\ldots n-1\}$. Assume M as a secret message, expressed as $M = M_0, M_1, M_2, \ldots\ldots M_{n-1}$ such that M_i shows a string of k-bits of the message M for $i \in \{0, 1, 2, \ldots\ldots n-1\}$. During the embedding mechanism of a message bit M_i into X_i, the pixel X_i is decomposed into two equal portions including LSB_i and MSB_i, where $X_i = MSB_i \,||\, LSB_i$ and then LSB_i is replaced with M_i for $i \in \{0, 1, 2, \ldots\ldots n-1\}$. The output of this process is the marked image Y with pixels $Y = Y_0, Y_1, Y_2, \ldots\ldots Y_{n-1}$ where $Y_i \in Y$ and $i \in \{0, 1, 2, \ldots\ldots n-1\}$, which can be then sent to the concerned receiver, transferring the secret message securely.

The idea of LSB based steganography is further explained using a simple example. Suppose X is a grayscale image consisting of eight pixels $[X = X_1, X_2, X_3, \ldots\ldots X_8]$,

having the following values for their decimal and the corresponding binary representations:

Decimal	Binary	Decimal	Binary	Secret letter	Binary
$X_1 = 141$	$(10001101)_2$	$X_2 = 40$	$(00101000)_2$	A	$(01000001)_2$
$X_3 = 130$	$(10000010)_2$	$X_4 = 132$	$(10000100)_2$		
$X_5 = 118$	$(01110110)_2$	$X_6 = 75$	$(01001011)_2$		
$X_7 = 97$	$(01100001)_2$	$X_8 = 119$	$(01110111)_2$		

Assume M as a secret message such that $M =$ "**A**" with binary representation $M = (0100001)_2$. To hide this secret message M inside the given image X, the LSBs of the pixels $[X = X_1, X_2, X_3, \ldots\ldots X_8]$ are replaced with the message bits $M = (01000001)_2$. The resultant pixels after embedding process are denoted by $Y = Y_1, Y_2, Y_3, \ldots\ldots Y_8$ with decimal and their corresponding binary values as follows:

Decimal	Binary	Decimal	Binary	Secret letter	Binary
$Y_1 = 140$	$(1000110\mathbf{0})_2$	$Y_2 = 41$	$(0010100\mathbf{1})_2$	A	$(01000001)_2$
$Y_3 = 130$	$(10000010)_2$	$Y_4 = 132$	$(10000100)_2$		
$Y_5 = 118$	$(01110110)_2$	$Y_6 = 74$	$(0100101\mathbf{0})_2$		
$Y_7 = 96$	$(0110000\mathbf{0})_2$	$Y_8 = 119$	$(01110111)_2$		

The bold faced LSBs in the pixels $Y = Y_1, Y_2, Y_3, \ldots\ldots Y_8$ represent the modified pixels, resulted from embedding process, i.e., pixels $[Y = Y_1, Y_2, Y_6, and\ Y_7]$. To increase the embedding capacity, more than 1 LSB can be used but it will degrade the visual quality of stego image which can then be easily detected by the HVS. This relationship between the number m of LSBs and the visual quality of stego images is illustrated by embedding a secret message "*Welcome to the land of Hospitality, Khyber Pakhtunkhwa, Pakistan*" in the cover image *Lena*, and the results are shown in Fig. 1.

LSB replacement method is straightforward and is vulnerable to simple attacks, therefore, Bailey and Curran nominated an extended version of this approach known as stego color cycle (SCC) [3] by embedding the message in all three channels of the color host image in a pre-determined cyclic order. The method uses one channel at a time for message hiding with channel order R, G, B, R, G, and B and so on till the end of secret message. The order indicates that red channel of $pixel_1$ will carry the 1st message bit, green channel of $pixel_2$ will carry the 2nd message bit, and blue channel

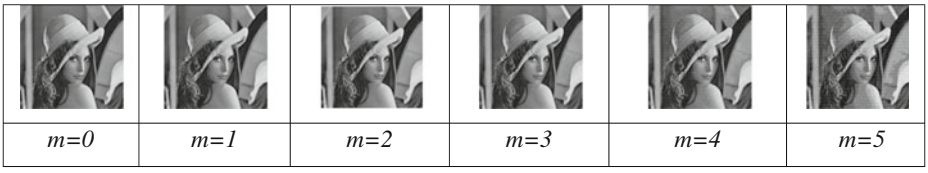

Fig. 1 Illustrating the relationship between number of LSBs ($m \in [0, 1, 2, \ldots.5]$) and visual quality of Lena stego image

of $pixel_3$ will carry the 3rd message bit, and so on. In this way, it disperses the message in three channels, making it slightly better than the simple LSB method, but its fixed cyclic order of message hiding enables attackers to easily extract the hidden information. To improve this approach, Muhammad et al. [34] proposed cyclic steganographic technique with randomization, providing relatively better security than SCC method.

SCC and Muhammad et al. [34] approaches are better than simple LSB replacement method in terms of visual quality and security; however, their payload is still small. In this context, Gutub proposed PIT [18], aiming to increase the payload of existing LSB based approaches. PIT divides cover image into data and indicator channels, where messages are embedded in the data channels as indicated by the indicator channel according to the embedding policy given in Table 1.

PIT achieves the property of robustness by keeping the indicator channel variable i.e. red, green, and blue channels acting as indicators for pixel one, pixel two, and pixel three respectively, and so on. The payload capacity of this approach is higher than LSB based schemes, however its payload decreases, when larger numbers of LSBs of the indicator channel are 00 in the cover image as shown in Table 1. The authors in [45] further improved PIT, utilizing partitioning mechanism and distributing sensitive information based on statistical theory. Karim et al. [36] nominated a new technique, improving the simple LSB method in terms of security. The embedding capacity is the same as LSB method that is 1bbp, but security gets improved as data extraction is infeasible for attackers without having correct secret key. Jassim presented a steganography five modulus method (ST-FMM) [23], which divides cover image into small blocks, having size of $K \times K$ pixels. The method disperses message in various blocks of cover image, making its extraction difficult up to some extent. However, the payload is dependent on window size, covering a limited set of characters in certain cases which in turn limits its effectiveness for various applications.

The LSB based data hiding methods discussed so far directly embed a message in a host image without consideration of smooth or edge area pixels. Tsai and Wu [53] discovered that edge-area pixels can carry more data, hence they presented the idea of edges based steganography, resulting in higher payload. Their method was further improved by authors of [9], utilizing hybrid edge detection mechanism that resulted

Table 1 Policy for embedding process [18]

Indicator channel	Data channels	
(Intermediate LSB, LSB)	Ch1	Ch2
00	No data embedding	No data embedding
01	No data embedding	2 secret bits embedding in 1st and 2nd LSB
10	2 secret bits embedding in 1st and 2nd LSB	No data embedding
11	2 secret bits embedding in 1st and 2nd LSB	2 secret bits embedding in 1st and 2nd LSB

from combination of canny and fuzzy edge detectors. The method in [9] achieves better image quality with the same payload of LSB approach, having resiliency against statistical analysis based steganalysis systems. This method was extended by authors of [21] for color images increasing further the embedding capacity. Unlike Chen et al. [9] scheme, the scheme in [21] uses Sobel edge detector instead of canny. The major weaknesses of this approach is the overhead of two separate additional files, containing embedding information such as width and height of the image, the number of bits changed in each channel of each pixel, and some other parametric data.

Grover and Mohapatra [17] resolved the problem of the scheme in [21] by incorporating an edge based adaptive approach for color images with the same aim of increasing the payload. The method has better security compared to the existing edge based schemes due to division of message into two different blocks, one for edgy pixels and another for non-edgy pixels, and traversal of image intensities from central pixels for data embedding. The edge based schemes discussed so far, produce stego images of fixed quality, limiting their applications. To resolve this issue, H.R. Kanan and B. Nazeri [26] presented a lossless spatial domain method, where the image quality is tuneable. It considers steganography as a searching problem and uses genetic algorithm for finding best positions in the cover image for message embedding, enhancing the stego quality and payload, however, it lacks security and is computationally complex.

The literature discussed so far indicates that various techniques have been used for secure transmission of secret information, focusing on payload, image quality, security, and computational complexity. The existing schemes are either too naïve or computationally too complex. The simpler methods are cost-effective, but fail to achieve better image quality and security with higher payload, restricting their applications in top-secret communication systems. On the other hand, the more sophisticated methods achieve higher payload with better visual quality and security; however, such methods are expensive in terms of computation, limiting their suitability in real-time security applications. Therefore, we propose a cost-effective image steganographic framework, which maintains a better trade-off between image quality, payload, security, and computational complexity.

3 The proposed framework

In this section, the proposed framework and its main modules are explained pictorially, making its novelty clear enough to be easily understood by the readers. The framework is proposed for color images based on steganography and multi-level cryptography. Unlike other steganographic systems, which fail to maintain an acceptable level of image quality with a reasonable payload in a cost-effective manner, our framework has the capability to keep a balance among image quality, payload, security, and computational complexity. Therefore, the proposed framework can be used for secure transmission of electronic patient records (EPR) to healthcare centres, top-secret sensitive communication between intelligence departments, and private communication, requiring privacy. The schematic representation of the proposed framework is shown in Fig. 2.

The proposed framework consists of four main sub-algorithms as follows: 1) TLEA is employed to encrypt secret key, which is then used for encryption in MLEA. 2) MLEA encrypts secret information using encrypted secret key resulted from TLEA prior to

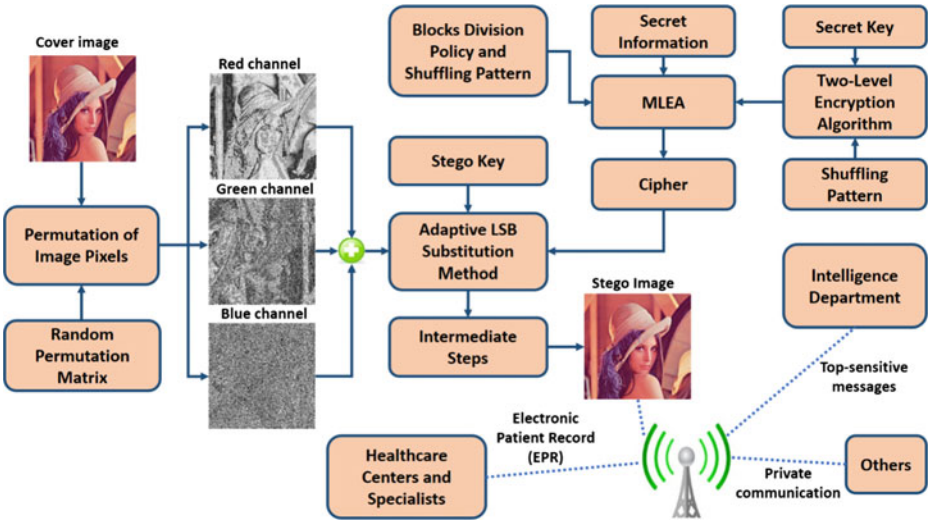

Fig. 2 Framework of the proposed system

embedding process. TLEA and MLEA are developed taking inspiration from [14] and [13] for the purpose to introduce several barriers for attackers during extraction of secret information, hence increasing security. Aziz et al. [2] recommends that secret information should be encrypted by AES algorithm prior to embedding which is used by Nguyen et al. [41], along with encryption of secret key. However, it is proved by Jinomeiq et al. [25], and Liu et al. [30] that AES is computationally expensive, hence cannot be used in real-time security applications. 3) The third embodiment is data embedding algorithm which adaptively hides encrypted secret data in cover images, resulting in stego images, which can be sent to the concerned departments and users. 4) Finally, the extraction algorithm extracts the intended information from stego image at receiver terminal, which can be then used accordingly. These four algorithms are briefly described in the subsequent sections.

3.1 Two-level encryption algorithm (TLEA)

The TLEA is a simple, but an effective algorithm, aiming to encrypt secret key, resulting in better security. It consists of two main functions; bitxor and secret pattern based bits shuffling. Although there exists various encryption algorithms for such tasks such as AES, DES, and Blowfish, such algorithms require huge computational cost, limiting their applicability in real-time security applications as indicated in [12] and [42]. Therefore, we have developed this light-weight but effective algorithm, which is incorporated in the proposed framework. To clarify the concept behind TLEA, consider the secret key, $K = 32741586$. To further simplify the example, only the first digit of secret key is encrypted i.e. $K = 3 = (00000011)_2$. Apply the bitxor function on secret key bits with logical 1, i.e., $K_1 = (K \oplus 11111111) \Rightarrow (00000011 \oplus 11111111) \Rightarrow (11111100)_2$. The second step is to apply secret pattern based bits shuffling algorithm, shuffling the binary bits of each byte in the secret key. Continuing the above example, shuffle K_1 bits according the Fig. 3.

Figure 3 shows an example of secret pattern, containing 8 digits, which is used in the encryption of secret key providing an extra layer of security. It should be noted that this pattern is not fixed and can be changed as per requirement of the security application, controlling the encryption level. In example, the K_1 bits are shuffled according to this pattern, i.e., the third bit of K_1 is swapped with the sixth bit of K_1; second bit is swapped with eighth bit; seventh bit is swapped with fifth bit; and fourth bit is swapped with first bit. For ease of understanding, we have repeated the same process for every character of secret key. The resultant bits obtained after applying this process on K_1 bits are represented by $K_2 = (10110111)_2$.

3.2 Multi-level encryption algorithm (MLEA)

The MLEA is used to encrypt the actual secret data in order to make its extraction from the stego image difficult for an attacker. It is relatively light-weight compared to AES, DES, and other complex algorithms [12], which is its motivational reason of choosing. It consists of four processes including (i) bitxor, (ii) blocks division of secret bits, (iii) secret key based shuffling, and (iv) encrypted secret key based encryption. The idea of MLEA can be explained with a simple example. Suppose *S* is a secret message S = "*B*" with binary equivalent $S = (01000010)_2$. First apply the bitxor operation, i.e., $M = (S \oplus 11111111) \Rightarrow (01000010 \oplus 11111111) \Rightarrow (10111101)_2$. The second operation is division of message bits into four blocks. There are several possibilities for this division, confusing the attacker in finding the actual pattern being used. The approach, we followed is described here as: message $block_1$ M_1 contains all the eighth and first bit of each byte of the secret bits; message $block_2$ M_2 contains all the seventh and second bit of each byte; message $block_3$ M_3 contains all the sixth and third bit of each byte; and message $block_4$ M_4 contains the fifth and fourth bit of each byte of message as described below.

M_1 = 8th bit of 1st byte of the message, 1st bit of 1st byte, 8th bit of 2nd byte, 1st bit of 2nd byte.....

M2 = 7th bit of 1st byte, 2nd bit of 1st byte, 7th bit of 2nd byte, 2nd bit of 2nd byte, 7th bit of 3rd byte, 2nd bit …....

M_3 = 6th bit of 1st byte, 3rd bit of 1st byte, 6th bit of 2nd byte, 3rd bit of 2nd byte, 6th bit of 3rd byte, 3rd bit …....

M_4 = 5th bit of 1st byte, 4th bit of 1st byte, 5th bit of 2nd byte, 4th bit of 2nd byte, 5th bit of 3rd byte, 4th bit …....

Now, concatenate the four message blocks, i.e., $MM = [M_1, M_2, M_3, M_4]$; dividing $M = (10111101)_2$ into four blocks result in $M_1 = (11)_2$, $M_2 = (00)_2$, $M_3 = (11)_2$, and $M_4 = (11)_2$, hence $MM = (11001111)_2$. The third step is to apply secret key based shuffling. Consider key = "*32741586*", shuffle the bits of

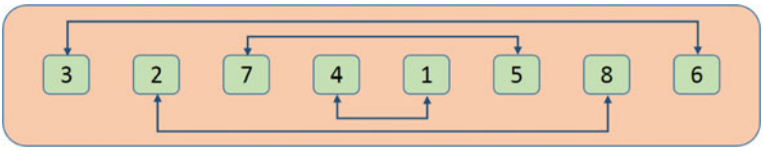

Fig. 3 An example of secret pattern used in bits shuffling

$MM = (11001111)_2$ and store the resultant bits into a variable MMM. This procedure works as follows.

a. Take the i_{th} digit from the secret key.
b. Separate the secret bit at the i_{th} digit position from MM.
c. Concatenate the separated bit with MMM and increment the value of i.
d. Repeat step (a) to (c) until all bits of MM are shuffled.

Now apply this procedure on MM bits. The first digit of secret key is 3 so the third bit from MM will be stored in MMM. Next digit of secret key is 2, so second bit of MM is concatenated with MMM. Continuing the same procedure for MM bits, the resultant bits attained are: $MMM = (01101111)_2$. The last step of MLEA is to apply the encrypted secret key based encryption on MMM. Consider $MMM = (01101111)_2$, secret key resulted from TLEA (Section 3.1) $K_2 = (10110111)_2$, and N as an array for storing the final resultant bits. This fourth sub-procedure works as follows:

a. Initialize the loop counters i and j such that i = 0 and j = 0
b. Select the i_{th} secret bit from MMM
c. Select the j_{th} bit from encrypted secret key K_2
d. If the j_{th} bit of secret key K_2 is 1
i. perform Temp = (MMM(i) ⊕ logical 1)
ii. concatenate Temp with N
Else
Concatenate MMM (i) with N without bitxor operation
End
e. Increment i and j by 1
f. Repeat step (b) to step (e) until all bits are encrypted

Apply this procedure on $MMM = (01101111)_2$ using $K_2 = (10110111)_2$. The first bit of K_2 is 1 so N (1) = (1 ⊕ 1) =0 and $N = (1)_2$. Second bit of K_2 is 0, so N becomes $N = (11)_2$. Third bit of K_2 is 1 so N (3) = (1 ⊕ 1) = 0 and $N = (110)_2$. Proceeding with the same procedure, the final bits attained are $N = (11011000)_2$. The resultant final bits (N) of MLEA clearly show that the encrypted bits are completely different from original bits, i.e., $S = (01000010)_2$ and increase the security of the proposed method. For decryption purposes, the above four steps are repeated in reverse order for obtaining the actual hidden data.

3.3 Embedding algorithm

The embedding algorithm is responsible for hiding secret information inside a cover image. It hides the encrypted message adaptively in blue or green channel on the basis of red channel's LSB and encrypted bits of secret key, following the scanning order of stego key. Algorithm 1 illustrates the major steps of the proposed embedding mechanism.

To understand the idea of the proposed embedding algorithm, consider a color host image P with pixels $[P_1, P_2, P_3, ... P_8]$ in the binary form, encrypted secret key bits (Section 3.1) $K_2 = (10110111)_2$, and encrypted secret message bits (Section 3.2)

$N=(11011000)_2$. To avoid confusion, we skip some intermediate steps and focus only on the core idea.

Algorithm 1. Embedding Algorithm

Input: Cover image (I_C), Stego Key (K_{STK}), and secret information (M_{SI})

1. Select an appropriate cover image (I_C) from dataset of cover images (DS_{CI})
2. Generate a random permutation matrix (RPM) based on pseudorandom random number generator (PRNG) and permute I_C using PRM, resulting in permuted image (I_P)
3. Divide I_P into three channels, resulting in permuted red, green, and blue channels as: $I_{PR}=I_P(:,:,1)$, $I_{PG}=I_P(:,:,2)$, and $I_{PB}=I_P(:,:,3)$, respectively
4. Select (K_{SK}) and secret pattern for TLEA and encrypt it , providing (K_{ESK})
5. Encrypt M_{SI} by MLEA using K_{ESK}, providing it the block division policy and shuffling pattern, resulting in encrypted secret information (M_{ESI})
6. Calculate $Size\,(M_{ESI})$ and embed it in some pre-defined locations of I_C
7. Embed M_{ESI} using K_{ESK} and I_{PR}, following the scanning order of stego key (K_{STK}) as follows:
8. Set $count \leftarrow 1$ and secret message index $i \leftarrow 1$;
9. **WHILE** $(count \le Size\,(M_{ESI}))$ **DO**
   a. $RLSB \leftarrow getLSB(I_{PR}(count))$;
   b. $Encrypted\,\text{sec}\,ret\,key bit\,(EKB) \leftarrow K_{ESK}(i)$;
   c. **IF** $((RLSB \oplus EKB)=1)$ **THEN**
      i. Divide $I_{PG}(count)$ into two sections as: $I_{PG} \rightarrow LSB^{I_{GP}}(count) \| MSB^{I_{GP}}(count)$;
      ii. Replace $LSB^{I_{GP}}(count) \leftarrow M_{ESI}(count)$;
      iii. Reconstruct pixel i.e. $I_{PGS}(count) \leftarrow LSB^{I_{GP}}(count) \| MSB^{I_{GP}}(count)$;
   ELSE
      i. $I_{PB} \rightarrow LSB^{I_{BP}}(count) \| MSB^{I_{BP}}(count)$;
      ii. Replace $LSB^{I_{BP}}(count) \leftarrow M_{ESI}(count)$;
      iii. Reconstruct pixel i.e. $I_{PBS}(count) \leftarrow LSB^{I_{BP}}(count) \| MSB^{I_{BP}}(count)$;
   END
   d. $count \leftarrow count+1$ and $i \leftarrow i+1$;
   e. **IF** $(i > length\,(K_{ESK}))$
      i. Set $i \leftarrow 1$ i.e. go to start of K_{ESK};
   END
   END
10. Re-permute I_{PR}, I_{PGS}, and I_{PBS}, and finally combine them to get stego image (I_S)

Output: Stego Image (I_S)

[p_1: 11110110, 11010110, 11010110], [p_2: 11010110, 10010110, 10010100],
[p_3: 11010111, 11100110, 11110110], [p_4: 10110110, 10100110, 11010111],
[p_5: 11011110, 10000111, 11010110], [p_6: 11011010, 10110110, 11010111],
[p_7: 11010101, 10010101, 10010110], [p_8: 11011110, 11010110, 11000110].

Start the embedding process from pixel P_1. First, decide the channel in which a secret message bit will be embedded with the help of bitxor operation of red channel LSB and encrypted secret bit of stego key. The LSB of the red channel in pixel P_1 is 0 and the first bit of K_2 is 1. The XOR result $(0 \oplus 1) = 1$, so replace the LSB of green channel of pixel P_1 with the first secret bit of N. For the second pixel P_2, $(0 \oplus 0) = 0$, so replace the LSB of blue channel. For pixel P_3, $(1 \oplus 1) = 0$, so replace the LSB of blue channel and so on. The pixels $[P_1', P_2', P_3', \ldots . P_8']$ are the resultant pixels of the stego image.

[p_1': 11110110, 1101011**1**, 11010110], [p_2': 11010110, 10010111, 1001010**1**],
[p_3': 11010111, 11100110, 1111011**0**], [p_4': 10110110, 1010011**1**, 11010110],
[p_5': 11011110, 10000110, 1101011**1**], [p_6': 11011010, 1011011**0**, 11010110],
[p_7': 11010101, 10010101, 1001011**0**], [p_8': 11011110, 1101011**0**, 11000110].

Herein, the bold face LSBs show the embedding positions in terms of pixels and channels. The bold face underlined LSBs signify that these LSBs are changed during data hiding. From stego image pixels, it is clear that approximately 50 % of the pixels change. Furthermore, the pixel value in the proposed approach is increased or decreased by just 1 and hence do not bring noticeable distortion in the stego image.

3.4 Extraction algorithm

The extraction algorithm is used to extract the hidden secret data from the stego image. To successfully extract data, various parameters, patterns, and secret keys are used including a random permutation matrix, secret key and shuffling pattern of TLEA, blocks division policy and MLEA, and stego key of data embedding method.

These properties augment the security feature of the proposed framework, making data extraction more challenging for attackers. Algorithm 2 illustrates the major steps of the proposed extraction mechanism.

Algorithm 2. Extraction Algorithm

Input: Stego image (I_S), Stego Key (K_{STK}), RPM, and (K_{ESK})

1. Select the received stego image I_S and its appropriate parameters for extraction
2. Permute I_S based on RMP and PRNG to obtain the permuted stego image (I_{PS})
3. Divide I_{PS} into three channels, resulting in permuted red, green, and blue channels as: $I_{PSR}=I_{PS}(:,:,1)$, $I_{PSG}=I_{PS}(:,:,2)$, and $I_{PSB}=I_{PS}(:,:,3)$, respectively
4. Extract the hidden message size from the pre-defined pixels of I_S
5. Select (K_{SK}) and secret pattern for TLEA and encrypt it , providing (K_{ESK})
6. Extract M_{ESI} using K_{ESK} and I_{PSR}, following the scanning order of stego key (K_{STK}) as follows:
7. Set $counter \leftarrow 1$ and secret key index $j \leftarrow 1$;
8. **WHILE** $(Size(\mathrm{M}_{ESI}) \geq counter)$ **DO**
   a. $RLSB \leftarrow getLSB(I_{PSR}(counter))$;
   b. $Encrypted\ \sec\ ret\ key\ bit(EKB) \leftarrow K_{ESK}(j)$;
   c. **IF** $((RLSB \oplus EKB) = 1)$ **THEN**
      i. Divide $I_{PSG}(counter)$ into two sections as:
      $I_{PSG} \rightarrow LSB^{I_{GSP}}(counter) \| MSB^{I_{GSP}}(counter)$;
      ii. $M_{ESI}(counter) \leftarrow LSB^{I_{GSP}}(counter)$;
   ELSE
      i. $I_{PSB} \rightarrow LSB^{I_{BSP}}(counter) \| MSB^{I_{BSP}}(counter)$;
      ii. $M_{ESI}(counter) \leftarrow LSB^{I_{BSP}}(counter)$;
   END
   d. $counter \leftarrow counter+1$ and $j \leftarrow j+1$;
   e. **IF** $(j > length(K_{ESK}))$
      i. Set $j \leftarrow 1$ i.e. go to start of K_{ESK};
   END
   END
9. Decrypt M_{ESI} by MLEA using K_{ESK} and its concerned parameters, producing secret information M_{SI}

Output: Secret information (M_{SI})

4 Experimental results and discussion

This section explains the experimental results of the proposed algorithm and other five algorithms including classic LSB (CLSB), ST-FMM [23], SCC [3], Karim's method [36], and PIT [18], which were coded using MATLAB R2013a with a Core i5 desktop PC, having 8 GB RAM and 3.40GHz processor. The images for the testing purposes were obtained from different open sources in Internet and public dataset USC-SIPI-ID [11], containing standard images of Lena, baboon, f16jet, house, building, and peppers, resulting in a dataset of 50 images. These images are considered as standard images for evaluation of steganography and watermarking algorithms and play an important role in benchmarking [33]. Most of the algorithms developed in this area are usually tested using these standard images due to their suitability of evaluation, because they contain both smooth and edgy images, having statistically rich information. Therefore, we have also considered these images for both quantitative and qualitative evaluation. Extensive experiments were performed from various perspectives, aiming at performance evaluation of the algorithms under consideration, which are illustrated in sub-sequent sections.

4.1 Quantitative evaluation

In this section, quantitative evaluation is performed using various IQAMs based on a set of standard test images. We have evaluated the performance of all methods under consideration using multiple IQAMs based on a dataset of 50 images. The results are collected based on four well-known IQAMs from three perspectives [40], considering varying image dimension and payloads. The evaluation metrics include PSNR, NCC, RMSE, and SSIM, which can be computed using Eqs. 1–5 as follows:

$$PSNR = 10\log_{10}\left(\frac{C_{\max}^2}{MSE}\right) \tag{1}$$

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left(S_{xy}-C_{xy}\right)^2 \tag{2}$$

$$NCC = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left(S_{xy}\times C_{xy}\right)}{\sum_{x=1}^{M}\sum_{y=1}^{N}\left(S_{xy}\right)^2} \tag{3}$$

$$RMSE = \sqrt{RMSE} \tag{4}$$

$$SSIM = \frac{\left(2\mu_x\mu_y + const_1\right)\times\left(2\sigma_{xy} + const_2\right)}{\left(\mu_x^2 + \mu_y^2 + const_1\right)\times\left(\sigma_x^2 + \sigma_y^2 + const_2\right)} \tag{5}$$

where C acts as a host image, S represents the stego image, C_{max} shows the maximum value of pixel in both original and stego image, x and y are subscripted variables, M and N indicate

image resolution in pixels, $const_1$ and $const_2$ avoid division by zero exception, and the rest of the symbols are statistical parameters.

PSNR is a well-known quality measuring metric, calculating the amount of distortion between the input image and marked image in unit of decibel (dB). A higher score of PSNR indicates better image quality, reducing the chances of detection by HVS [27]. Tables 2, 6, and 10 represent PSNR scores of the proposed scheme in comparison with other schemes, showing better quality of the marked images obtained by the proposed scheme, which in turn validates its effectiveness. To visualize the quality of the marked images, a set of popular host and marked images are shown in Figs. 4, 5, and 6, indicating different perspectives of the experiments conducted.

Sajjad et al. [48] and Muhammad et al. [40] argued that multiple IQAMs should be used for quality measurement to fully assess the performance of a given method. Therefore, we have used another metric *RMSE* to estimate the amount of error between the input image and marked image. Chai and Draxler [5] also proved that RMSE is more suitable than MAE in measuring the error distribution. A minimum score of RMSE indicates minimum amount of error, illustrating the efficiency of the method. Tables 4 and 9 show that the RMSE results of the proposed method are smaller than other mentioned schemes in many cases, indicating its superiority.

To further evaluate the effectiveness of various techniques, we have used NCC, which measures the closeness of the input image to its corresponding marked image in the range of 0 to 1. A value of NCC close to 1 represents better quality of the marked image [35]. Tables 3 and 8 show NCC based results where the proposed approach obtains higher scores of NCC than other schemes, highlighting its better performance.

RMSE along with its PSNR produces incorrect results in certain circumstances [49]; therefore, another metric "SSIM" has been used for evaluation, filtering out the performance of each method. The closer the score of SSIM to 1, the better is the performance and vice versa. Tables 5 and 7 show the quantitative results based on SSIM, where the proposed method achieves higher values of SSIM, demonstrating its better results compared to other methods.

Tables 2, 3, 4 and 5 illustrate the quantitative results of various methods including the proposed method from perspective[1], where a message of size 8 KB is embedded in 50 images using the proposed method and other schemes. The mean value of each metric for each corresponding scheme is shown in bold font over fifty images. The average scores of the proposed method in most of the cases are equal or higher than existing methods, indicating its better performance.

Tables 6, 7, 8 and 9 show the quantitative results of the proposed scheme and other mentioned schemes using perspective[2], keeping the message size variable with multiple images of same resolution. The average values using four different IQAMs are shown in bold font in Tables 6, 7, 8 and 9 which successfully dominate the competing methods in terms of PSNR, SSIM, and NCC. However in case of RMSE, the proposed method gives better results compared to PIT and ST-FMM only.

Table 10 shows the quantitative analysis based results using perspective[3]. This type of experiment embeds a message of size 8 KB in different images of variable dimensions. The average PSNR values shown in bold face in Table 10 confirm that the proposed method outperforms the existing methods in case of perspective[3] also, hence verifying its improved performance. In addition, the comparative results of the proposed method based on PSNR with recent high-payload state-of-the-art methods including LSB-M [31], PIT [18], and LSB-MR

[1] Hiding a fixed size message inside multiple images of same resolution, e.g., embedding 8 KB data in 50 images of resolution 256 × 256 pixels

[2] Hiding variable amount of message inside multiple images of same resolution

[3] Hiding fixed size message in same images of varying resolutions

Table 2 Quantitative results using PSNR for comparison between the proposed scheme and other schemes from perspective[1]

Serial#	Image name	CLSB method	SCC [3] method	PIT [18]	ST-FMM [23]	Karim's method [36]	Proposed method
1	F16jet	47.4882	47.4852	45.6879	40.2347	47.4902	53.1665
2	House	51.1659	51.1776	47.6956	40.2518	51.1564	52.7303
3	Trees	39.0436	38.5418	38.2702	39.5397	38.5421	49.7496
4	Scene-2	44.1834	40.0355	39.6545	40.0388	40.0353	46.8066
5	Flowers	33.932	28.5347	28.5169	29.6394	28.5347	42.0526
6	Baboon-2	41.3208	33.932	33.8367	34.4471	33.9322	42.3607
7	Building-1	28.8451	28.8451	28.8213	40.2552	28.8451	43.4071
8	Parrot	55.9115	28.0434	28.0249	27.7969	28.0434	49.2153
9	Baboon	51.1648	48.9531	46.5568	39.9997	48.9536	47.8747
10	Masjid	30.6466	28.5361	28.5173	39.6331	28.5363	44.7425
Avg. of 50 images		**43.1736**	**36.3208**	**34.7621**	**33.9232**	**36.3187**	**45.0309**

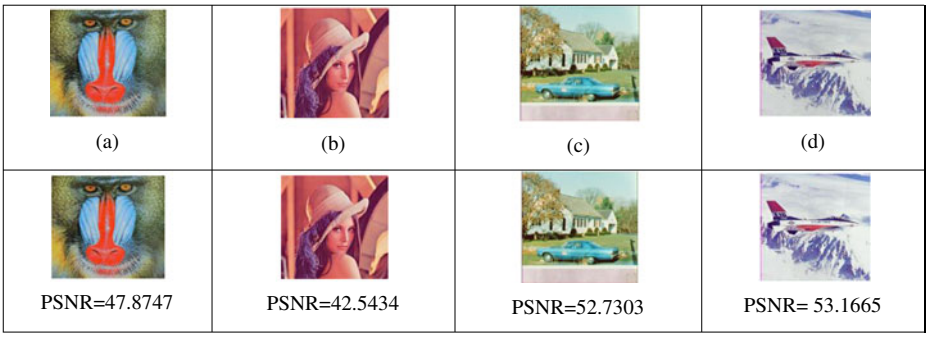

Fig. 4 Visualization of dataset test images for perspective[1]. First row represents host images of **a** baboon, **b** Lena, **c** house, and **d** F16jet, respectively. The second row visualizes marked images with their corresponding PSNR score, containing secret information

[32] are shown in Fig. 7. It is clear from Fig. 7 that the performance of SCC, PIT, and Karim's method are relatively same. LSB-M obtains better results than SCC, PIT, ST-FMM, and Karim's method. However, LSB-MR dominates other algorithms except CLSB and the proposed scheme. ST-FMM provides worse results according to this experiment. The proposed scheme also produces encouraging results in this experiment, validating its high image quality, which in turn reduces the chances of HVS detection.

4.2 Qualitative evaluation

The performance of a steganographic algorithm can also be measured by qualitative evaluation using HVS, considering visual quality of marked images and histogram changeability. For this purpose, the histograms of stego images, produced by CLSB, SCC, PIT, ST-FMM, Karim's technique, and the proposed technique are calculated as shown in Fig. 8. The marked images are generated hiding a message of size 8 KB in the famous Lena image using the proposed scheme as well as other schemes under consideration. From HVS based results in Fig. 8, it can be noted that the stego image and its histogram produced by the proposed approach are almost same as the original cover image and its histogram, thus validating the superiority of the proposed method.

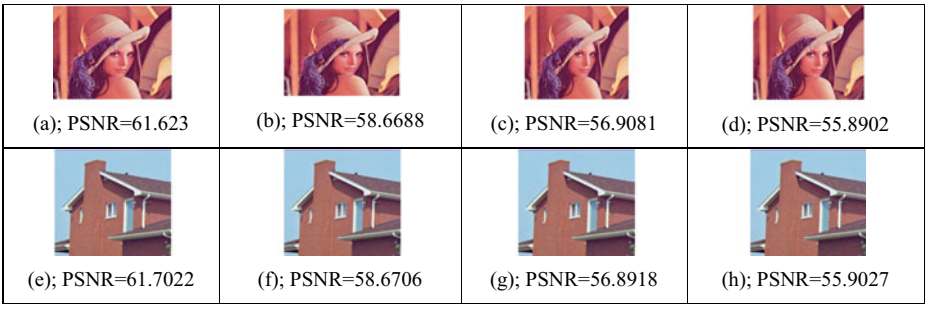

Fig. 5 Visualization of marked images for perspective[2]. In first row, Lena image with its four versions is shown, containing (2, 4, 6, and 8) KB payload, respectively. In row 2, the stego image "building" is depicted with various sizes of payload

Fig. 6 Visualization of marked images for perspective[3]. **a**, **b**, **c**, **d**: standard Lena image with resolutions (128 × 128, 256 × 256, 512 × 512 and 1024 × 1024 pixels); **e**, **f**, **g**, **h**: Building image with different dimensions

4.3 Stego-key sensitivity analysis

To increase the security of a steganographic algorithm, the length of stego key should be large enough to make a brute force attack infeasible because the larger the stego key length is, the more time is required by an attacker breaking the algorithm. In a brute force attack, the attacker tries all possible combination of letters in breaking the algorithm [15, 37]. In the proposed method, if the attacker successfully unravels the secret key, he or she is still unable to extract the original secret data because secret key is encrypted by TLEA, requiring additional secret parameters.

The length of stego key in the proposed method is set to eight digits for simplicity avoiding large computational cost. The range of digits for stego key is 1–8 with no repetition. One can further increase the security by increasing the length of the stego key. For instance, the current stego key is Key = 32741586, which can be extended to Key = "32741586 73461258 84275163", increasing further its security. It should be noted that there is no repetition in each 8 digit block of extended key. The same way other supporting keys can be extended for better security. In this case, the first byte can be embedded using the first eight digits of the key, the second byte can be hidden using the next eight digits of the key, and so on up to end of secret information.

Table 3 Quantitative evaluation using NCC for comparison between the proposed technique and other techniques from perspective[1]

Serial#	Image name	CLSB method	SCC [3] method	PIT [18]	ST-FMM [23]	Karim's method [36]	Proposed method
1	F16jet	0.9997	0.9997	0.9996	0.9993	0.9997	0.9997
2	Building-1	0.9795	0.9795	0.9795	0.9993	0.9796	0.9796
3	Baboon	0.9998	0.9998	0.9997	0.999	0.9998	0.9998
4	House	0.9999	0.9999	0.9998	0.9994	0.9999	0.9999
5	Trees	0.999	0.999	0.9989	0.9997	0.999	0.999
6	Moon	0.9998	0.9998	0.9997	0.999	0.9998	0.9998
7	Lena	0.9999	1	0.9999	0.9994	1	1
8	Parrot	0.9999	0.9991	0.999	0.9985	0.9991	0.9991
9	Laser-Light	0.9967	0.9938	0.9937	0.9992	0.9938	0.9938
10	Kite	0.9762	0.9582	0.9582	0.9974	0.9582	0.9582
Avg. of 50 images		**0.96682**	**0.952922**	**0.9529**	**0.998448**	**0.952924**	**0.952924**

Table 4 Quantitative RMSE based results for comparison between the proposed scheme and other competitive methods from perspective[1]

Serial#	Image name	CLSB method	SCC [3] method	PIT [18]	ST-FMM [23]	Karim's method [36]	Proposed method
1	Lena	0.2785	0.2773	0.6019	1.7284	0.2785	0.2855
2	F16jet	0.2787	0.2801	0.5981	1.7358	0.2789	0.2856
3	Trees	0.2783	0.2784	0.5966	1.7276	0.2782	0.2853
4	Peppers	0.2714	0.273	0.5879	1.6937	0.272	0.2803
5	House	0.2781	0.2757	0.5957	1.7255	0.2771	0.2833
6	Baboon-3	0.2769	0.2775	0.5977	1.7242	0.2772	0.2838
7	Moon	0.2768	0.2775	0.5936	1.7383	0.2768	0.2845
8	Temple	0.2792	0.2796	0.5999	1.7311	0.2773	0.2839
9	Building1	0.2757	0.2764	0.5921	1.7233	0.2771	0.2838
10	Baboon	0.276	0.2772	0.5971	1.737	0.2764	0.284
Avg. of 50 images		**0.27129**	**0.27469**	**0.586944**	**1.695824**	**0.27405**	**0.280874**

4.4 Execution time based comparison

Execution time is an important factor for measuring the efficiency of any steganographic scheme. An algorithm is considered to be the best one if it takes small segment of time during computation. Due to this, the proposed algorithm was analysed for its execution time, comparing with other state-of-the-art schemes. Table 11 indicates the time required by each algorithm for data embedding and PSNR calculation.

The results were collected executing each mentioned algorithm fifteen times while hiding a text file of size 8 KB in fifty images. The scores with bold font show the average running time of each algorithm over 15 iterations. The results indicate that the proposed algorithm requires less time for data embedding than SCC technique and high-payload scheme, e.g., PIT, but takes slightly more time than Karim's method. On the other hand, CLSB and ST-FMM are relatively fast algorithms compared to the proposed scheme, PIT, and SCC, but fail to achieve

Table 5 Quantitative SSIM based results for performance evaluation of proposed method and other methods from perspective[1]

Serial#	Image name	CLSB method	SCC [3] method	PIT [18]	ST-FMM [23]	Karim's method [36]	Proposed method
1	Peppers	0.8843	0.8774	0.8756	0.9488	0.8773	0.8773
2	F16jet	0.9976	0.9985	0.9964	0.9797	0.9985	0.9985
3	Building-1	0.9963	0.9973	0.9948	0.9765	0.9972	0.9973
4	Baboon	0.9989	0.9993	0.9985	0.9925	0.9992	0.9992
5	House	0.9983	0.999	0.9974	0.986	0.9989	0.9989
6	Trees	0.9964	0.997	0.9956	0.9858	0.997	0.997
7	Moon	0.9976	0.9986	0.9965	0.9786	0.9985	0.9985
8	Lena	0.9981	0.9989	0.9971	0.9822	0.9989	0.9988
9	Masjid	0.9843	0.9828	0.981	0.9881	0.9828	0.9828
10	Baboon-2	0.9953	0.9938	0.9928	0.9888	0.9937	0.9937
Avg. of 50 images		**0.96897**	**0.95604**	**0.954382**	**0.975158**	**0.955982**	**0.955984**

Table 6 Quantitative results using PSNR for comparison between the proposed scheme and other schemes from perspective[2]

Image name	Secret data (KBs)	Cipher size in bytes	CLSB method	SCC method [3]	PIT [18]	ST-FMM [23]	Karim's method [36]	Proposed method
Lena with resolution 256 × 256	2	2406	45.8307	45.8314	49.2562	40.3354	45.8317	61.623
	4	4177	45.7183	45.7193	49.2242	40.3033	45.7193	58.6688
	6	6499	45.6108	45.6128	49.2061	40.2696	45.61	56.9081
	8	8192	45.53	45.5296	49.2044	40.249	45.5267	55.8902
	Average		**45.6725**	**45.6733**	**49.2227**	**40.28933**	**45.6719**	**58.2725**
Building image with dimension 256 × 256	2	2406	28.8513	28.8513	28.8378	40.3785	28.8514	61.7022
	4	4177	28.8491	28.849	28.8315	40.3356	28.8491	58.6706
	6	6499	28.8468	28.8468	28.8253	40.3044	28.8468	56.8918
	8	8192	28.8451	28.8451	28.8213	40.2552	28.8451	55.9027
	Average		**28.8481**	**28.8481**	**28.829**	**40.31843**	**28.8481**	**58.2918**

acceptable image quality and security, limiting their usability. Overall, the proposed framework is a better combination of running time, security, and visual quality.

4.5 Evaluation of steganography strength

One of the most important parameter of any steganographic algorithm is to find the number of iterations required for its breaking [6]. An algorithm is considered to be more secure if it requires large number of iterations applying any brute force approach [12]. Keeping in view this concern, the number of iterations required to break the proposed algorithm are calculated as shown in Table 12.

To clarify the procedure of calculating the number of iterations, consider the key length k = 2 and key = 23. Now to apply the brute force approach, we need 100 iterations for key breaking, i.e., 00, 01, 02.....99. The next operation is to find out the shuffling pattern by applying different combinations of the given two digits of the stego key, i.e., 23 and 32. Thus, the total number of iterations required are $10^k \times k! = 100 \times 2! = 200$. If we increase the key length such as k = 3 and key = 236, then we need to iterate 000–999, i.e., 1000 iterations.

Table 7 SSIM based quantitative results for comparison between the proposed scheme and other schemes from perspective[2]

Image name	Secret data (KBs)	Cipher size in bytes	CLSB method	SCC method [3]	PIT [18]	ST-FMM[23]	Karim's method [36]	Proposed method
Lena with resolution 256 × 256	2	2406	0.9991	0.9993	0.9971	0.9819	0.9993	0.9997
	4	4177	0.9987	0.9991	0.997	0.9818	0.9991	0.9995
	6	6499	0.9981	0.9988	0.9968	0.9818	0.9987	0.9991
	8	8192	0.9977	0.9985	0.9983	0.9818	0.9984	0.9988
	Average		**0.9984**	**0.99893**	**0.9973**	**0.98183**	**0.99888**	**0.99928**
Building image with dimension 256 × 256	2	2406	0.998	0.9983	0.9964	0.9765	0.9995	0.9995
	4	4177	0.9974	0.998	0.9952	0.9765	0.9991	0.9991
	6	6499	0.9968	0.9976	0.995	0.9766	0.9987	0.9986
	8	8192	0.9963	0.9973	0.9948	0.9765	0.9983	0.9983
	Average		**0.99713**	**0.9978**	**0.99535**	**0.97653**	**0.9989**	**0.99888**

Table 8 NCC based quantitative results for comparison between the proposed scheme and other schemes from perspective[2]

Image name	Secret data (KBs)	Cipher size in bytes	CLSB method	SCC method [3]	PIT [18]	ST-FMM [23]	Karim's method [36]	proposed method
Lena with resolution 256 × 256	2	2406	0.9996	0.9996	0.9999	0.9994	0.9996	1
	4	4177	0.9996	0.9996	0.9999	0.9994	0.9996	1
	6	6499	0.9996	0.9996	0.9999	0.9994	0.9996	1
	8	8192	0.9996	0.9996	0.9999	0.9994	0.9996	1
	Average		**0.9996**	**0.9996**	**0.9999**	**0.9994**	**0.9996**	**1**
Building image with dimension 256 × 256	2	2406	0.9796	0.9796	0.9795	0.9993	0.9796	1
	4	4177	0.9796	0.9796	0.9795	0.9993	0.9796	1
	6	6499	0.9796	0.9796	0.9795	0.9993	0.9796	1
	8	8192	0.9795	0.9795	0.9795	0.9993	0.9796	1
	Average		**0.97958**	**0.97958**	**0.9795**	**0.9993**	**0.9796**	1

To find out the key shuffling pattern, we iterate the combinations 236, 263, 326, 362, 623, and 632, i.e., 6 combinations. Therefore, the total number of iterations = $10^k \times k! = 1000 \times 3! = 1000 \times 6 = 6000$. Continuing the same procedure, a number of keys with different lengths are taken and the number of iterations required for its breakage are calculated as shown in Table 12, where K shows the number of digits in the stego key and N represents the image dimensions (that is, 128, 256, 512, and 1024 pixels). The statistics of Table 12 indicate that enlarging the key length in the proposed framework increases the number of iterations for its cracking which in turn improves its security.

To further analyze the security strength of the proposed framework, we use Kirchhoff's principle [2] and compare the strength of our algorithm with Para et al. [44] and El Hennawy et al. [12] schemes. The Kirchhoff's principle assumes that the data hiding algorithm is known to the public. In this case, the adversaries need the information about secret keys, making their selection more challenging. Therefore, it is desired to use complex secret keys, having enough larger length to resist against brute-force attacks.

Our framework uses four main sub-keys with minimum length of 64 bits each, resulting in a master key of 216 bits. The keys include secret key of 64 bits and secret pattern of 64 bits for TLEA, shuffling pattern of 64 bits for MLEA, and stego

Table 9 Perspective[2] based comparison of proposed approach with other methods using RMSE

Image name	Secret data (KBs)	Cipher size in bytes	CLSB method	SCC method [3]	PIT [18]	ST-FMM [23]	Karim's method [36]	Proposed method
Lena with resolution 256 × 256	2	2406	0.1437	0.1429	0.6019	1.734	0.143	0.1473
	4	4177	0.2039	0.2036	0.6082	1.7383	0.2031	0.2063
	6	6499	0.249	0.2484	0.619	1.7432	0.2491	0.2523
	8	8192	0.2785	0.2787	0.6199	1.7457	0.2794	0.2855
	Average		**0.21878**	**0.2184**	**0.61225**	**1.7403**	**0.21865**	**0.22285**
Building image with dimension 256 × 256	2	2406	0.1416	0.1433	0.5863	1.7176	0.142	0.1447
	4	4177	0.2012	0.2023	0.5911	1.718	0.202	0.2058
	6	6499	0.2454	0.2468	0.5914	1.7156	0.2477	0.2526
	8	8192	0.2757	0.2764	0.5921	1.717	0.2775	0.2847
	Average		**0.21598**	**0.2172**	**0.59023**	**1.71705**	**0.2173**	**0.22195**

Table 10 Quantitative results using PSNR for comparison between the proposed scheme and other schemes from perspective[3]

Image name	Image dimensions	CLSB method	SCC method [3]	PIT [18]	ST-FMM [23]	Karim's method [36]	Proposed method
Lena image	128 × 128	42.1208	42.1201	41.368	40.3257	42.121	42.1212
	256 × 256	45.531	45.5286	45.9463	40.2378	45.5343	47.4919
	512 × 512	47.0517	47.0523	47.1957	40.3152	47.0515	48.7445
	1024 × 1024	48.9022	48.9023	48.9566	40.3378	48.902	49.8573
	Average	**45.90143**	**45.90083**	**45.86665**	**40.30413**	**45.9022**	**47.053725**
Building image	128 × 128	64.8137	64.656	49.1793	40.4385	64.72	64.716
	256 × 256	46.3978	46.3994	46.9153	40.2848	46.3958	47.49
	512 × 512	48.7443	48.7432	48.9566	40.4097	48.7425	47.9844
	1024 × 1024	49.0109	49.0109	49.0666	40.4239	49.0106	48.9023
	Average	**52.24168**	**52.20238**	**48.52945**	**40.38923**	**52.21723**	**52.273175**

key of 64 bits for SKA-LSB scheme. The detailed security analyses is described as follows:

Master key length = 216 bits

Key space = $2^{216} = 1.0531 \times 10^{65}$ keys

If a malicious user produces 1 million keys per second, then it will take the following amount of time.

$$\text{Amount of time required for breakage} = \frac{2^{216}}{10^6 \times 365 \times 86400} = 3.3394 \times 10^{51} \text{ (Years)}$$

Average= 1.6697×10^{51} (Years)

The analysis is repeated for Parah et al. [44] method and El Hennawy et al. [12] scheme. The results shown in Table 13 indicate that the proposed framework offers much better security against brute-force attack in a cost-effective manner.

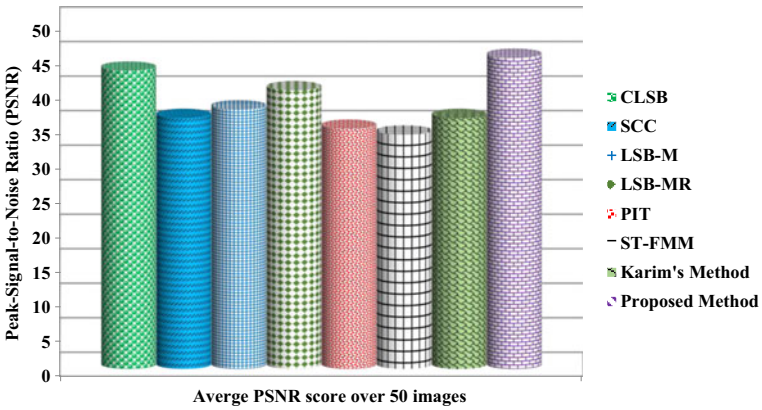

Fig. 7 PSNR based comparison of the proposed scheme with high-payload state-of-the-art methods

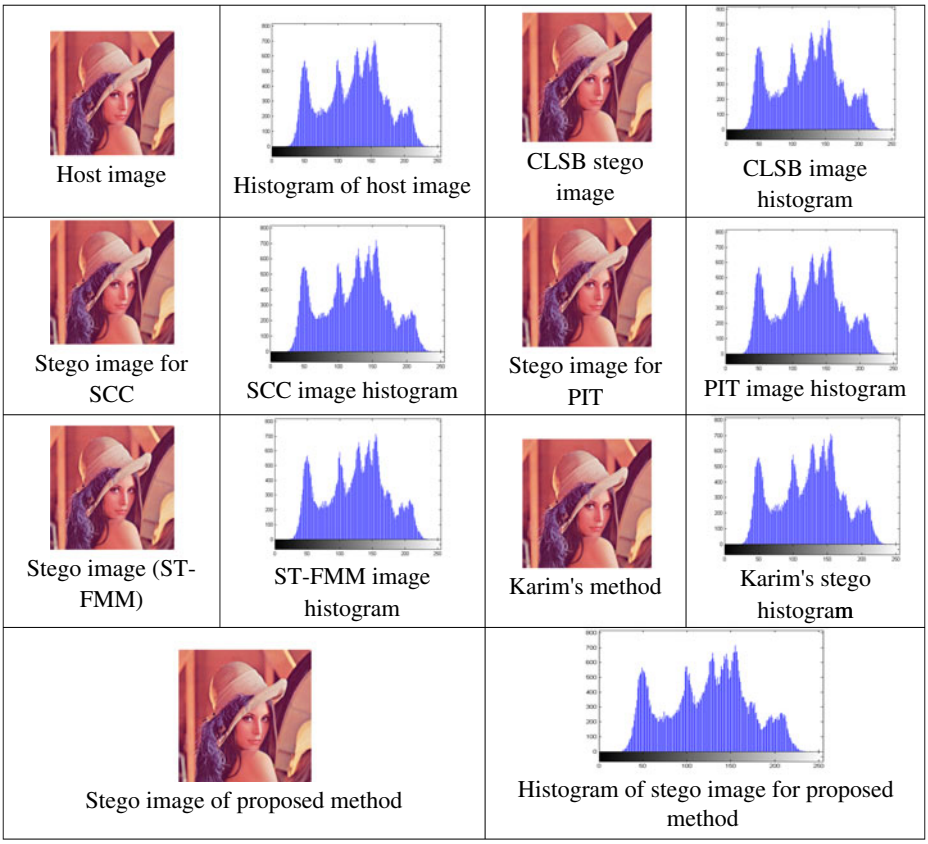

Fig. 8 Visual quality assessment of marked images, produced by CLSB, SCC, PIT, ST-FMM, Karim's method, and the proposed scheme

Table 11 Execution Time (sec) analysis based comparison of the proposed method with other methods

Iteration#	Classic LSB method	SCC [3] method	PIT [18]	ST-FMM [23]	Karim's method [36]	Proposed method
1	15.01356	26.10996	54.5771	13.4949	24.06434	27.040601
2	14.67699	25.72126	53.20009	13.00397	18.80812	26.168807
3	14.72414	25.14657	59.37227	13.74399	20.94527	25.900238
4	14.78429	22.59389	56.3436	13.68522	16.79531	19.13727
5	14.90971	22.01468	52.99203	10.58982	17.59227	22.221128
6	14.8083	24.42272	101.4388	13.13324	20.28701	18.004642
7	14.65838	25.71366	64.84528	11.52611	21.52358	22.398091
8	17.83017	25.74213	77.006943	12.62904	23.1757	24.800869
9	14.68486	27.89002	105.1882	18.38744	21.73853	22.761408
10	14.748854	25.57424	93.43394	14.6174	22.41336	18.215256
Average of 15 iterations	**14.9386**	**25.1488**	**65.6662**	**13.3804**	**21.2342**	**22.8586063**

Table 12 Evaluation of the strength of the proposed algorithm in terms of number of iterations

Serial#	Key length (K) [Number of digits]	Number of operations for stego key ($10^k \times k!$)	Number of operations for whole algorithm ($10^k \times k!) \times N^2$
1	8	4.032E + 12	$4.032E + 12 \times N^2$
2	16	2.09228E + 29	$2.09228E + 29 \times N^2$
3	32	2.63131E + 67	$2.63131E + 67 \times N^2$
4	64	1.2689E + 153	$1.2689E + 153 \times N^2$
5	72	6.1234E + 175	$6.1234E + 175 \times N^2$
6	80	7.1569E + 198	$7.1569E + 198 \times N^2$
7	88	1.8548E + 222	$1.8548E + 222 \times N^2$
8	96	9.9168E + 245	$9.9168E + 245 \times N^2$
9	104	1.0299E + 270	$1.0299E + 270 \times N^2$
10	112	1.9745E + 294	$1.9745E + 294 \times N^2$

4.6 Summary of overall performance evaluation

In this section, we summarize the performance of all steganographic schemes under consideration based on the aforementioned results. It is highly agreed by researchers of this area that there are three main metrics based on which the performance of new steganographic techniques can be evaluated using a magic triangle, represented by Chen at al. [9]. These metrics include imperceptibility, capacity, and robustness. Imperceptibility refers to the image quality of marked images measured based on various IQAMs. Capacity indicates the amount of secret data embedded in cover images known as payload. Robustness shows the resiliency of a given technique against image processing attacks. In addition, there are two more metrics which are also considered for evaluation known as security and computational complexity [38, 52]. The target of any steganographic algorithm is to achieve high payload with better security and imperceptibility, keeping itself computationally in-expensive with resiliency against attacks. However, there is a trade-off among these metrics, making the achievement of this target more challenging.

The payload of an algorithm is measured using bits per pixel (bpp) which is 1 bpp for our algorithm and other competing techniques excluding PIT and ST-FMM. The capacity of PIT is higher among the mentioned schemes, but it is computationally very expensive as validated from Table 11, restricting its usage in real-time applications. The capacity of ST-FFM is not guaranteed to be 1bpp in all cases due to its

Table 13 Comparison of the proposed method with other schemes in terms of security

Technique name	Key length (bits)	Key space	Amount of time for breaking (years)
Parah et al. [44]	57	2^{57}	2.2849×10^3
El Hennawy et al. [12]	128	2^{128}	5.3951×10^{24}
Proposed Method	216	2^{216}	1.6697×10^{51}

Table 14 PSNR based robustness evaluation of the proposed method with other competing methods using salt & pepper noise having density 0.05

Serial#	Image name	CLSB method	SCC [3] method	PIT [18]	ST-FMM [23]	Karim's method [36]	Proposed method
1	Peppers	34.7114	30.0124	32.4389	26.7176	27.1899	34.7691
2	Lena	34.5886	30.1284	28.4031	26.9048	26.918	35.6021
3	House	34.7114	29.4317	29.1023	27.0404	26.9328	34.0198
4	Baboon	34.6121	26.4387	25.6398	26.7673	26.9407	34.5581
5	Airplane	34.6595	25.3821	28.5631	26.914	26.9504	35.9032
Average over 50 images		**35.0131**	**28.6666**	**29.1458**	**27.1272**	**27.1321**	**35.2071**

dependency on window size, hence limiting its suitability for applications requiring a minimum of 1 bpp. The imperceptibility has been evaluated by various IQAMs for all techniques under consideration, and it can be confirmed from Tables 2, 3, 4, 5, 6, 7, 8, 9 and 10 that the proposed framework achieves higher scores in most cases, validating its improved performance.

The robustness property is highly demanded in watermarking applications, which can be achieved by exploring transform domain techniques such as DWT[4] and DCT,[5] compromising on high computational complexity and lower payload [28]. The spatial domain techniques are relatively less robust, indicating their limitation [6], we performed an experiment for robustness evaluation as conducted by authors of [33]. In this experiment, we hide a secret image of size 64×64 pixels inside 50 cover images of the dataset, setting their size to 512×512 pixels. The secret image is then extracted from the stego images, which have been attacked by salt & pepper noise having density 0.05. The quality of retrieved images is then measured using PSNR, whose results are shown in Table 14, indicating comparatively better resiliency of the proposed method against noise attack and thus validating its robustness.

The fourth metric *security* refers to the level of barriers in the way of attackers and difficulty in extraction of secret data. CLSB method is straight forward with no security consideration; hence, it is easy for adversaries to extract data. The SCC method disperses data in three channels but in fixed cyclic order, enabling attackers for easy extraction if some initial pixels get cracked. Karim's method provides better security compared to CLSB, SCC, and ST-FMM due to concept of indicators but direct embedding of sensitive information without encryption limits its applications. The security of the proposed method can be confirmed using Tables 12 and 13 of Section 4.5, providing enough security against brute-force attack. The computational complexity of the proposed scheme is also lower than SCC and PIT approaches as shown in Table 11. With these achievements, it can be concluded that the proposed framework successfully maintains a better trade-off among image quality, payload, security, and computational complexity, extending its suitability for secure communication over the Internet.

[4] Discrete Wavelength Transform
[5] Discrete Cosine Transform

5 Conclusion and future work

In this paper, we proposed a secure image steganographic framework for secure transmission of secret information over the public network. A steganographic technique that focuses only on payload or image quality is not sufficient to be used in current security applications. Our framework used SKA-LSB substitution method and multi-level cryptography, producing a security system that maintains a better trade-off between image quality, payload, security, and computational complexity. We explored TLEA and MLEA for encryption of secret key and secret information, respectively, and used SKA-LSB substitution method for its embedding, making data extraction more challenging for adversaries. We evaluated our framework quantitatively and qualitatively based on various IQAMs, producing better image quality with a reasonable payload. Our framework is also computationally in-expensive and provides higher security compared to other state-of-the-art techniques. Due to these characteristics, our framework is relatively more suitable for secure transmission of EPR to healthcare centers, top-secret sensitive communication between intelligence departments, and secure private communication.

In future, the authors tend to increase the payload by analyzing the correlation between pixels and use saliency detection models to hide data in relatively less salient regions, avoiding attackers' attention. Sparse coding is also a future consideration for integration with the proposed framework to make it more resilience against image processing attacks.

Acknowledgments We are sincerely thankful to the prolific suggestions and constructive comments of the associate editor and anonymous reviewers which improved the quality of this work.

References

1. Al-Taani AT, Al-Issa AM (2009) A novel steganographic method for gray-level images. Int J Comput, Inform Syst Sci, Eng 3: 1 2009
2. Aziz M, Tayarani-N MH, Afsar M (2015) A cycling chaos-based cryptic-free algorithm for image steganography. Nonlinear Dyn 80:1271–1290
3. Bailey K, Curran K (2006) An evaluation of image based steganography methods. Multimed Tools Appl 30:55–88
4. Bin L, Ming W, Xiaolong L, Shunquan T, Jiwu H (2015) A strategy of clustering modification directions in spatial image steganography. IEEE Trans Inf Forensics Secur 10:1905–1917
5. Chai T, Draxler RR (2014) Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. Geosci Model Dev 7:1247–1250
6. Cheddad A, Condell J, Curran K, Mc Kevitt P (2010) Digital image steganography: survey and analysis of current methods. Signal Process 90:727–752
7. Cheddad A, Condell J, Curran K, Mckevitt P (2014) Encryption method. ed: Google Patents
8. Chen W-Y (2008) Color image steganography scheme using DFT, SPIHT codec, and modified differential phase-shift keying techniques. Appl Math Comput 196:40–54
9. Chen W-J, Chang C-C, Le THN (2010) High payload steganography mechanism using hybrid edge detector. Expert Syst Appl 37:3292–3301
10. Chen P-Y, Lin H-J (2006) A DWT based approach for image steganography. Int J Appl Sci Eng 4:275–290
11. Cheng W-C, Pedram M (2004) Chromatic encoding: a low power encoding technique for digital visual interface. IEEE Trans Consum Electron 50:320–328
12. El Hennawy HM, Omar AE, Kholaif SM (2015) LEA: link encryption algorithm proposed stream cipher algorithm. Ain Shams Eng J 6:57–65
13. El-Emam NN (2015) New data-hiding algorithm based on adaptive neural networks with modified particle swarm optimization. Comput Secur 55:21–45
14. Emam NNE, Diabat MA (2015) A novel algorithm for colour image steganography using a new intelligent technique based on three phases. Appl Soft Comput

15. Eng PMKM, Zaynab Najeeb Abdulhameed (2014) High capacity steganography based on chaos and contourlet transform for hiding multimedia data. Int J Electron Comm Eng Technol (IJECET) 5:26–42
16. Fakhredanesh M, Rahmati M, Safabakhsh R (2013) Adaptive image steganography using contourlet transform. J Electron Imaging 22:043007
17. Grover N, Mohapatra A (2013) Digital image authentication model based on edge adaptive steganography. In: Advanced Computing, Networking and Security (ADCONS), 2013 2nd International Conference on, p 238–242
18. Gutub AA-A (2010) Pixel indicator technique for RGB image steganography. Journal of Emerging Technologies in Web Intelligence 2:56–64
19. Hong W (2013) Adaptive image data hiding in edges using patched reference table and pair-wise embedding technique. Inform Sci 221:473–489
20. Hong W, Chen T-S (2012) A novel data embedding method using adaptive pixel pair matching. IEEE Trans Inf Forensics Secur 7:176–184
21. Ioannidou A, Halkidis ST, Stephanides G (2012) A novel technique for image steganography based on a high payload method and edge detection. Expert Syst Appl 39:11517–11524
22. Ishtiaq M, Jaffar MA, Khan MA, Jan Z, Mirza AM (2009) Robust and imperceptible watermarking of video streams for low power devices. In: Signal Processing, Image Processing and Pattern Recognition, ed: Springer, p 177–184
23. Jassim FA (2013) A novel steganography algorithm for hiding text in image using five modulus method. Int J ComputAppl 72
24. Jiang N, Zhao N, Wang L (2015) LSB based quantum image steganography algorithm. Int J Theor Phys, p 1–17
25. Jinomeiq L, Baoduui W, Xinmei W (2007) One AES S-box to increase complexity and its cryptanalysis. J Syst Eng Electron 18:427–433
26. Kanan HR, Nazeri B (2014) A novel image steganography scheme with high embedding capacity and tunable visual image quality based on a genetic algorithm. Expert Syst Appl 41:6123–6130
27. Khan Muhammad IM, Sajjad M, Ahmad J, Yoo SJ, Han D, Baik SW (2015) Secure visual content labelling for personalized image retrieval. In: The 11th International Conference on Multimedia Information Technology and Applications (MITA 2015) June 30-July2, 2015, Tashkent, Uzbekistan, p 165–166
28. Khan A, Siddiqa A, Munib S, Malik SA (2014) A recent survey of reversible watermarking techniques. Inform Sci 279:251–272
29. Lee Y-P, Lee J-C, Chen W-K, Chang K-C, Su I-J, Chang C-P (2012) High-payload image hiding with quality recovery using tri-way pixel-value differencing. Inform Sci 191:214–225
30. Liu Q, Li P-y, Zhang M-c, Sui Y-x, Yang H-j (2015) A novel image encryption algorithm based on chaos maps with Markov properties. Commun Nonlinear Sci Numer Simul 20:506–515
31. Luo W, Huang F, Huang J (2010) Edge adaptive image steganography based on LSB matching revisited. IEEE Trans Inf Forensics Secur 5:201–214
32. Mielikainen J (2006) LSB matching revisited. IEEE Signal Process Lett 13:285–287
33. Muhammad K, Ahmad J, Farman H, Jan Z, Sajjad M, Baik SW (2015) A secure method for color image steganography using gray-level modification and multi-level encryption. KSII T Internet Inf 9:1938–1962
34. Muhammad K, Ahmad J, Rehman NU, Jan Z, Qureshi RJ (2015) A secure cyclic steganographic technique for color images using randomization. Technical Journal UET Taxila, Pakistan 19(3):57–64
35. Muhammad K, Ahmad J, Sajjad M, Zubair M (2015) Secure image steganography using cryptography and image transposition. NED Univ J Res 12(4):81
36. Muhammad K, Jan Z, Ahmad J, Khan Z (2015) An adaptive secret key-directed cryptographic scheme for secure transmission in wireless sensor networks. Technical Journal, University of Engineering and Technology (UET) Taxila, Pakistan 20(3):48–53
37. Muhammad K, Mehmood I, Lee MY, Ji SM, Baik SW (2015) Ontology-based secure retrieval of semantically significant visual contents. J Kor Inst Next Gener Comput 11(3):87–96
38. Muhammad K, Mehmood I, Sajjad M, Ahmad J, Yoo SJ, Han D, Wook S (2015) Secure visual content labelling for personalized image retrieval. In: The 11th international conference on multimedia information technology and applications (MITA 2015), 2015, Tashkent, Uzbekistan, pp 165–166
39. Muhammad K, Sajjad M, Baik SW (2016) Dual-level security based cyclic18 steganographic method and its application for secure transmission of keyframes during wireless capsule endoscopy. J Med Syst 40:1–16
40. Muhammad K, Sajjad M, Mehmood I, Rho S, Baik SW (2015) A novel magic LSB substitution method (M-LSB-SM) using multi-level encryption and achromatic component of an image. Multimed Tools Appl:1–27
41. NguyenTD, Arch-int S, Arch-int N (2015) An adaptive multi bit-plane image steganography using block data-hiding. Multimed Tools Appl:1–27,
42. Nie T, Zhang T (2009) A study of DES and Blowfish encryption algorithm. In: TENCON 2009–2009 I.E. Region 10 Conference, p 1–4
43. Noda H, Niimi M, Kawaguchi E (2006) High-performance JPEG steganography using quantization index modulation in DCT domain. Pattern Recogn Lett 27:455–461

44. Parah SA, Sheikh JA, Hafiz AM, Bhat G (2014) Data hiding in scrambled images: a new double layer security data hiding technique. Comput Electr Eng 40:70–82
45. Parvez MT, Gutub AA-A (2011) Vibrant color image steganography using channel differences and secret data distribution. Kuwait J Sci Eng 38:127–142
46. Qazanfari K, Safabakhsh R (2014) A new steganography method which preserves histogram: generalization of LSB^{++}. Inform Sci 277:90–101
47. Roy R, Sarkar A, Changder S (2013) Chaos based edge adaptive image steganography. Procedia Technol 10: 138–146
48. Sajjad M, Mehmood I, Abbas N, Baik SW (2016) Basis pursuit denoising-based image superresolution using a redundant set of atoms. SIViP 10(1):181–188
49. Sajjad M, Mehmood I, Baik SW (2014) Sparse representations-based super-resolution of key-frames extracted from frames-sequences generated by a visual sensor network. Sensors 14:3652–3674
50. Tang M, Zeng S, Chen X, Hu J, Du Y (2015) An adaptive image steganography using AMBTC compression and Interpolation Technique. Op-Int J Light Electron Opt
51. Wang C-M, Wu N-I, Tsai C-S, Hwang M-S (2008) A high quality steganographic method with pixel-value differencing and modulus function. J Syst Softw 81:150–158
52. Wu S, Liu Y, Zhong S, Liu Y (2015) What makes the stego image undetectable?. In Proceedings of the 7th International Conference on Internet Multimedia Computing and Service, p 47
53. Wu D-C, Tsai W-H (2003) A steganographic method for images by pixel-value differencing. Pattern Recogn Lett 24:1613–1626
54. Zhang W, Zhang X, Wang S (2007) A double layered "plus-minus one" data embedding scheme. IEEE Signal Process Lett 14:848–851

Khan Muhammad received his BCS degree in Computer Science from Islamia College, Peshawar, Pakistan in 2014 with research in image processing. Currently, he is pursuing Joint Master-PhD degree in digital contents from Sejong University, Seoul, South Korea. His research interests include image and video processing, data hiding, image and video steganography, video summarization, diagnostic hysteroscopy, and wireless capsule endoscopy.

Jamil Ahmad received his BCS and MS degree in Computer Science from the University of Peshawar, and Islamia College, Peshawar, Pakistan respectively. Currently, he is pursuing PhD degree in digital contents from Sejong University, Seoul, Korea. His research interests include image analysis, semantic image representation and content based multimedia retrieval.

Naeem Ur Rehman received his BS degree in Computer Science from Islamia College, Peshawar, Pakistan in 2014. His research interests include image processing, data hiding, and image steganography.

Zahoor Jan is currently holding the rank of an associate professor in computer science at Islamia College Peshawar, Pakistan. He received his MS and PHD degree from FAST University Islamabad in 2007 and 2011 respectively. He is also the chairman of Department of Computer Science at Islamia College Peshawar, Pakistan. His areas of interests include image processing, machine learning, computer vision, artificial intelligence and medical image processing, biologically inspired ideas like genetic algorithms and artificial neural networks, and their soft-computing applications, biometrics, solving image/video restoration problems using combination of classifiers using genetic programming, optimization of shaping functions in digital watermarking and image fusion.

Muhammad Sajjad received his PhD degree in Digital Contents from Sejong University, Seoul, South Korea. He is now working as a research associate at Islamia College Peshawar, Pakistan. His research interests include digital image super-resolution and reconstruction, sparse coding, video summarization and prioritization, image/video quality assessment, and image/video retrieval. He is the corresponding author of this paper.