Iran Journal of Computer Science (2018) 1:175–186
https://doi.org/10.1007/s42044-018-0016-3

ORIGINAL ARTICLE

An image distortion-based enhanced embedding scheme

A. H. M. Kamal[1,2] · **Mohammad Mahfuzul Islam**[2]

Received: 10 January 2018 / Accepted: 3 March 2018 / Published online: 20 March 2018
© Springer International Publishing AG, part of Springer Nature 2018

Abstract
Many clandestine applications send their secret information, e.g., investigation reports, to a destination by implanting them into an image document, like forensic evidence. In that case, both the document and the implanted information are secret and equally important. To protect the document's information, called the cover information, from being disclosed, many reversible data embedding (RDE) schemes first destroy the cover information intentionally and then embed secrets into these destroyed contents. A reversible process in the receiver end retrieves both the implanted secrets and the cover information. The existing schemes suffer from less embedding capacity, i.e., embedded bits per pixel (bpp), because their reversible processes either are unable to implant bit(s) into every pixel or implant a chunk of message bits into a group of pixels where the length of the message bits is smaller than the number of pixels in the group. The article proposes a novel distortion-based RDE scheme that achieves an embedding capacity of 2^n bpp, where $0 \leq n \leq 3$. The proposed scheme destroys the information in the image before and after the data implantation task to strongly obliterate both the cover information and the embedded bits. During implementing this proposed process, the scheme establishes seven levels of encapsulated securities and, thus, strengthens the security of the scheme. The maximum embedding capacity and the lowest level of image distortion that are achieved by the proposed scheme are 8 bpp and 5 dB, respectively. These two values significantly dominate the same figures that are achieved in its competing schemes.

Keywords Reversible data hiding · Embedding capacity · Encryption · Encapsulated security · Cover image · Stego image

1 Introduction

In the field of forensic, medical, military and satellite applications or in operational parts of the industrial control units, two parties communicate between themselves to exchange secret information like personal information and images, military commands, medical images and medical history of patients, crime reports, investigation reports and forensic evidence. In this communication system, an application that uses image steganography implants the secret message bits into an official image like a person's photograph, forensic evidence, satellite image, medical image and scanned document [1,2]. After the bit implantation task, the official image, known as the cover image, is termed as the stego image. The sender generates that stego image by applying either an irreversible [3–8] or a reversible embedding algorithm [9–27]. In the irreversible case, the encoder implants bit in the mechanism that the intended decoder will be able to extract the implanted secrets only and it does not work for the reconstruction of the cover information [3]. On the other hand, the decoder extracts the secrets as well as reconstructs the original cover image from the stego image if it is a reversible mechanism [13]. The sender side sends the stego image to the destination over a communication channel. After receiving the stego image, the receiver applies the exact reversible or the irreversible algorithm, known as a decoder, to take actions accordingly. The reversible algorithms are more secure than the irreversible schemes because these apply more complex implanting rules and use more features in the shared keys and thus improve the security of the implanted data [18,19]. Nevertheless, when the contents of the cover image, i.e., its own contents, carry secret information, the reversible schemes will be inapplicable if the schemes allow the cover information to remain visible in the stego image, because any third party will then grab these secrets of the cover for their personal use [14–

✉ A. H. M. Kamal
ahmkctg@yahoo.com; kamal@jkkniu.edu

Mohammad Mahfuzul Islam
mahfuz@cse.buet.ac.bd

[1] Jatiya Kabi Kazi Nazrul Islam University, Trishal, Bangladesh

[2] Bangladesh University of Engineering and Technology, Dhaka, Bangladesh

16]. In such applications, the process of damaging the cover information in the stego image is a good technique because then any third party will not be able to guess and retrieve the cover secrets from the transmitted stego [12,14,16]. At the receiver end, the decoder extracts the implanted secrets and reconstructs the original cover image by a reversible mechanism.

The image distortion is performed by either the data implantation rules [12–14] or by applying an encryption process before the start of the bit implantation process [15,16]. In the premier case, the data implantation rules translate each block of pixels by an equal amount in a direction while implanting bits. These rules do not change the values of the block pixels, while implanting a message chunk of all zeros. During the implantation of other valued chunks, the translation amount of block pixels depends on the binary value of the implanted message chunk and the range of the block pixels [14]. Still, the block shifting strategy cannot ensure pure distortions of the cover image because it leaves lots of the cover blocks as unchanged while implanting the message chunk of 0 bits only. Many applications [14–16], therefore, first destroy the cover image entirely by an encryption process and then embed the data bits into these destroyed values. These pre-distortion based embedding schemes suffer from less embedding capacity because, for the constraint of maintenance of reversibility, these schemes implant a bit either in a working pixel [15] or in a block of pixels [16]. Additionally, many of these schemes, e.g., [15], implant a large quantity of assistant information. Consequently, the pure embedding capacity is poor.

In this article, an intentional image destruction-based reversible data hiding scheme is presented where encryptions are performed before and after the data embedment with two different 8-bit keys to maximize the level of distortions and the security of the implanted data bits. An algorithm is proposed to generate these two keys from another arbitrary 16-bit secret key, which is chosen by the data hider. This 16-bit key is placed in an arbitrary position in the image without destroying the values of the two cover pixels. The 8-bit key generation algorithm increases the security of the system. The freedom of both choosing a 16-bit key of any value by the data hider and placing the key at any position in the image have increased the robustness of the scheme. During the implantation, data bits are distributed over the binaries of each pixel in an equal distance manner to protect the least significant bit (LSB) extraction attacker from their successful mission. The number of implanted bits varies from pixel to pixel. This varying quantity of implanted bits in pixels has improved the security and the robustness of the proposed scheme. To further enhance the security, the dimension of the image is changed to make it more dissimilar to the cover image. In the proposed scheme, the embedding capacity is definable according to the demand of the application. The experimental results state that the proposed scheme dominates its competing methods [14–16,28] by all of its measuring features such as embedding capacity and image distortions.

The remaining parts of this article are organized into four more sections. Section 2 illustrates the related works on which the proposed work builds its basement. The proposed scheme is detailed in Sect. 3. Section 4 delineates the performance of the scheme over the competing schemes. Finally, Sect. 5 concludes the article.

2 Related schemes

As the proposed scheme intentionally destroys the quality of the cover image, two schemes of implanting data in the pre-distorted images [15,16] and another benchmark scheme of distorting the image during the bit implantation period by the data implantation rules [14] are accounted as the related works. The proposed method implants variant quantities of bits in the image pixels. To implant divergent quantities of bits in the pixels, the concept of Liao et al. [28] is applied in this article. Hence, though the scheme in [28] tries to manage better stego image quality, the research also deems it as a related work. These stated related schemes are briefly explained in this section.

2.1 First image distortion and then embedding

In 2015, Liao and Shu [16] proposed a reversible data embedment scheme where the image was destroyed first by encrypting it. The scheme next divides the encrypted image into blocks of $m \times n$ pixels each. In each block, it then separates these $m \times n$ pixels into two equal sized sets S_0 and S_1. It embeds a bit of information in each of the blocks by the code fragment of Algorithm 1 of Fig. 1.

At the receiver end, the decoder first decrypts the stego image by the decryption key. Likewise in the embedding phase, the decoder divides the decrypted image into blocks of $m \times n$ pixels and then separates the pixels of each block

Algorithm 1: Data Embedment

If the to-be-embedded secret bit is "0"

Flip 3 LSBs of all the pixels in the set S_0.

Else

Flip 3 LSBs of all the pixels in the set S_1.

End

Fig. 1 Data embedment by flipping LSBs of half of the pixels of a block

into two equal sized sets $\bar{S}_0$ and $\bar{S}_1$. As three LSBs of pixels of S_0 or S_1 were flipped by the embedding rules, the cover block would be generated from $\bar{S}_0 \cup \bar{\bar{S}}_1$ or $\bar{\bar{S}}_0 \cup \bar{S}_1$, where $\bar{\bar{S}}_0$ and $\bar{\bar{S}}_1$ represent the three LSBs flipped pixels of the set of $\bar{S}_0$ and $\bar{S}_1$ respectively; thus, $\{S_0 = \bar{\bar{S}}_0, S_1 = \bar{S}_1\}$ or $\{S_0 = \bar{S}_0, S_1 = \bar{\bar{S}}_1\}$. Let $H_0 = \bar{S}_0 \cup \bar{\bar{S}}_1$,and $H_1 = \bar{\bar{S}}_0 \cup \bar{S}_1$. It is certainly true that the original cover block is one of the H_0 and H_1. The pixels in H_0 and H_1 are applied in a relation to measuring the block complexity ,F_0 for H_0 and F_1 for H_1. These complexities are measured by summing up the differences between each of the pixels and the mean of its neighbors. The measured complexity is used in an analysis (more details of that analysis will be found in [16]) to identify whether H_0 or H_1 belongs to the original block. If H_0 belongs to the original block, the scheme extracts message bit '0' and reconstructs the cover block by the H_0. Otherwise, it extracts '1' and reconstructs the cover block by H_1. The scheme repeats the stated process for all blocks. Thus, it retrieves the message and the cover image from the stego image. Nevertheless, its yielded embedding capacity is very poor, as it implants a single bit only in each block of $m \times n$ pixels.

Again, in 2014, Zhang et al. [15] proposed another scheme where the embedding capacity was much better than that in [16]. The scheme first destroys the image by encrypting its contents. Considering a chessboard-like distribution of the encrypted pixels in the encrypted image plane, the pixels are divided into two parts "black" and "white". The scheme extracts the fourth LSB of each white pixel. These extracted LSBs are compressed. The scheme implants these compressed LSBs and the secret message bits by replacing the fourth LSBs of the white pixels. The data extractor collects the fourth LSBs of the white pixels. It then separates the secret message and the compressed LSBs. The original LSBs are then generated by decompressing these. These LSBs are sequentially placed in the corresponding fourth significant bit of each of the binaries of the white pixels. Thus, the message bits are extracted and the cover image is reconstructed. The compressed LSBs are addressed as an additional information. This additional information drastically reduces the pure embedding capacity.

2.2 Image distortion by embedding data

In 2014, Ong et al. [28] proposed a different RDH scheme where the image is destroyed by embedding data. The scheme divides the image into blocks of $m \times n$ pixels. In each ith block, the minimum and the maximum gray values are recorded as $\min_i$ and $\max_i$ and these two are considered as assistant information, which assist in the recovery phase. The scheme measures the range R_i of the ith block by $R_i = \max_i - \min_i + 1$. Next, it divides the gray scale for each block into P_i parts, where $P_i = 2^8 - \lceil \log_2^{Ri} \rceil$. It then embeds N_i bits of information into the ith block, where $N_i = 8 - \lceil \log_2 R_i \rceil$.

A histogram of the block pixels is computed. A partition of P_i, which contains all the frequencies of the pixels, is termed as original partition. The other partitions, i.e., $P_i - 1$ partitions, are termed as reflective partitions. For $32 < R_i \leq 63$, $P_i = 4$ and $N_i = 2$, possible embeddable message chunk is one of the {00, 01, 10, 11}. Based on the patterns of N_i message bits, the original partition is moved to another partition. This is known as the reflection process. To carry out this reflection, at first, an association of the original partition is performed with one of the parts P_i according to the message chunk N_i. Each bin of the histogram is then mapped to the corresponding position in the associated partition to perform the reflection. The methodology is called histogram association and mapping (HAM). In the recovery stage, the assistant information guides the scheme. The $\min_i$ and $\max_i$ of the ith block are used to reconstruct the original partitions and to extract the embedded N_i message bits. The scheme suffers from a big load of assistant information and, thus, it will decrease the pure embedding capacity and the space for the data embedment. It also suffers from the management of partitioning. Consider a block where min = 27, max = 57, then the original block will occupy portions of two partitions of the gray scale, i.e. partitions 0–31 and 32–63. There, the stated scheme will not operate properly as the original partition cannot be defined. Some of such limitations are reduced by Habiba et al. [13] by shifting the gray partitions just enough to allocate the histogram bins and by proposing a circular mapping scale. Kamal and Islam [12] in 2016 have improved the embedding capacity by managing the HAM of prediction errors rather than HAM of pixels. A prediction error histogram is used to shift the pixels in the gray scale. As the range of prediction errors are smaller than the range of pixels, the prediction error-based HAM scheme allows to embed more data. However, these processes also cannot embed more than 1.0 bpp.

2.3 Implanting a variant quantity of bits into the pixels

The scheme in [28] works in the spatial domain. It implants bits by replacing the least significant bits (LSBs) of the image pixels. The scheme divided the cover image into blocks of four pixels. The average distance of the pixels from the minimum one in the block is measured. If the distance value is smaller than a threshold, the data bits to be implanted replaces a few numbers of LSBs; otherwise, more LSBs are replaced by a similar number of message bits. This way, message bits are embedded in all the blocks by applying the rules of LSB substitutions. The receiver end applies the reversible process to extract the exact quantity of implanted bits from each block using the LSB extraction rules.

3 The proposed image encryption-based embedding scheme

The proposed scheme takes a cover image I of size $x \times y$ Each pixel at (i, j) location of the image is read by $I_{i,j}$. The proposed scheme produces an encrypted stego image D from I in three steps—first, the image I is encrypted using an 8-bit key K, say, the encrypted image is C; message bits are embedded into C which is then termed as the stego image S; the stego image S is further encrypted by another 8-bit key R to further strengthen the security of the system. The image generated by using the key R,say, the image D, is called the encrypted stego image. These two encryption processes ensure the security of the implanted data as well as the contents of the cover media from being realized by an adversary. Thus, three levels of securities, e.g., two-time encryptions and a data concealment process are implemented. During the data implantation, the secret bits are distributed in an equal distance over the binaries of each of the contents of the image C. For this reason, 1, 2, 4 or 8 bits are implanted in a content of C because each content of C is as long as 8 bits and the 8 bits are equally dividable into 1 bit, 2 , 4 or 8 bits. This bit distribution technique confirms the fourth level of the data embedment security, as the attack by LSBs is not possible and the distance between two implanted bits is unknown to the third party. The number of implanted bits varies from pixel to pixel. A modified policy of [28] is used in selecting the number of bits for embedding into a pixel. Thus, a fifth level of security is established in the proposed scheme. The stated keys K and R are generated from L and M, respectively, where L and M are the parts of another encoder generated 16-bit secret key E such that $E = L||M$, where .||. stands for the concatenation of the binary string. The process of generating K and R from L and M promotes the safety of the system to the sixth level of security because other than knowing the exact process of generating K and R from E, the adversary cannot breed these two keys and, thus, the challenger fails to decrypt it. The process of generating K and R is explained in Sect. 3.1. As a final step of the security measure, i.e., the seventh level of the system's security, the scheme generates hybrid stego image H by shuffling the values of the encrypted cover image C and the encrypted stego image D to make H dimensionally and visually more dissimilar to the cover image I. All of these seven levels of securities ensure stronger protection of the implanted data in the proposed scheme against attacks.

3.1 Key generation and image encryption

At the first stage of the proposed method, the data hider arbitrarily generates a 16-bit key E. The key E is either negotiated between the sender and the receiver or it is implanted into a specific part of the transmitted image. The E is partitioned into two disjoint parts L and M, i.e., $E = L||M$ as shown in Fig. 2a based on the first two bits of E. The decimal value of the first two LSBs of E is used for pointing to one of the four predefined values as the length of L. Nevertheless, in the proposal, L is allowed to be 2, 4, 8 or 12-bits. The L is formed by the defined number of most significant bits (MSBs) of E (i.e., $|L|$ {2-bit MSB, 4-bit MSB, 8-bit MSB, 12-bit MSB}, where |.| stands for the length of a string).

L and M are measured using the pseudo-code shown in Fig. 2b. In the code, the function LSBs (B, s, n) is used to extract n bits of information starting at the position s from a binary string B and the symbol $\otimes$ represents the bitwise exclusive or operation. Figure 2b also constructs two 8-bit keys K and R from L and M. These two keys K and R are used to encrypt the working image at two different phases. The scheme first encrypts each pixel of I by Eq. (1).

$$C_{i,j} = I_{i,j} \otimes R. \tag{1}$$

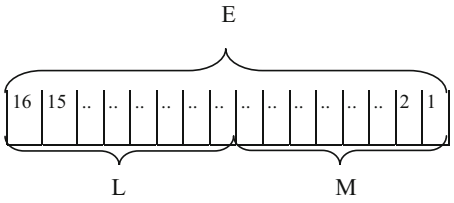
This encrypted image C is used in hiding data. The data embedment process is explained in Sect. 3.2. After the data concealment, the image C is termed as the stego image S. The stego image S is further encrypted by another key K using the Eq. (2) so that any third party cannot extract the concealed message by using any steganalyzer.

$$D_{i,j} = S_{i,j} \otimes K, \tag{2}$$

3.2 Defining the number of implantable bits in a pixel

The proposed scheme implants a varying quantity of bits in the contents of the encrypted image C. The scheme distributes the bits to be implanted over the binaries of the encrypted pixels in C. To estimate the quantity of bits that could be embedded in (i, j) location of C, the scheme first measures the average value of a negotiated number, say t, of pixels within the contents of C that were accessed immediately before, e.g., the average of the encrypted pixels located from $(i, j - t)$ to $(i, j - 1)$ in C. As the first t pixels do not have t number of immediate previous pixels, first t contents of C are not used for implanting bits. While working at (i, j) location of C, say, the average of t number of immediately accessed contents of C is m. The modulus value of m and 8, i.e., $d = \text{mod}(m, 8)$, is computed. The value of d is the estimated quantity of bits that could be embeddable in the content, i.e., the pixel at (i, j) location. However, for the constraint of equally distributing the bits over the binaries of the contents of C, the scheme allows to implant in each content either 1, 2, 4 or 8 bits of information because each content of C consists of 8 and 8 bits of the contents are equally dividable into the parts of 1, 2, 4 or 8 bits. To distribute the n bits of information over the binary of a content

Fig. 2 Process of generating encryption keys



(a) : Encryption key and its sub-keys

```
If LSBs(E, 1, 2)='00'
        L= LSBs(E, 1, 2)
        M= LSBs(E, 3, 12)
        K=L||L
        R= LSBs(M, 1, 8) ⊗ LSBs(M, 9, 6)
If LSBs(E, 1, 2)='01'
        L= LSBs(E, 1, 4)
        M= LSBs(E, 5, 12)
        K=L||L
        R= LSBs(M, 1, 8) ⊗ LSBs(M, 9, 4)
Else if LSBs(E, 1, 2)='10'
        L= LSBs(E, 1, 8)
        M= LSBs(E, 9, 8)
        K=L
        R=M
Else if LSBs(E, 1, 2)='11'
        L= LSBs(E, 1, 12)
        M= LSBs(E, 13, 4)
        K= LSBs(L, 1, 8) ⊗ LSBs(L, 9, 4)
        R= M||M
End
```

(b)*: Generation of encryption keys K and R.*

```
If d ≥ 6 then
        n=8
Else if d ≥ 4 but d<6 then
        n=4
Else if d ≥ 2 but d<4 then
        n=2
Else
        n=1
End
```

Fig. 3 Selecting the number of embeddable bits

of C in an equal distance manner, the exact value of n will be 1, 2, 4 or 8; rather than d. The value of n is measured by using the pseudo-code shown in Fig. 3. While measuring n, the immediate previous t contents of C are accounted for improving the security only.

3.3 Data embedment process

Let the length of the message to be embedded be LL. The encoder implants LL bits of information in the encrypted image C. The implantation process is done taking into account that the receiver knows the value of E, t and LL. The bit length of E, t and LL are 16, 8 and 24, respectively. If the value of LL is not 24 bits, it is padded with a sufficient number of zeros to make it 24 bits long. These 48 bits are used as the assistant information. Without knowing the assistant information, the de-embedment is not possible. The 48-bit assistant information is negotiated between the communicating parties before sending the stego image. The assistant information is sent to the receiver end through another communication channel or implanting them at a separate location in the image C. If the assistance information is sent through embedding, the method of implanting them is made complex for the convenience of increasing the security of the system. The value of E is stored in C in an arbitrary position (u, v), i.e., E is stored in (u, v) and $(u, v+1)$. The value of u and v are stored at (1, 1) and (1, 2) of C, respectively. The value of t and LL are stored in the last four contents of the C. Let LL and E be divided into 8-bit components called L_1, L_2, L_3 and E_1,E_2, respectively. The implantation of assistant information is done using the pseudo-code containing eight assignment instructions as shown in Fig. 4. In these expressions, both the keys K and R are used rather than a single one just to mislead the challenger during their accessing attempts.

For these reasons, before starting the implantation process, the data hider picks eight contents, i.e., pixel values, of C from the locations of (1, 1) and (1, 2), two arbitrarily selected

$$C_{1,1} = u$$
$$C_{1,2} = v$$
$$C_{u,v} = E_1$$
$$C_{u,v+1} = E_2$$
$$C_{x,y-3} = t \otimes K$$
$$C_{x,y-2} = L_1 \otimes R$$
$$C_{x,y-1} = L_2 \otimes R$$
$$C_{x,y} = L_3 \otimes K$$

Fig. 4 Implanting mechanism of assistant information

locations (u, v) and $(u, v+1)$, and last four positions $(x, y-3)$, $(x, y-2)$, $(x, y-1)$ and (x, y), i.e., the contents of $C_{1,1}$, $C_{1,2}$, $C_{u,v}$, $C_{u,v+1}$, $C_{x,y-3}$, $C_{x,y-2}$, $C_{x,y-1}$ and $C_{x,y}$ to protect them from being lost. These eight picked contents are concatenated with the secret message so that the decoder can reconstruct these picked values after the data extraction. Say, the concatenated result of the secret message and these eight content values is T. The length of T is, therefore, $LL+64$ bits. The resulting binaries of T are then implanted into the remaining contents of C.

During the implantation process of T, eight encrypted contents of $C_{1,1}$, $C_{1,2}$, $C_{u,v}$, $C_{u,v+1}$, $C_{x,y-3}$, $C_{x,y-2}$, $C_{x,y-1}$ and $C_{x,y}$ and t contents of $C_{1,3}$ to $C_{1,t+3-1}$ are not utilized in hiding data and, thus, these are skipped by the data hider. These values assist the data extractor to retrieve the data and the cover values. The data hider implants each n bits of information, where $n \varepsilon \{1, 2, 4, 8\}$, into each remaining values of C. The value of n is computed by using Fig. 3. Let the n bits of data be $b_n \cdots b_2 b_1$, which is a part of T. During the implantation period, the embedded bits, b_z, $1 \leq z \leq n$, are distributed over the binaries of the processed pixel in an equally distant manner, e.g., if $n = 4$, the 1st, 3rd, 5th and 7th bits of each 8 bits encrypted pixel are modified, rather than into n LSBs. Such distribution not only enhances the security and robustness of the scheme, but also increases the distortions in the stego image. Each processing pixel $C_{i,j}$ is first stored in a temporary variable P. After the data implantation, this stego pixel is assigned to the (i, j) location of the stego image S. Thus, C is kept unchanged. To realize the implantation method, say, the binary of a working pixel $C_{i,j}$ is P, i.e., $P = \text{Dec2Bin}(C_{i,j})$, where Dec2Bin returns the binary of a decimal number. The scheme divides P into n components, each of which is $l = 8/n$ bits long, e.g., if $C_{i,j} = 123$, then $P = 01111011$; and using Fig. 3 the computed value of n is 4. The components of P are $\{01, 11, 10, 11\}$. Let each chunk of P be p_z, for $1 \leq z \leq n$, e.g., $p_1 = 01$, $p_2 = 11$, $p_3 = 10$ and $p_4 = 11$. The sequential concatenation of the entire p_z is P. Each b_z is embedded into each part p_z by using Eq. (3).

$$\tilde{p}_z = p_z \otimes b_z. \tag{3}$$

The stego pixel $\tilde{\boldsymbol{P}}$ is formed by sequentially concatenating all the $\tilde{\boldsymbol{P}}_{\boldsymbol{z}}$ and $\tilde{\boldsymbol{P}}$ is the (i, j)th pixel of the stego image S, i.e., $\boldsymbol{S_{i,j}} = \tilde{\boldsymbol{P}}$. By implanting all the message bits of T into the values of C, the stego image S is formed. The stego image is then again encrypted using Eq. (2).

3.4 Hybrid stego image generation

A hybrid stego image H is constructed by shuffling the pixels of the encrypted cover image C and the encrypted stego image D to change the dimension of the transmitted stego image by $2x \times y$ and to present a more meaningless image to the third party.

To shuffle the pixels of these two images, first, the hybrid image H of size $2x \times y$ is marked into black and white cells in a chessboard fashion. The pixels in C and D are assigned into black and white cells, respectively, as shown in Fig. 5. It is noticeable that the first row starts with C, while the second row starts with D and so forth in the following. In this shuffling method, rather than starting with the pixels of C in every row, the chessboard-like distribution is chosen to influence the adversary into straying.

3.5 Message extraction process

The decoder receives the hybrid image H. It then writes all the black-located pixel values into $C_{i,j}$ and all the white-

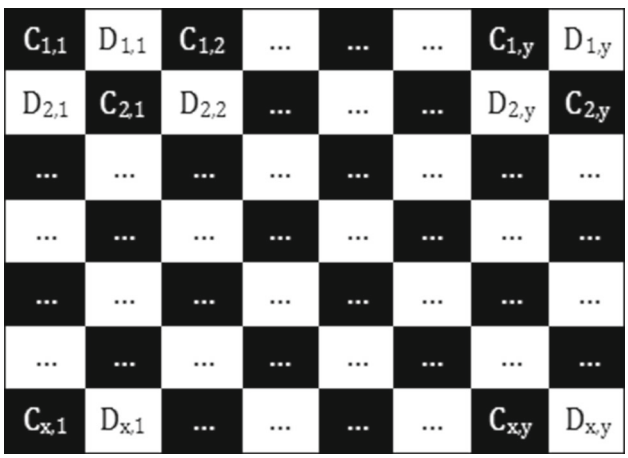


Fig. 5 Formation of hybrid stego image H

located pixel values into $D_{i,j}$ from H, where $1 \leq i \leq x$, $1 \leq j \leq y$. The values of $C_{1,1}$ and $C_{1,2}$ point to a position in C, where the key E is stored. Let $u = C_{1,1}$ and $v = C_{1,2}$. The bits of E were stored in $C_{u,v}$ and $C_{u,v+1}$ by the data hider. Therefore, the key E is extracted from $E = \text{Dec2Bin}(C_{u,v}) \,||\, \text{Dec2Bin}(C_{u,v+1})$. The pseudo-code shown in Fig. 2 is executed to generate the keys K and R from E. The number of associated contents t, which were used for computing the quantity of implanted bits, is found by using Eq. (4).

$$t = C_{x,y-3} \otimes K, \tag{4}$$

Similarly, the last three pixels of C decrypted using Eq. (5) are used to find out the length of the total embedded bits LL.

$$\left.\begin{aligned} L_1 &= C_{x,y-2} \otimes R \\ L_2 &= C_{x,y-1} \otimes R \\ L_3 &= C_{x,y} \otimes K \end{aligned}\right\}, \tag{5}$$

where $LL = L_1 \,||\, L_2 \,||\, L_3$. Equation (6) decrypts the encrypted stego image D by the decryption key K to find the stego image S.

$$S_{i,j} = D_{i,j} \otimes K. \tag{6}$$

Let $\tilde{P} = \text{Dec2Bin}(S_{i,j})$, $Q = \text{Dec2Bin}(C_{i,j})$ and $\tilde{Q} = \tilde{P} \otimes Q$. One bit of data was embedded into every chunk of l bits of $\tilde{P}$, where $l = 8/n$. The value of n is measured for each pixel of C from the immediately accessed t pixels. The process is stated in Sect. 3.2. Every $(z-1) \times l + 1$ positioned bit of $\tilde{Q}$, where $1 \leq z \leq n$ represents the bit of message stream that was implanted by the encoder end at (i, j) location. The process of extracting message bits from each (i, j) location of stego image S is outlined in Fig. 6.

Figure 6 does change the stego pixel at (i, j) locations, where $(i, j) \in \{(1, 1)$ to $(1, t), (u, v), (u, v+1), (x, y-3), (x, y-2), (x, y-1), (x, y)\}$. The extracted messages from all the stego pixels are concatenated to form the final message string. The extracted message also contains the original values of $C_{1,1}$, $C_{1,2}$, $C_{u,v}$, $C_{u,v+1}$, $C_{x,y-3}$, $C_{x,y-2}$, $C_{x,y-1}$ and $C_{x,y}$. These values are reconstructed to form the original encrypted image C. Finally, the cover image I is generated using Eq. (7).

$$I_{i,j} = C_{i,j} \otimes R. \tag{7}$$

1. $\tilde{P} = Dec2Bin(S_{i,j})$
2. $Q = Dec2Bin(C_{i,j})$
3. $\tilde{Q} = \tilde{P} \otimes Q$

Fig. 6 Extracting message from a single stego pixel

4 Analysis of the experimental results

The experiments are conducted on different image datasets in MATLAB. A few of these are presented in Fig. 7. The objective of the research work is to increase both the embedding capacity and the image distortions. Therefore, these two are analyzed in this part of the thesis. The proposed method is compared to the scheme proposed by Ong et al. [14], Liao et al.'s [16], Zhang et al. [15] and Liao et al.'s [28]. Though the scheme presented in [28] is not an intentional image distortion-based process, this scheme is used in the comparison, as the concept of variant bit implantation of this scheme is used in the proposed work. Very generally, the scheme in [28] provides higher image quality than the other experimented schemes.

4.1 Capacity analysis

In two different experiments, the proposed scheme is allowed to implant up to 8 bpp and a variable amount of quantity as required. The results of 50 images are depicted in Fig. 8. The experimental results delineate that the proposed scheme dominates the others noticeably. While the competing schemes presented in [14–16] fail to embed even 1 bpp, the proposed scheme implants up to 8 bpp. The scheme presented in [28] provides an embedding capacity within 2–3 bpp because, according to the objective of the scheme, to preserve better image quality, it is allowed to implant either 3 or 2 bits in each pixel. The embedding capacity of 8 bpp is achieved by fixing $n = 8$. This capacity is several multiples of that found in the other schemes. In another experiment, the value of n is predicted from 4 previous pixels for $t = 4$. The achieved capacity varies from 2.1 to 5.7 bpp depending on the image properties. In 92% of images (46 out of 50), the obtained embedding capacity in the proposed scheme is higher than the highest performing competing scheme [28]. Thus, it is proved that the proposed scheme outperforms in all the experimented images and the obtained capacity is several multiples of others.

The embedding capacity, presented for the proposed scheme, is measured regarding the image size of $x \times y$. Nevertheless, the size of the hybrid image is $2x \times y$. The hybrid image is transmitted over the Internet. Hence, with respect to the size of the hybrid image, the embedding capacity of the proposed scheme is half of the presented figure. Hence the embedding capacity is 4 bpp for $n = 8$ and 1.1–2.9 bpp for variant bits implantation. These figures also noticeably domi-

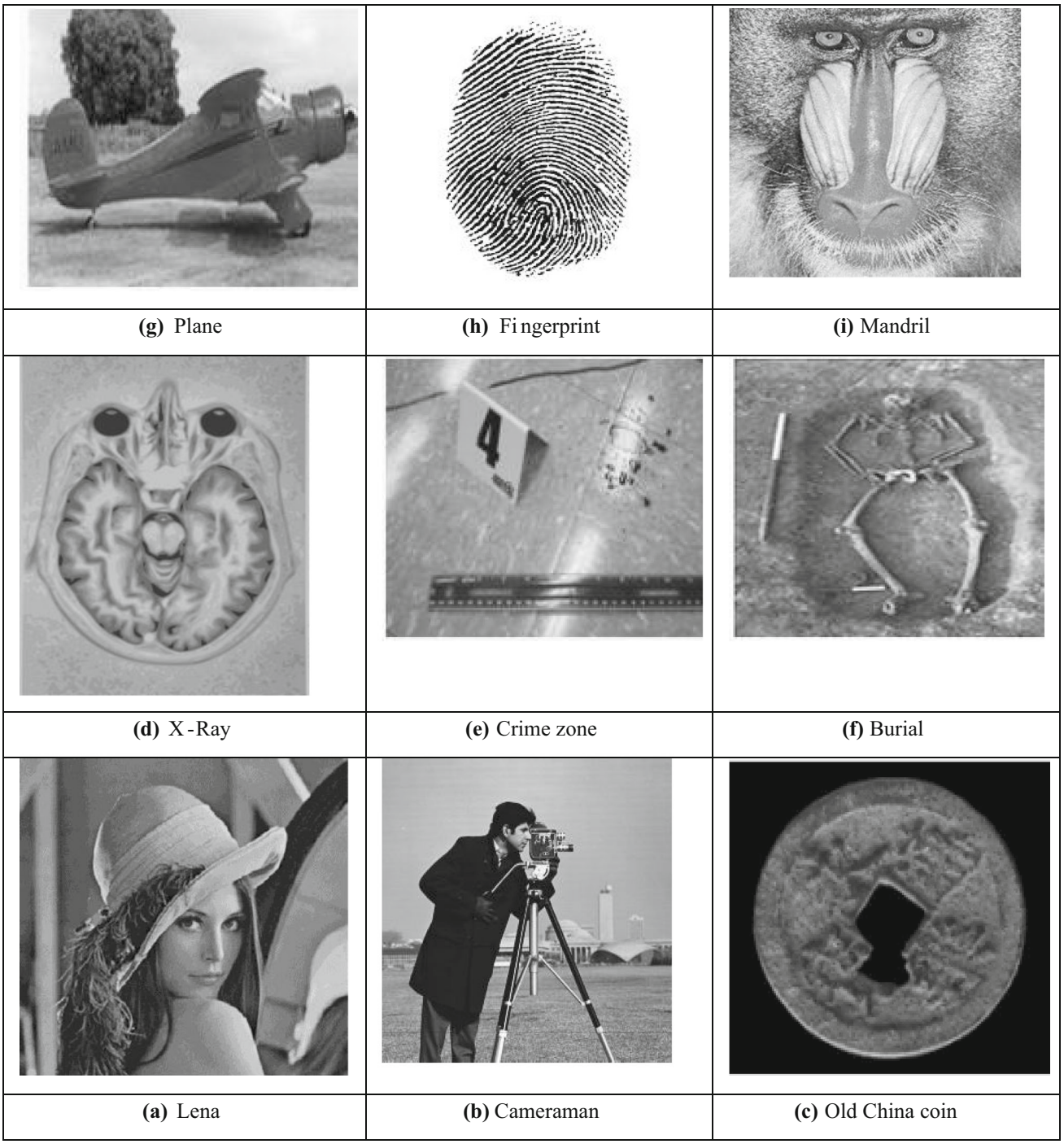


Fig. 7 Sample cover images that are used in the experiments

nate the achieved embedding capacity found by the schemes presented in [14–16] for all the images and [28] for many images.

4.2 Distortion analysis

The cover image, the encrypted cover image, the encrypted stego image and the hybrid stego image of the fingerprint image are demonstrated in Fig. 9a–d, respectively. From the encrypted cover image shown in Fig. 9b and the encrypted stego image shown in Fig. 9c, nothing is realizable about the cover information, because the stated process of the proposed scheme destroys the visual and statistical information about the cover image. The hybrid stego image is depicted in Fig. 9d. This is also like a noisy image. Hence, the objective of destroying the cover information

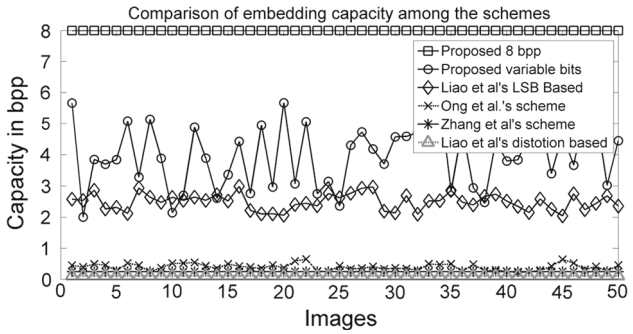


Fig. 8 Capacity of the proposed and its competing schemes

in the stego image is fully achieved through the proposed scheme.

The distortions in the stego image and in the encrypted image are found due to the abrupt changes to the image pixels. Therefore, the pixel histograms of the cover image, encrypted cover image and encrypted stego image exhibit dissimilar properties, as shown in Fig. 10a–c, respectively.

To compare the distortion levels, the quality of the stego image is measured by the peak signal to noise ratio (PSNR) [29]. The Eq. (8) is used to measure the PSNR values.

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{255^2}{\mathrm{MSE}} \right). \tag{8}$$

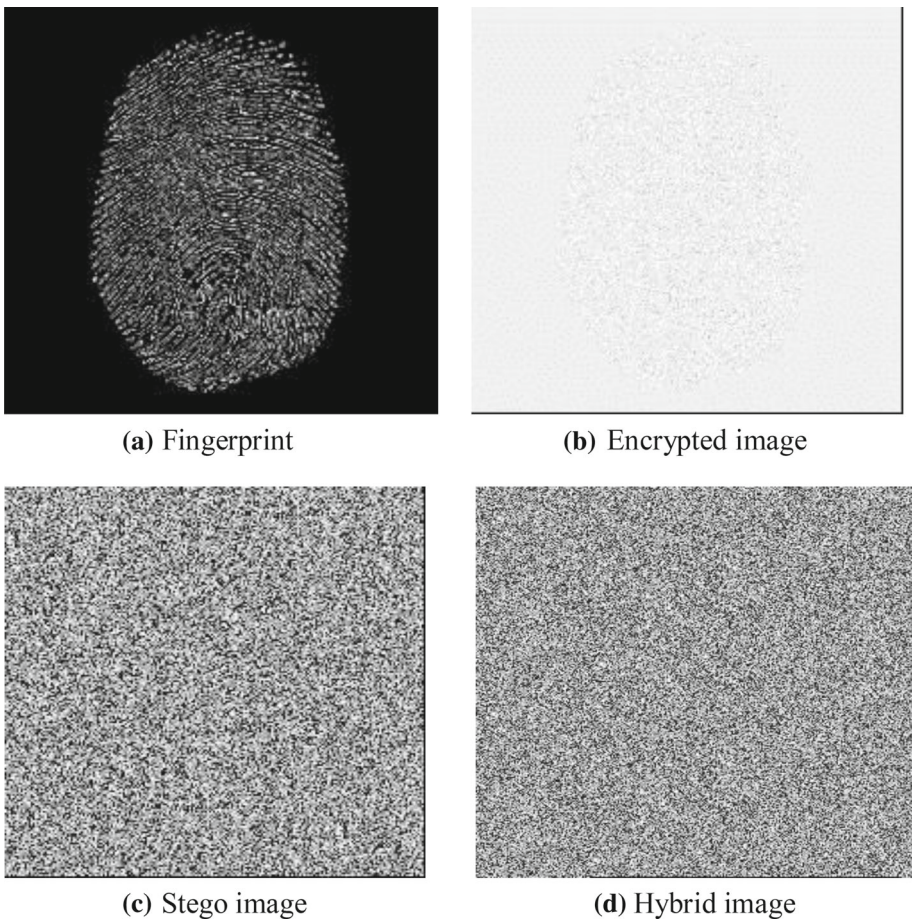
The mean square error (MSE) is computed using the Eq. (9).

$$\mathrm{MSE} = \frac{1}{x \times y} \sum_{i=1}^{x} \sum_{j=1}^{y} \left(S_{i,j} - I_{i,j} \right)^2. \tag{9}$$

The results are depicted in Fig. 11. The results state that the proposed scheme provides the lowest PSNR values in all the images because the proposed scheme encrypts the image for two times and implants message bits at different, but equally apart positions in the binaries of each content. The distortion level achieved in [28] is not demonstrated because the objective of this scheme is to manage higher image quality, while the proposed scheme tries to maximize the distortion level.

To check the state of distortions for various levels of embedding capacities, the proposed scheme embeds into each cover image separately for four times by fixing $n = 1$, $n = 2$, $n = 3$ and $n = 4$, respectively. First, 1 bit is embedded into each pixel and the level of distortion is measured. Thereafter, 2, 4, and 8 bits are implanted in each pixel in sub-

Fig. 9 Images at various processing steps



(a) Fingerprint **(b)** Encrypted image

(c) Stego image **(d)** Hybrid image

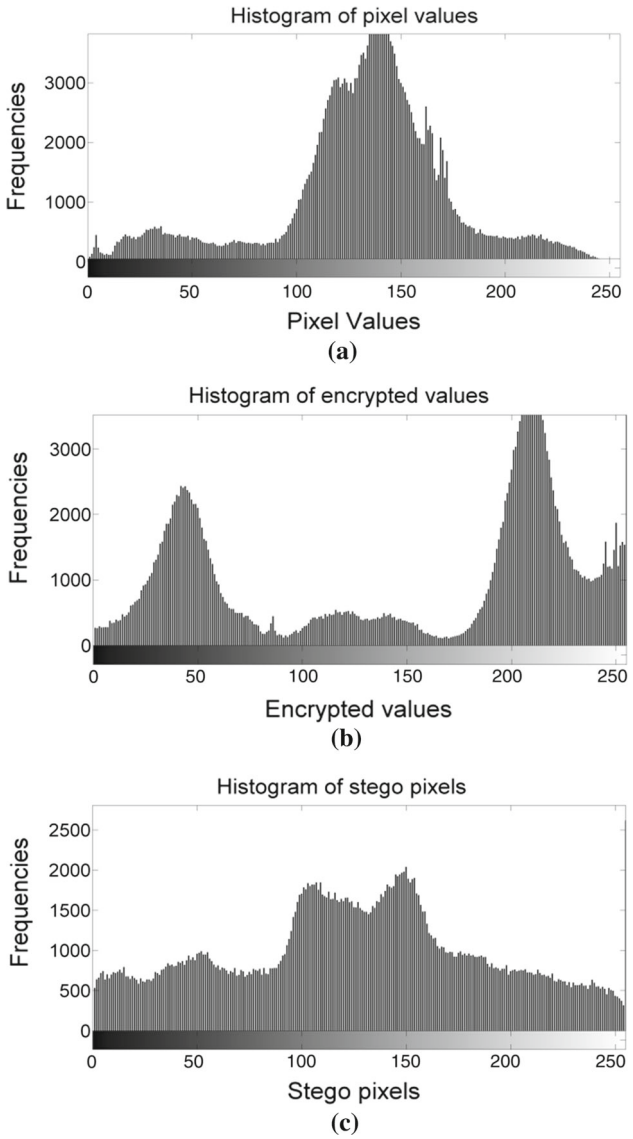


Fig. 10 Histograms of **a** the cover image; **b** the encrypted image; and **c** the stego image, respectively

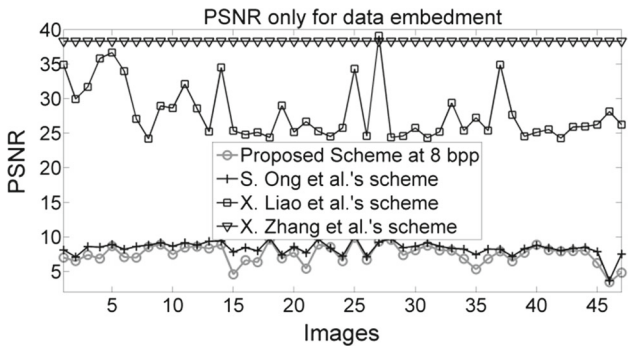


Fig. 11 Comparison of PSNR values among the schemes

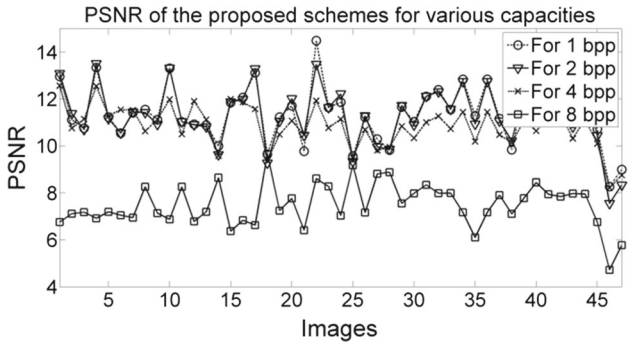


Fig. 12 PSNR comparisons for various embedding capacity

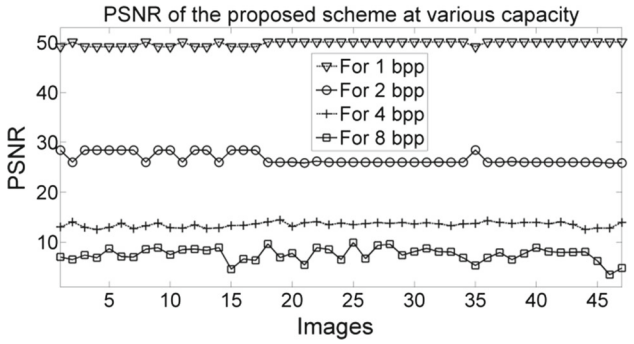


Fig. 13 PSNR of the proposed scheme for only data embedment at various capacities

sequent executions and the distortion amounts are measured, respectively. The distortions for the embedding capacity of 1, 2, 4 and 8 bpp are analyzed in the experiments. The results are demonstrated in Fig. 12. It is observed that the values of PSNR decrease in all the images when the embedding capacity is increased. This is happening because, with the increment in the embedding capacity, more bits of each pixel are altered in the bit implantation phase. This, indeed, increases the number of changes in the stego contents regarding their cover contents. As a result, the PSNRs decrease in the proposed scheme and that decrease is proportionate to the embedding capacity.

Both the encryption and the data concealment processes are responsible for lowering the values of PSNR. The effects of only the data embedment on the values of PSNR, excluding the effect of the image encryption, for different levels of embedding capacity are also examined in the proposed scheme. The results are delineated in Fig. 13. The PSNR is decreased with the increment in the embedding capacity. The reason is same as it is observed in the Fig. 12. When the embedding capacity is improved, the scheme implants more bits per pixel. This implies that the scheme alters more numbers of bits in each pixel's binary.

4.3 Effects of encryption keys on image distortions

Encryption keys do not have any effect on the embedding capacity; however, these affect the image distortions by the

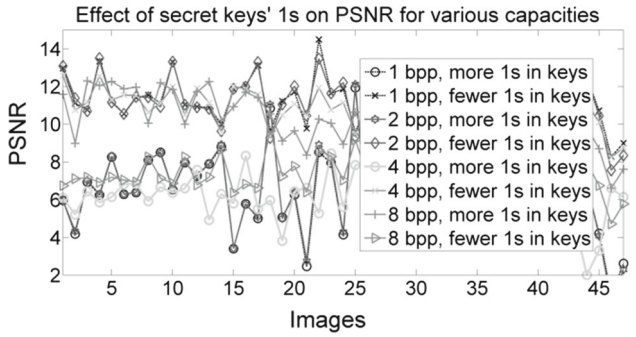


Fig. 14 Effect of encryption keys to PSNR values

frequency of '1's in their binary values. It is investigated that keys with more '1's in their binary values destroy the image on a large scale because during the exclusive or operation in the Eq. (1) and the Eq. (2) each binary bit 1 in the encryption key modifies the respective binary bit in the image pixel. To check the effect, the values of key $\{R, K\}$ are set to {100, 137} and {235, 183} in two separate execution periods. The produced results are compared. The key values of {235, 183} contain more '1's in their binaries than for the key values of {100, 137}. The results are demonstrated in Fig. 14. In the figure, it is noticeable that the number of '1s in the binaries of the keys plays an important role in affecting the values of PSNR. For all the investigated embedding capacities, it is observed that the values of PSNR decrease for the uses of keys with more '1s. Thus, it is found that {235, 183} valued keys destroy the image quality more than the keys of {100, 137}.

5 Conclusion

The distortion-ased reversible data hiding schemes do not care about the quality of the image. Rather, these try to destroy all the cover information in the stego image including the smallest one. These schemes provide higher embedding capacity,because it is easy to manage the pixel values during the implantation of more quantity of message bits in a distorted image. Nevertheless, to the best of the author's knowledge, none of the reversible schemes in the literature provide an embedding capacity of 8 bpp. The proposed work shows enough novelty in achieving both the embedding capacity of up to 8 bpp and degrades the image quality up to a value of 5 dB. Besides, it offers the data hider either to embed different quantities of bits into the pixels or to select the embedding capacity to one of the four values—1, 2, 4 and 8 bpp. The proposed scheme is very effective in hiding the large volume of data and securing the secret messages and evidence related to forensic, medical, military, law-enforcing agency application. The seven levels of security features are implemented into the scheme in an encapsulation way. Hence, breaking the security is difficult. As a whole, it will be a useful contribution to the field of reversible data hiding arena.

Acknowledgements The author AHMK is funded by the ICT division of the Ministry of Post, Telecommunication and Information Technology of the Government of Bangladesh through a fellowship program. Therefore, the authors like to acknowledge the stated ministry of Bangladesh.

Compliance with ethical standards

Conflict of interest The authors do not have any economical interest from the aticle. The first author is a PhD student and working under the supervision of the second author. To meet the requirement for achieving the PhD degree, the first author has to publish his research works on ranked journals published by well-recognized publishers. Therefore, the authors have chosen this journal to publish the work. Both the authors are aware of the ubmission. The first author is a fellow of ICT division of the Ministry of Post, Telecommunication and Information Technology of the Government of Bangladesh. However, the fellowship neither covers any publication charges nor claims any financial interest from the research.

Author contributions The first author, AHMK is a PhD student of the Department of Computer Science and Engineering of the Bangladesh University of Engineering and Technology. He is working under the supervision of second author, MMI. Hence, the whole work was supervised and guided by MMI. Mr. MMI has been consulted all the way to the progress of the research work by the author AHMK. Mr. AHMK completed the experiments and made the draft of the manuscript. Mr. MMI revised the manuscript and gave final approval to submit it to that journal.

References

1. Ulutas, M., Ulutas, G., Nabiyev, V.V.: Medical image security and EPR hiding using Shamir's secret sharing scheme. J. Syst. Softw. **84**(3), 341–353 (2011)
2. Kamal, A.H.M., Islam, M.M.: Facilitating and securing offline e-medicine service through image steganography. Healthc. Technol. Lett. **1**(2), 74–79 (2014)
3. Wien, H., Chen, T.-S.: A novel data embedding method using adaptive pixel pair matching. Inf. Forensics Secur. IEEE Trans. **7**(1), 176–184 (2012)
4. Brindha, S., Vennila, I.: Hiding fingerprint in face using scattered LSB embedding steganographic technique for smart card based authentication system. Int. J. Comput. Appl. **26**(10), 51–55 (2011)
5. Chao, R.-M., et al.: A novel image data hiding scheme with diamond encoding. EURASIP J. Inf. Secur. **1**, 658047 (2009)
6. Hong, W., Tung-Shou, C., Chih-Wei, L.: Data embedding using pixel value differencing and diamond encoding with multiple-base notational system. J. Syst. Softw. **85**(5), 1166–1175 (2012)
7. Liao, X., Wen, Q., Zhang, J.: A steganographic method for digital images with four-pixel differencing and modified LSB substitution. J. Vis. Commun. Image Represent. **22**(1), 1–8 (2011)
8. Kamal, A.H.M., Islam, M.M.: Enhancing the performance of the data embedment process through encoding errors. J. Electron. **5**(4), 79–95 (2016)
9. Kamal, A.H.M., Islam, M.M.: Enhancing embedding capacity and stego image quality by employing multi predictors. J. Inf. Secur. Appl. **32**, 59–74 (2017)

10. Kamal, A.H.M., Islam, M.M.: Boosting up the data hiding rate multi cycle embedment process. J. Vis. Commun. Image Represent. **40**, 574–588 (2016)
11. Kamal, A.H.M., Islam M.M.: Capacity improvement of reversible data hiding scheme through better prediction and double cycle embedding process. In: Proceedings of IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), 16–18 December, Kolkata, India (2015)
12. Kamal, A.H.M., Islam M.M.: Enhancing the embedding payload by handling the affair of association and mapping of block pixels through prediction errors histogram. In: Proceedings of International Conference on Networking, Systems and Security (NSysS), BUET, 5–8 January, Dhaka (2016)
13. Habiba, S., Kamal, A.H.M., Islam, M.M.: Enhancing the robustness of visual degradation based HAM reversible data hiding. J. Comput. Sci. **12**(2), 88–97 (2016)
14. Ong, S.Y., Wong, K.S., Tanaka, K.: Scrambling-embedding for JPEG compressed image. Signal Process. **109**, 38–53 (2015)
15. Zhang, X., et al.: Efficient reversible data hiding in encrypted images. J. Vis. Commun. Image Represent. **25**(2), 322–328 (2014)
16. Liao, X., Shu, C.: Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels. J. Vis. Commun. Image Represent. **28**, 21–27 (2015)
17. Hong, W., Chen, T.-S.: A local variance-controlled reversible data hiding method using prediction and histogram-shifting. J. Syst. Softw. **83**(12), 2653–2663 (2010)
18. Hong, W.: Adaptive reversible data hiding method based on error energy control and histogram shifting. Opt. Commun. **285**(2), 101–108 (2012)
19. Tai, W.-L., Yeh, C.-M., Chang, C.-C.: Reversible data hiding based on histogram modification of pixel differences. Circ. Syst. Video Technol. IEEE Trans. **19**(6), 906–910 (2009)
20. Ou, B., et al.: Reversible data hiding based on PDE predictor. J. Syst. Softw. **86**(10), 2700–2709 (2013)
21. Yang, W.-J., et al.: Efficient reversible data hiding algorithm based on gradient-based edge direction prediction. J. Syst. Softw. **86**(2), 567–580 (2013)
22. Chen, X., et al.: Reversible watermarking method based on asymmetric-histogram shifting of prediction errors. J. Syst. Softw. **86**(10), 2620–2626 (2013)
23. Tsai, P., Hu, Y.-C., Yeh, H.-L.: Reversible image hiding scheme using predictive coding and histogram shifting. Signal Process. **89**(6), 1129–1143 (2009)
24. Lu, Y.-Y., Huang, H.-C.: Adaptive reversible data hiding with pyramidal structure. Vietnam J. Comput. Sci. **1**(3), 179–191 (2014)
25. Ma, X., et al.: High-fidelity reversible data hiding scheme based on multi-predictor sorting and selecting mechanism. J. Vis. Commun. Image Represent. **28**, 71–82 (2015)
26. Leung, H.Y., et al.: Adaptive reversible data hiding based on block median preservation and modification of prediction errors. J. Syst. Softw. **86**(8), 2204–2219 (2013)
27. Chang, I.-C., Hu, Y.-C., Chen, W.-L., Lo, C.-C.: High capacity reversible data hiding scheme based on residual histogram shifting for block truncation coding. Signal Process. **108**, 376–388 (2015)
28. Liao, X., Wen, Q.Y., Zhang, J.: A steganographic method for digital images with four-pixel differencing and modified LSB substitution. J. Vis. Commun. Image Represent. **22**(1), 1–8 (2011)
29. Zhao, Z., Luo, H., Lu, Z.-M., Pan, J.-S.: Reversible data hiding based on multilevel histogram modification and sequential recovery. Int. J. Electron. Commun. (AEÜ) **65**, 814–826 (2011)