

# Pixel rearrangement based statistical restoration scheme reducing embedding noise

Arijit Sur · Vignesh Ramanathan · Jayanta Mukherjee

© Springer Science+Business Media, LLC 2012

**Abstract** In this paper, a block based steganographic algorithm has been proposed where a sequence of secret bits are embedded into a set of pixels by rearranging the pixel locations. This algorithm has been devised as an improvement over existing statistical restoration based algorithms in order to reduce the additive noise which occurs due to embedding. It is shown that the proposed scheme substantially reduces the additive noise compared to existing statistical restoration based schemes.

**Keywords** Steganography · Steganalysis · Pixel swapping · Additive noise

## 1 Introduction

Steganography is the art of hiding information in an innocent looking cover objects and thus visual and statistical undetectability is one of the major concerns in the steganographic security. In recent years, a good number steganalysis algorithms (e.g. [8, 25]) were reported using first order image statistics. Blind attacks also use the statistical features to train their steganalytic classifiers. It is observed in the literature (e.g. [7, 25]) that, the performance of additive noise based blind attacks (WAM [5]) degrades substantially if *never compressed images* [14] are used as cover images. It is also observed that some of the first order image statistics based targeted attacks

---

A. Sur (✉)  
Dept. of CSE, IIT Guwahati, Assam, India  
e-mail: arijit@iitg.ernet.in

V. Ramanathan  
Dept. of EE, IIT Kharagpur, West Bengal, India  
e-mail: vigneshram.iitkgp@gmail.com

J. Mukherjee  
Dept. of CSE, IIT Kharagpur, West Bengal, India  
e-mail: jay@cse.iitkgp.ernet.in

(e.g. [7, 25]) detect LSB type embedding more accurately when *never compressed images* are used. Restoration of the first order statistics of the stego image may be a possible countermeasure to those kind of steganalytic attacks. Motivated by this fact, a few statistical restoration based schemes had been proposed in the past [17, 18, 20, 21]. The main problem of restoring image statistics (e.g. [17, 18, 20, 21]) is that an extra amount of additive noise is added during restoration. This extra additive noise makes these algorithms more vulnerable against additive noise based blind attacks like WAM [5].

In this paper, a block based adaptive scheme, called *Pixel Rearrangement based Steganographic Algorithm (PRSA)*, has been introduced in order to reduce the embedding noise for statistical restoration based schemes. In the proposed approach, message strings (binary combination of bits) are represented by different pixel ordering in a set of pixels. For a fixed number of message bits, all possible message strings are denoted by a particular pixel ordering of a set. If a input message string does not correspond to the pixel ordering of the stego message, its ordering is rearranged to the desired one.

A brief survey on LSB embedding, possible attacks on it, statistical restoration based schemes and their limitations is presented in Section 2. In Section 3, proposed *PRSA* scheme is introduced. An empirical analysis is provided in Section 4 to show the higher reduction of additive noise by the proposed approach compared to other statistical restoration based schemes. Experimental results are presented in Section 5. Finally, the paper is concluded in Section 6.

## 2 Related work

### 2.1 LSB steganography and corresponding attacks

Least Significant Bit (LSB) Replacement is one of the most well referred steganographic methods where secret bits are embedded by replacing least significant bits of the image. LSB replacement can be detected by structural asymmetry based attacks such as Sample Pair Analysis, proposed by Dumitrescu et al. [2], RS Steganalysis, proposed by Fridrich et al. [4] etc.

To overcome this undesirable asymmetry, the decision of changing the least significant bit is randomized. In this case, if the message bit does not match with the pixel's least significant bit (*lsb*), the pixel's *lsb* is either increased or decreased by 1. This technique is popularly known as *LSB Matching*. To further reduce the noise, the use of a binary function of two cover pixels to embed the data bits is suggested in [11]. In rest of the paper, this scheme is referred as Improved LSB Matching (*ILSBM*) and the LSB Matching scheme is referred as *LSBM*.

There are many spatial domain blind attacks which can detect the LSB embedding scheme such as Wavelet Absolute Moment (WAM) steganalysis [5], statistical moments based attacks [1, 15, 16, 23, 24], etc. But blind attacks are not always very accurate if *never compressed images* are used for embedding [25]. In *never compressed images*, high frequency components are presented in a greater proportion. These high frequency components generally mask the steganographic noise and make the steganographic detection difficult for additive noise based blind attacks.

It is also observed in the literature that image histogram based attacks perform well when *never compressed images* are used for embedding. Histogram based

attack on LSB embedding (sometimes called  $\pm 1$  embedding) was first introduced by Harmsen [6]. This attack was further extended to detect LSB Matching algorithm by Ker in [8]. More recently, a few image histogram based targeted attacks [7, 25] were reported to perform well especially for *never compressed images*.

Restoration of the first order statistics of the stego image may be a possible countermeasure to this kind of histogram based steganalytic attacks. Some statistical restoration based steganographic techniques along with their limitations are discussed in next subsection.

## 2.2 Statistical restoration based embedding and its limitations

As a countermeasure to these histogram based attacks, restoration of cover image statistics has been employed in the recent past. Provos' Outguess algorithm [12] was an early attempt which tries to preserve the original histogram even after LSB embedding. Eggers et al. [3] have suggested a more rigorous approach using histogram-preserving data-mapping (HPDM) and adaptive embedding respectively. Another restoration based approach is Model Based Steganography, proposed by Sallee [13]. A very recent approach in this direction is proposed by Solanki et al. in [17] and [18] for JPEG steganography. A statistical restoration scheme for non Gaussian cover images is proposed by Sur et al. in [21] which can restore histogram efficiently for spatial domain images.

In [20], another embedding algorithm called *Pixel Swapping based Steganographic Algorithm (PSSA)* is proposed which is based on the simple idea of swapping two pixels in the spatial domain embedding. In this work, first two consecutive elements in a pseudorandom walk ( $\xi$ ) (with shared secret seed) of an 8 bit gray scale cover image are taken (say  $a$  and  $b$ ). The image pixels corresponding to  $a$  and  $b$  locations are denoted as  $I_a$  and  $I_b$ . If  $I_a$  is greater than  $I_b$ , the embedded bit is taken as 1, on the other hand if  $I_b$  is greater than  $I_a$  the embedded bit is taken as 0. If  $I_a$  and  $I_b$  are the same or the absolute difference between  $I_a$  and  $I_b$  is greater than or equal to prescribed threshold value, the pair is not used for embedding. Now if present secret bit is 1 and present  $I_b$  is greater than  $I_a$ ,  $I_a$  and  $I_b$  pixels are swapped. Similarly swapping is done if present secret bit is 0 and present  $I_a$  is greater than  $I_b$ . Then next two consecutive pixels are taken from  $\xi$  and this process is continued until all secret bits are being embedded. Authors have shown experimentally that *PSSA* scheme greatly outperforms LSB matching and its improved version [11] against histogram based attacks.

It is experimentally observed that, for any statistical restoration based schemes, an extra amount of noise is added due to restoration. Although histogram based attacks can not detect these statistical restoration based schemes, these extra additive noise makes those algorithms more vulnerable against additive noise based blind attacks. So reduction of embedding noise in statistical restoration based steganography is a major issue. In this paper, an adaptive scheme has been proposed which not only restores the image histogram of the cover images, but also adds relatively less noise due to embedding.

## 3 Proposed *PRSA* scheme

In this paper, a steganographic scheme called *Pixel Rearrangement based Steganographic Algorithm (PRSA)* has been proposed which can be considered as an

improvement over statistical restoration based schemes. In this proposed scheme, a block based adaptive algorithm has been introduced in order to reduce the embedding noise.

### 3.1 Adaptive block selection

It is observed from the steganographic literature that image zones with higher information contents are suitable for embedding because the high frequency components present in these high textured areas, effectively mask the steganographic noise. In the proposed scheme, embedding pixels are chosen from the relatively high textured areas of the cover image. Firstly, the image is partitioned into non-overlapping blocks having  $9 \times 9$  pixels (say  $\beta$ ). Partitioning the image in non-overlapping blocks enables us to adapt the embedding based on local texture properties. Each block ( $\beta$ )(having  $9 \times 9$  pixels) undergoes with 2D Discrete Cosine Transform (DCT) and DCT coefficients are quantized by standard JPEG quantization matrix. After



(a) Block Selection Threshold ( $\lambda_{nnz}$ ) = 5



(b) Block Selection Threshold ( $\lambda_{nnz}$ ) = 10



(c) Block Selection Threshold ( $\lambda_{nnz}$ ) = 15



(d) Block Selection Threshold ( $\lambda_{nnz}$ ) = 20

**Fig. 1** Embedding block locations for different block threshold ( $\lambda_{nnz}$ ). **a** Block selection threshold ( $\lambda_{nnz}$ ) = 5. **b** Block selection threshold ( $\lambda_{nnz}$ ) = 10. **c** Block selection threshold ( $\lambda_{nnz}$ ) = 15. **d** Block selection threshold ( $\lambda_{nnz}$ ) = 20

rounding off the coefficients into nearest integer, quantized DCT coefficients of each blocks ( $\beta$ ) are obtained. In this proposed method, number of non-zero quantized DCT coefficients of the block [say  $nnz(\beta)$ ] are used to track the texture of the block [10]. Those blocks ( $\beta$ ) (having  $9 \times 9$  pixels) are used for embedding where non-zero quantized DCT coefficients of the block [ $nnz(\beta)$ ] are greater than a prescribed threshold. Let it be called Block Selection Threshold (say  $\lambda_{nnz}$ ). This Block Selection Threshold ( $\lambda_{nnz}$ ) controls the trade off between the quality of stego image and the payload. For example, higher value of this threshold ( $\lambda_{nnz}$ ) increases the payload but degrades the stego image quality or increases the chance of detection. In Fig. 1, embedding block locations are shown (denoted by black squares) with different Block Selection Thresholds ( $\lambda_{nnz}$ ).

### 3.2 Message representation using pixel ordering

In this paper, the proposed *PRSA* algorithm is called *PRSA*( $n, m$ ). This implies that,  $m$  bits message string is embedded in a set of  $n$  pixels. So each blocks ( $\beta$ ) [having  $9 \times 9$  pixels] are further partitioned into non-overlapping sets of  $n$  pixels. Let  $b_k$  be one such set of  $n$  pixels. The set  $b_k$  is considered suitable for embedding if all  $n$  pixels in the set are distinct and the difference between maximum and minimum pixel values is less than a prescribed threshold. Let this threshold is called Noise Threshold ( $\tau_{TH}$ ). This Noise Threshold ( $\tau_{TH}$ ) controls the noise addition due to embedding. Now there are  $n!$  possible pixel arrangements for the set ( $b_k$ ). Since  $m$  bits message is going to be embedded in  $b_k$ , this is only possible if  $2^m \leq n!$ .

In this paper, the proposed *PRSA*( $n, m$ ) algorithm is called *PRSA*(3, 2) with  $n = 3$  and  $m = 2$ . Let the pixel values of the set be denoted by  $a, b$  and  $c$  such that  $a < b < c$ . For example, if pixels of a set is [143, 141, 142],  $a = 141, b = 142$  and  $c = 143$ , and thus the present pixel ordering becomes [ $cab$ ]. With this convention,  $2^2 = 4$  different message strings are represented by  $3! = 6$  different orderings as given in Table 1. It is worth mentioning here that this information is a shared secret to both sender and receiver. The *PRSA* algorithm is described in the next sub-section.

### 3.3 The PRSA algorithm

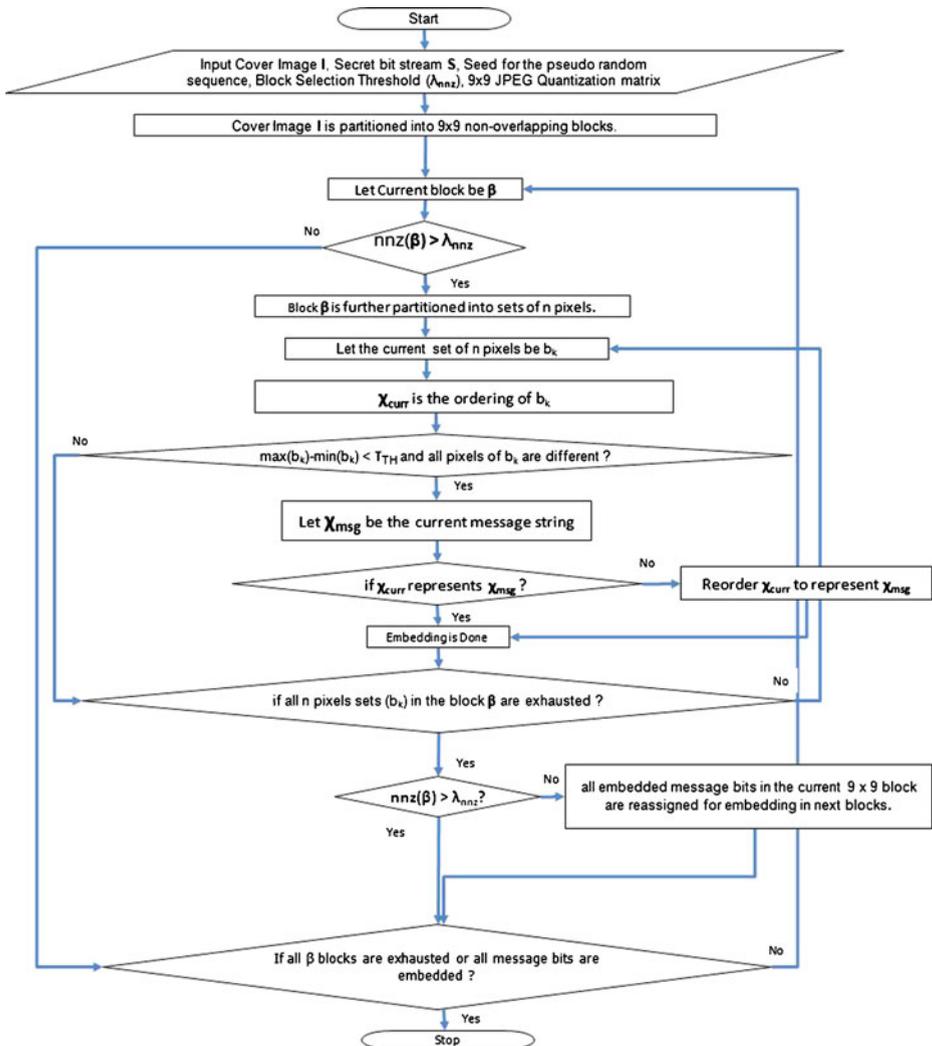
In the embedding algorithm, the cover image ( $I$ ) is partitioned into non-overlapping blocks of ( $\beta$ ) [having  $9 \times 9$  pixels]. If the non-zero quantized DCT coefficients of the block [ $nnz(\beta)$ ] is greater than the prescribed threshold ( $\lambda_{nnz}$ ), the block ( $\beta$ ) is used for the embedding. In this algorithm,  $9 \times 9$  quantization matrix is formed by using the same standard JPEG  $8 \times 8$  quantization matrix (with 75% quality factor) and the 9th row and the 9th column are kept the same as the 8th row and the 8th column respectively.

**Table 1** Message string representation using pixel ordering

Pixel ordering	Message string
$abc$	00
$acb$	00
$bac$	01
$bca$	01
$cab$	10
$cba$	11

Now each selected block is again partitioned into non-overlapping sets of  $n$  pixels. Let one such set ( $b_k$ ) be considered. If all the pixels in a set ( $b_k$ ) are different to each other and  $\max(b_k) - \min(b_k) < \tau_{TH}$  (refer to Section 3.2), the set is used for embedding; otherwise the set is not considered suitable for the embedding. To embed an  $m$  bits message string, the pixel ordering of the current set is checked according to the rules described in Table 1. If the pixel ordering is the same as the desired ordering representing current  $m$  bit message sequence, no change is required; otherwise they are reordered to represent the current  $m$  bit message string.

There may be a case that the embedding suitability of a block ( $\beta$ ) [ $nnz(\beta)$  is greater than  $\lambda_{nnz}$ ] is destroyed due to addition of the embedding noise. In decoder,



**Fig. 2** Flow chart of the embedding algorithm of proposed PRSA scheme

at the time of the extraction, such block is not considered for the extraction since it lost its embedding suitability due to embedding noise. To counter this problem, the block suitability is checked after the embedding and if it is found that the block ( $\beta$ ) becomes unsuitable due to embedding noise, the corresponding message bits (embedded in this particular block) are embedded to next possible blocks. But the changes due to embedding in that particular block are kept unchanged to mark this block unsuitable. So at the time of extraction this block is treated as unsuitable. This modification is expected to reduce the payload. But it is experimentally observed that the said incident (suitable blocks become unsuitable due to embedding noise) is a rare one. Experimental evidence reveals that on the average the payload reduces by 0.012 of the total payload for an entire image.

A flowchart of the embedding scheme is shown in Fig. 2. A step-wise description of the algorithm is given below:

**Algorithm** *Pixel Rearrangement based Steganographic Algorithm (PRSA)*

**Input:** Cover Image  $I$ , Secret Bit Sequence  $S$ , present secret bit string is represented by  $\chi_{\text{msg}}$

**Input Parameters:** Shared secret seed for generating pseudorandom sequence, Block Selection Threshold ( $\lambda_{\text{nnz}}$ ), Noise Threshold ( $\tau_{\text{TH}}$ ),  $9 \times 9$  JPEG Quantization matrix

**Output:** *Stego Image  $I_s$*

1. Cover image ( $I$ ) is partitioned into non-overlapping blocks ( $\beta$ ) [having  $9 \times 9$  pixels].
2. If the non-zero quantized DCT coefficients of a block [ $\text{nnz}(\beta)$ ] is greater than the Block Selection Threshold ( $\lambda_{\text{nnz}}$ ), the block is used for the embedding.
3. The selected block ( $\beta$ ) is further divided into non-overlapping sets of  $n$  pixels. A set ( $b_k$ ) is taken for embedding using a pseudorandom walk with shared secret seed.
4. The current set pixel ordering (let it be denoted as  $\chi_{\text{curr}}$ ) is determined as described in Section 3.2
  - if** ( $(\max(b_k) - \min(b_k) < \tau_{\text{TH}})$  & (all the pixels of  $b_k$  are different))
    - $m$  message bits are taken for embedding such that  $2^m \leq n!$ .
    - The set pixel ordering required to represent the current message string (let it be denoted as  $\chi_{\text{msg}}$ ) is also determined as described in [Section 3.2].
    - if** ( $\chi_{\text{curr}} == \chi_{\text{msg}}$ )
      - no change is necessary
    - else**
      - $\chi_{\text{curr}}$  is reordered to represent  $\chi_{\text{msg}}$ . If more than one options exist, then reorder  $\chi_{\text{curr}}$  to the nearest ordering of the  $\chi_{\text{curr}}$ .
    - endif**
  - else**
    - this set ( $b_k$ ) is *not suitable* for embedding.
  - endif**
  - if** (all sets ( $b_k$ ) are exhausted of the current block ( $\beta$ ))
    - if** (non-zero quantized DCT coefficients of the currently embedded block [ $\text{nnz}(\beta)$ ] is greater than the Block Selection Threshold ( $\lambda_{\text{nnz}}$ ))
      - Go to step 2

- ```

else
    all embedded message bits in the current block ( $\beta$ )
    are reassigned for embedding in next blocks.
    Go to step 2
endif
else
    Go to step 4
endif

```
5. Steps 2 to 4 are continued until all message bits are embedded or all suitable blocks ( $\beta$ ) have been exhausted.

**End Pixel Rearrangement based Steganographic Algorithm (PRSA)**

### 3.4 Extraction algorithm

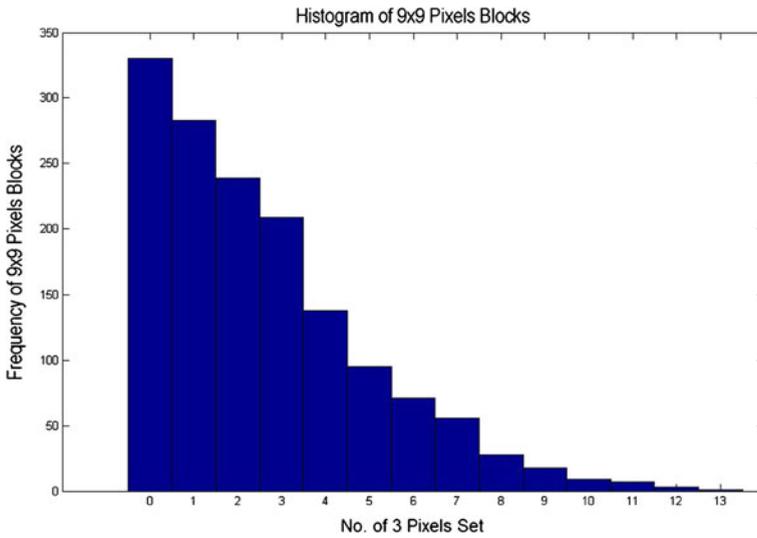
Extraction algorithm is quite simple. In the decoder, first the suitable block ( $\beta$ ) is located using the same Block Selection Threshold ( $\lambda_{\text{nnz}}$ ) which is a shared secret between the encoder and the decoder. From the selected block ( $\beta$ ), a set of  $n$  ( $n = 3$ ) pixels ( $b_k$ ) is chosen for extracting using a pseudorandom walk with the shared secret seed. If the current set satisfies the embedding conditions (all pixels are different and  $\max(b_k) - \min(b_k) < \tau_{\text{TH}}$ ), message bits are extracted from that set. The pixel ordering of that set is determined as described in Section 3.2. The corresponding message string, represented by the present ordering of  $b_k$ , is taken as message bits using Table 1. The total message sequence is obtained from all the suitable sets. Since every step exactly follow the same order as in the encoder, message bits are extracted in the same sequence as it was embedded in the encoder. If a less number of bits are embedded than maximum capacity, then a terminator string is used to stop the extraction algorithm.

### 3.5 Illustrative example

In this subsection, an illustrative example is given to explain the proposed *PRSA*(3, 2) algorithm. Let us assume that the prescribed Noise Threshold ( $\tau_{\text{TH}}$ ) is 3 and the pixels of the set is [132, 131, 130]. For the given set, all the pixels are different and the condition  $\max(b_k) - \min(b_k) < \tau_{\text{TH}}$  is satisfied. So the given set can be used for embedding. Now, the present pixel ordering is determined as [cba] as described in Section 3.2. Now, let the message bits to be embedded be [00]. So, the required ordering is [abc] or [acb]. The nearest ordering is [abc]. Swapping of a and c is made. So after embedding the set is [130, 131, 132].

### 3.6 Embedding capacity

The embedding capacity is adaptive to the image and depends on two parameters namely Block Selection Threshold ( $\lambda_{\text{nnz}}$ ) and Noise Threshold ( $\tau_{\text{TH}}$ ). For a given Block Selection Threshold ( $\lambda_{\text{nnz}}$ ) if the number of suitable embedding blocks ( $\beta$ ) is



**Fig. 3** The frequency of blocks ( $\beta$ ) having  $9 \times 9$  pixels with respect to the number of embeddable 3 pixels set

$\alpha$  and in if average number of  $n$ -pixels set in one block ( $\beta$ ) is  $\gamma$  for a given Noise Threshold ( $\tau_{TH}$ ), then the embedding capacity (say  $\omega$ ) is determined as follows:

$$\omega = 2 \times \alpha \times \gamma \tag{1}$$

where

$$\gamma = \frac{\sum_{i=1}^{\alpha} \gamma_i}{\alpha} \tag{2}$$

and  $\gamma_i$  = number of suitable  $n$ -pixels sets in  $i$ th block ( $\beta$ ) [having  $9 \times 9$  pixels].

A histogram describing the frequency of blocks ( $\beta$ ) having  $9 \times 9$  pixels with respect to the number of embeddable  $n$ -pixels set (here  $n = 3$ ) is shown in Fig. 3.

### 4 Noise analysis

#### 4.1 Embedding noise for PRSA(3, 2) scheme

In this subsection, the embedding noise is computed for PRSA(3, 2). Let an  $n$  pixel set satisfy the embedding criterion as described in Section 3.2. The pixel values of the set are denoted by  $a, b$  and  $c$  such that  $a < b < c$ . Let  $\tau_1 = b - a$  and  $\tau_2 = c - b$ , then  $\tau = (\tau_1 + \tau_2)$ . The range of  $\tau_1$  and  $\tau_2$  is  $1 \leq \tau_1, \tau_2 \leq (\tau - 1)$ . The net noise embedded in the set in different cases are summarized in Table 2.

From the Table 2, it is observed that the noise added due to conversion from  $[abc]$  to  $[cba]$  (or vice versa) is  $2(\tau_1 + \tau_2)$ , as the resulting noise is  $|a - c| + |b - d| + |c - a|$  i.e.  $2 \times |a - c| = 2\tau = 2(\tau_1 + \tau_2)$ . Now the noise added due to conversion between  $[bca]$  and  $[cab]$  is again  $2(\tau_1 + \tau_2)$ , as in this case  $|b - c| + |c - a| + |a - b| = \tau_1 + \tau + \tau_2 = \tau_1 + \tau_1 + \tau_2 + \tau_2 = 2(\tau_1 + \tau_2)$ .

**Table 2** Noise analysis table for PRSA(3,2)

| Pixel ordering | Message string       |           |                      |                      |
|----------------|----------------------|-----------|----------------------|----------------------|
|                | 00                   | 01        | 10                   | 11                   |
| <i>abc</i>     | 0                    | $2\tau_1$ | $2(\tau_1 + \tau_2)$ | $2(\tau_1 + \tau_2)$ |
| <i>acb</i>     | 0                    | $2\tau_1$ | $2(\tau_1 + \tau_2)$ | $2(\tau_1 + \tau_2)$ |
| <i>bac</i>     | $2\tau_1$            | 0         | $2(\tau_1 + \tau_2)$ | $2\tau_2$            |
| <i>bca</i>     | $2\tau_1$            | 0         | $2\tau_2$            | $2(\tau_1 + \tau_2)$ |
| <i>cab</i>     | $2(\tau_1 + \tau_2)$ | $2\tau_2$ | 0                    | $2\tau_1$            |
| <i>cba</i>     | $2(\tau_1 + \tau_2)$ | $2\tau_2$ | $2\tau_1$            | 0                    |

**Lemma 1** *The worst case average noise added to a single pixel to embed 1 bit message using PRSA(3, 2) is  $\frac{7\tau-1}{36}$  where  $\tau$  is the prescribed threshold.*

The average noise per set ( $\eta_{\text{blk}}$ ) during embedding using PRSA(3, 2) scheme can be computed using Table 2 as follows:

$$\eta_{\text{blk}} = \frac{(2(\tau_1 + \tau_2) \times 8) + (2\tau_1 \times 6) + (2\tau_2 \times 4)}{24} \quad (3)$$

Since  $(\tau_1 + \tau_2) = \tau$ , the  $\eta_{\text{blk}}$  can be written as

$$\eta_{\text{blk}} = \tau + \frac{\tau_1}{6} \quad (4)$$

It is assumed that message bits are randomly generated. So the occurrence of four different message strings is equiprobable. Again, since it is assumed that pixels ordering in natural images are random, six pixel orderings (namely *abc*, *acb*, *bac*, *bca*, *cab*, *cba*) are also equiprobable. With these assumptions, 24 equiprobable cases of noise addition can occur due to embedding which are tabulated in Table 2. For computing the average noise added due to embedding of 2 message bits in a 3 pixels set, the sum of errors contributed on account of embedding in each case, is divided by 24 (the number of possible cases) which is represented in (3).

Maximum noise is added when  $\tau_1 = (\tau - 1)$ . It implies  $\tau_1 \geq \tau_2$  since  $1 \leq \tau_1 \leq \tau - 1$ . Hence, the average noise per set under worst case situation ( $\eta_{\text{blk}}^{\text{wc}}$ ) would be

$$\eta_{\text{blk}}^{\text{wc}} = \frac{7\tau - 1}{6} \quad (5)$$

Now, for PRSA(3,2) each set consists of 3 pixels and can embed 2 bit messages then worst case average noise added to a single pixel due to embed a 1 bit message using PRSA(3,2) scheme is  $\frac{7\tau-1}{6 \times 3 \times 2} = \frac{7\tau-1}{36}$  where  $\tau$  is a prescribed threshold.

#### 4.2 Noise comparison with existing statistical restoration based schemes

In most statistical restoration based schemes, cover image pixels are divided into two groups. Pixels from one group are used for embedding and pixels from another group are used for restoration. For example, let a pixel (from embedding group) is changed from 145 to 143 due to embedding. In restoration process, a pixel (with value 143) from restoration set (if it exists) be changed to 145 to restore the image histogram. This procedure is the same as the swapping two pixels (say 145 and 143) from their respective positions. PSSA scheme [20] follows the same approach for embedding

**Table 3** Noise analysis table for PSSA

| Pixel     | Message string |    |
|-----------|----------------|----|
| ordering  | 0              | 1  |
| <i>ab</i> | 0              | 2τ |
| <i>ba</i> | 2τ             | 0  |

bits using swapping of two pixels. The only difference is that, the restoration is done concurrently with the embedding. However, it also suffers from the introduction of additional noise in the stego image, like other statistical restoration schemes. In this work, we take the PSSA as a representative scheme of that category for comparing the performance with the proposed approach.

For the PSSA scheme, the amount of noise embedded in a set of two pixels in different cases is summarized in Table 3. The average noise added per set in the PSSA is

$$\eta_{blk}^{pssa} = \frac{4\tau}{4} \tag{6}$$

Now, for PSSA each set consists of 2 pixels and can embed 1 bit messages. Then average noise added to a single pixel due to embed a 1 bit message using PSSA scheme is  $\frac{4\tau}{4 \times 2 \times 1} = \frac{\tau}{2}$

Thus the reduction in noise added due to embedding ( $\eta_{diff}$ ) in PRSA compared to any other restoration based schemes in worst case can be written as

$$\eta_{diff} = \frac{\tau}{2} - \frac{7\tau - 1}{36} = \frac{11\tau + 1}{36} \tag{7}$$

### 4.3 Choice of PRSA parameters

In this work the value of *n* and *m* for the PRSA(*n,m*) are kept as 3 and 2, respectively. One constraint for choosing *n* and *m* is that  $n! \geq 2^m$  (refer Section 3.2). There may be other choices for (*n, m*) such that (4, 4), (5,6) etc. However if *n* is high, number of suitable *n* pixels sets become very less because all the pixels of the *n* pixel set should be different. There are a few reasons why (*n,m*) is chosen as (3,2), they are as follows:

1. It is observed experimentally that quite a good number of 3 pixels sets ( $b_k$ ) are available satisfying the *n* pixels set suitability i.e. (all pixels are different and  $\max(b_k) - \min(b_k) < \tau_{TH}$ ).
2. The implementation of the algorithm becomes easier if *n* and *m* are not very high. For example, the number of entries of Table 2 becomes 384 (24 rows and 16 columns) for *n* = 4, *m* = 4 and that becomes 7680 (120 rows and 64 columns) for *n* = 5, *m* = 6.

**Table 4** Noise analysis for different parameters of PRSA scheme

| <i>n, m</i> | Noise per block                                                                                    |
|-------------|----------------------------------------------------------------------------------------------------|
| 3, 2        | $(7/6) * \tau_1 + \tau_2 = \tau + \frac{\tau_1}{6}$ , since $(\tau_1 + \tau_2) = \tau$ (refer (4)) |
| 4, 4        | $(45/32) * \tau_1 + (7/4) * \tau_2 + (21/16) * \tau_3$                                             |
| 5, 6        | $(333/160) * \tau_3 + (29/20) * \tau_4 + (29/20) * \tau_1 + (669/320) * \tau_2$                    |

**Table 5** Noise per bit per pixel for different parameters of PRSA scheme

| $n, m$ | Noise per bit embedded per pixel |
|--------|----------------------------------|
| 3, 2   | 0.361                            |
| 4, 4   | 0.279                            |
| 5, 6   | 0.235                            |

Embedding noise for different pair of  $n, m$  are shown in the Table 4 where  $\tau_i$ s are differences between two consecutive valued pixels. For example, for a 5 pixels block ( $abcde$ ) where  $a < b < c < d < e$ ,  $\tau_1 = b - a$ ,  $\tau_2 = c - b$ ,  $\tau_3 = d - c$  and  $\tau_4 = e - d$ .

For a typical case (when  $\tau_1 = \tau_2 = \tau_3 = \tau_4 = 1$ ), noise per pixel per bit embedded is given in Table 5. It can be observed from Table 5, the noise per bit per pixel is reducing for higher value of  $n$ . Since the number of suitable  $n$  pixels sets is very less for a higher  $n$ , payload of the scheme becomes very less.



(a) Cover Image

(b) Stego Image at Emb. Rate = 0.10  
bpp(c) Stego Image at Emb. Rate = 0.15  
bpp(d) Stego Image at Emb. Rate = 0.25  
bpp

**Fig. 4** Cover image and stego images at different embedding rates. **a** Cover image. **b** Stego image at emb. rate = 0.10 bpp. **c** Stego image at emb. rate = 0.15 bpp. **d** Stego image at emb. rate = 0.25 bpp

## 5 Experimental results

### 5.1 Visual quality

A standard cover image and the corresponding stego images at different embedding rates are presented in Fig. 4. It can be observed that there is hardly any visual artifacts due to the embedding using the proposed scheme.

As it is explained in Section 4.2, the *PSSA* scheme is taken as the representative for any statistical restoration based techniques for noise addition, two visual quality metrics like PSNR and Structural Similarity Measure (SSIM) [22] between cover and stego images for the proposed *PRSA* scheme is compared only with the *PSSA* scheme. In Table 6, the PSNR between cover and stego images are listed for different embedding rates for this purpose. Similar results for SSIM are also presented in Table 7. It is observed from Tables 6 and 7, that the PSNR of cover and stego images for the proposed *PRSA* scheme are relatively higher than the *PSSA* at the same embedding rate and the SSIM values are almost comparable.

### 5.2 Image data set

For studying the performance of the *PRSA* algorithm, experiments have been conducted on one thousand *never compressed test images* from the UCID Database [14]. In this context, a *never compressed image* denotes an image which is not previously compressed. Two tests have been carried out to check the applicability of the UCID database. In the first test where the cover images are taken from previously JPEG compressed dataset, the detection performance of the WAM steganalyzer is observed on standard LSB Matching algorithm for different embedding rates. Same experiment is repeated for the *never compressed cover images* from the UCID Database [14]. Results are given in Table 8

**Table 6** PSNR comparison of PRSA(3,2) with PSSA for some standard images at different embedding rates

| Images<br>emb rate | Schemes<br>→ | PSNR at different emb. rates (bpp) |       |       |       |       |
|--------------------|--------------|------------------------------------|-------|-------|-------|-------|
|                    |              | 0.03                               | 0.12  | 0.25  | 0.36  | 0.43  |
| Peper              | PSSA         | 60.99                              | 49.51 | 40.75 | 35.84 | 33.59 |
|                    | PRSA(3,2)    | 62.68                              | 51.13 | 42.33 | 36.86 | 33.81 |
| Lena               | PSSA         | 59.59                              | 48.99 | 41.65 | 36.80 | 33.37 |
|                    | PRSA(3,2)    | 61.37                              | 50.54 | 42.88 | 37.73 | 33.60 |
| Crowd              | PSSA         | 59.82                              | 51.09 | 43.86 | 37.65 | 32.94 |
|                    | PRSA(3,2)    | 61.69                              | 52.47 | 45.28 | 38.75 | 33.33 |
| Goldhill           | PSSA         | 60.65                              | 49.87 | 41.61 | 36.12 | 32.94 |
|                    | PRSA(3,2)    | 62.13                              | 51.38 | 43.08 | 37.22 | 33.10 |
| Airplane           | PSSA         | 57.27                              | 48.66 | 43.24 | 39.09 | 34.99 |
|                    | PRSA(3,2)    | 58.85                              | 49.94 | 44.06 | 39.68 | 35.38 |
| Man                | PSSA         | 59.83                              | 50.04 | 42.52 | 36.65 | 32.51 |
|                    | PRSA(3,2)    | 61.71                              | 51.78 | 43.94 | 37.67 | 32.76 |
| Boats              | PSSA         | 57.63                              | 48.42 | 42.83 | 38.14 | 33.85 |
|                    | PRSA(3,2)    | 59.32                              | 49.72 | 43.93 | 39.07 | 34.40 |
| Harbour            | PSSA         | 59.84                              | 52.68 | 45.08 | 38.89 | 33.61 |
|                    | PRSA(3,2)    | 59.79                              | 52.04 | 45.28 | 39.37 | 34.03 |

**Table 7** SSIM comparison of PRSA(3,2) with PSSA for some standard images at different embedding rates

| Images<br>emb rate | Schemes<br>→ | SSIM at different emb. rates (bpp) |        |        |        |        |
|--------------------|--------------|------------------------------------|--------|--------|--------|--------|
|                    |              | 0.03                               | 0.12   | 0.25   | 0.36   | 0.43   |
| Peper              | PSSA         | 0.9982                             | 0.9926 | 0.9856 | 0.9027 | 0.8653 |
|                    | PRSA(3,2)    | 0.9992                             | 0.9952 | 0.9967 | 0.9074 | 0.8719 |
| Lena               | PSSA         | 0.9979                             | 0.9919 | 0.9842 | 0.9296 | 0.8953 |
|                    | PRSA(3,2)    | 0.9993                             | 0.9939 | 0.9694 | 0.9285 | 0.8908 |
| Crowd              | PSSA         | 0.9990                             | 0.9961 | 0.9926 | 0.9565 | 0.9316 |
|                    | PRSA(3,2)    | 0.9991                             | 0.9970 | 0.9983 | 0.9659 | 0.9373 |
| Goldhill           | PSSA         | 0.9983                             | 0.9944 | 0.9892 | 0.9237 | 0.8815 |
|                    | PRSA(3,2)    | 0.9991                             | 0.9962 | 0.9881 | 0.9333 | 0.8815 |
| Airplane           | PSSA         | 0.9982                             | 0.9916 | 0.9866 | 0.9582 | 0.9412 |
|                    | PRSA(3,2)    | 0.9989                             | 0.9930 | 0.9877 | 0.9600 | 0.9390 |
| Man                | PSSA         | 0.9987                             | 0.9943 | 0.9899 | 0.9335 | 0.8887 |
|                    | PRSA(3,2)    | 0.9990                             | 0.9964 | 0.9814 | 0.9452 | 0.8894 |
| Boats              | PSSA         | 0.9978                             | 0.9910 | 0.9846 | 0.9484 | 0.9144 |
|                    | PRSA(3,2)    | 0.9992                             | 0.9946 | 0.9828 | 0.9571 | 0.9235 |
| Harbour            | PSSA         | 0.9992                             | 0.9966 | 0.9932 | 0.9620 | 0.9278 |
|                    | PRSA(3,2)    | 0.9993                             | 0.9980 | 0.9911 | 0.9720 | 0.9395 |

### 5.3 Performance comparison of PRSA scheme against histogram based attacks

The security of the PSSA steganographic algorithm is evaluated against Ker's HCF COM based attacks using calibration by down sampling Image [8], Ker's HCF COM based attacks using Adjacency Histogram [8], Jun Zhang et al.'s attack [25], Fangjun Huang et al.'s attack [7] and Xiaolong Li et al.'s attack [9]. For comparison, LSB matching (LSBM) and Improved LSB Matching (ILSBM) [11] are considered since most of the above attacks generally targeted LSB embedding [(±1) embedding] based schemes.

A receiver operating characteristic (ROC) is used to evaluate the performance of the classifier. A receiver operating characteristic (ROC) curve is defined as a plot of the sensitivity against (1 – specificity) in signal detection theory as its discrimination threshold is varied. It is generally used for measuring the performance of a binary classifier. The ROC is also equivalently obtained by plotting the fraction of true positives (TPR = true positive rate) versus the fraction of false positives (FPR = false positive rate). Detection accuracy ( $P_{\text{detect}}$ ) is computed using (8) and (9) as described in [19].

$$P_{\text{detect}} = 1 - P_{\text{error}} \quad (8)$$

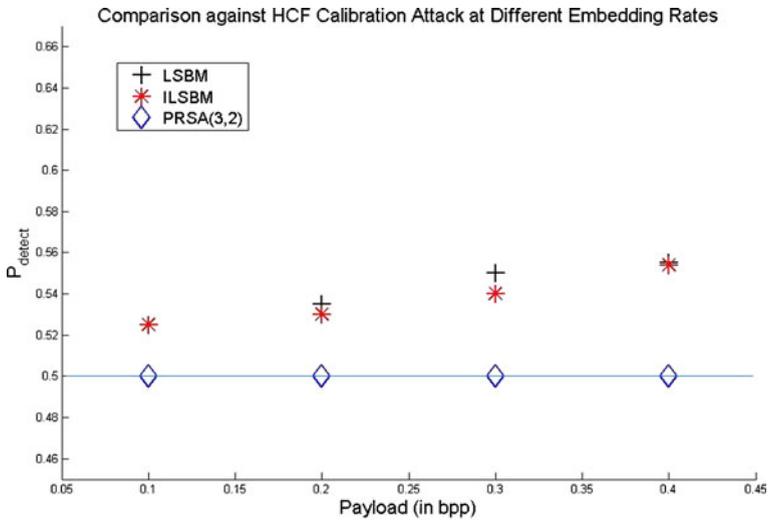
**Table 8** Comparison between never compressed images and previously compressed images

| Embedding rate | Image dataset           | AROC | $P_{\text{detect}}$ |
|----------------|-------------------------|------|---------------------|
| 0.10           | Previously compressed   | 0.43 | 0.86                |
|                | Never compressed (UCID) | 0.12 | 0.57                |
| 0.30           | Previously compressed   | 0.45 | 0.91                |
|                | Never compressed (UCID) | 0.20 | 0.64                |
| 0.50           | Previously compressed   | 0.47 | 0.95                |
|                | Never compressed (UCID) | 0.25 | 0.67                |

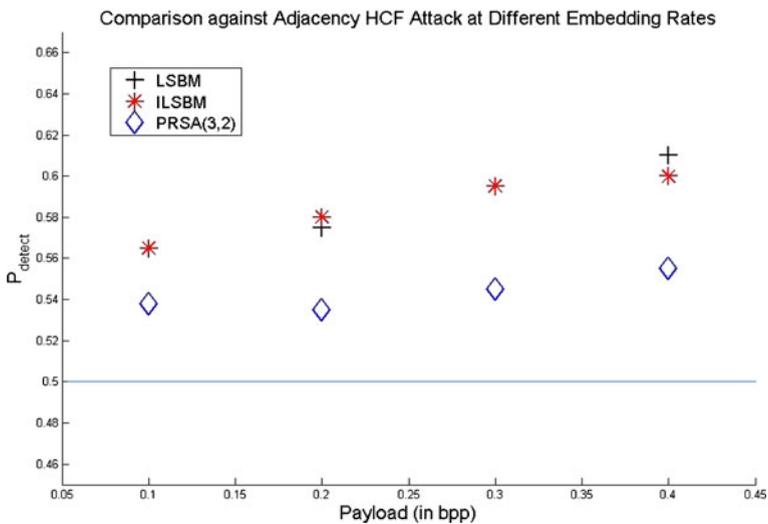
$$P_{\text{error}} = \frac{1}{2} \times P_{\text{FP}} + \frac{1}{2} \times P_{\text{FN}} \tag{9}$$

where  $P_{\text{FP}}$ ,  $P_{\text{FN}}$  are the probabilities of false positive and false negative respectively. A value of  $P_{\text{detect}} = 0.5$  shows that the classification is as good as random guessing and  $P_{\text{detect}} = 1.0$  shows a classification with 100% accuracy.

In Figs. 5 and 6, the proposed *PRSA* algorithm is compared with the *LSBM* and the *ILSBM* against different targeted attacks. Comparison with the *PSSA* [20] is



(a) Against HCF Calibration Attack

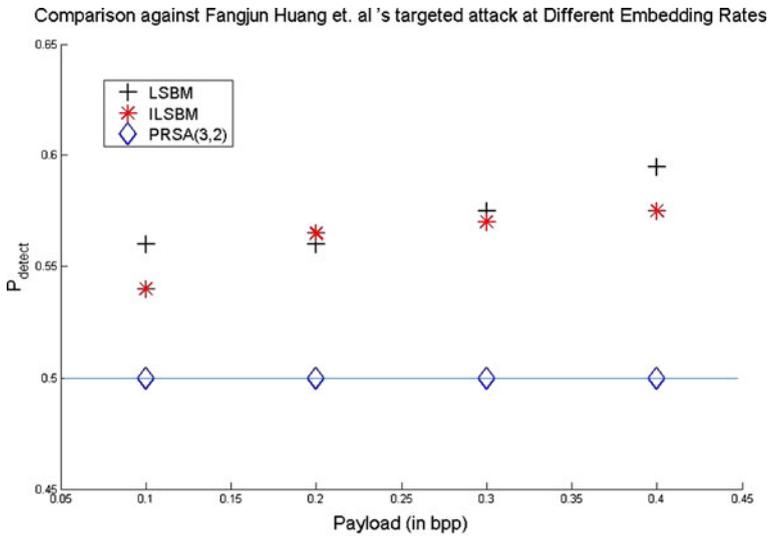


(b) Against Adjacency HCF Attack

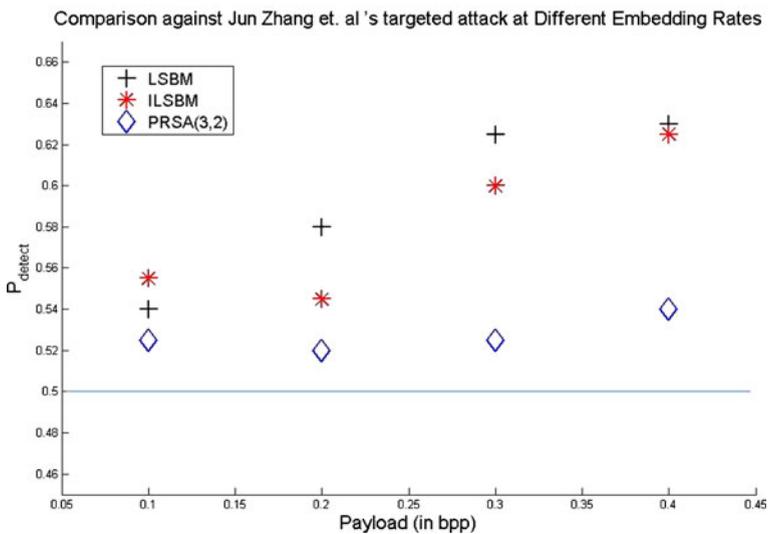
**Fig. 5** Against HCF calibration and HCF adjacency attacks. **a** Against HCF calibration attack. **b** Against adjacency HCF attack

not shown because both the schemes have produced almost identical results against histogram based attacks. Since  $AROC$  and  $P_{\text{detect}}$  are similar metrics to denote the detection accuracy, comparative plot for  $P_{\text{detect}}$  is produced here only. A summary of observation and some intuitive explanation of the results against different attacks are provided below:

Experimental results from Fig. 5a shows that the proposed  $PRSA(3, 2)$  scheme can not be detected by HCF COM with Calibration by Down sampling attack [8].

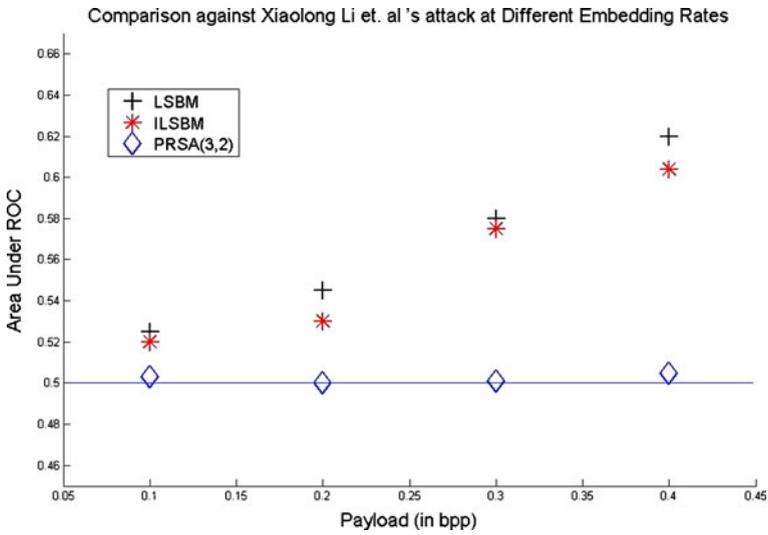


(a) Against Fangjun Huang et al.'s Attack



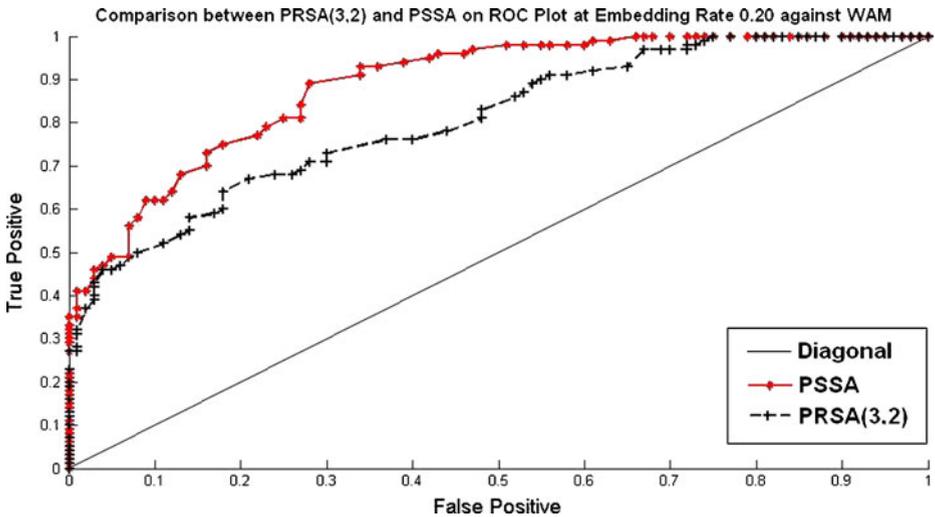
(b) Against Jun Zhang et al.'s Attack

**Fig. 6** Against Fangjun Huang et al.'s and Jun Zhang et al.'s attack. **a** Against Fangjun Huang et al.'s attack. **b** Against Jun Zhang et al.'s attack



**Fig. 7** Against Xiaolong Li et al.'s attack

Since, it is image histogram based attack and proposed scheme completely restores the cover image histogram, this attack can not detect proposed scheme even at very large embedding rate. Moreover, in spite of poor detection, it is observed that the detection accuracy ( $P_{detect}$ ) is increasing with increase of payload for *LSBM* and *ILSBM*. It implies that this attack can detect *LSBM* and *ILSBM* scheme with a higher payload.



**Fig. 8** Comparison of ROC plot between proposed algorithm and PSSA against WAM based steganalysis

**Table 9** Comparison of detection accuracy ( $P_{\text{detect}}$ ) between PSSA and PRSA(3,2) against WAM at different embedding rates

| Embedding rate | PSSA  | PRSA(3,2) |
|----------------|-------|-----------|
| 0.05           | 0.565 | 0.540     |
| 0.10           | 0.625 | 0.575     |
| 0.175          | 0.678 | 0.655     |
| 0.20           | 0.697 | 0.673     |

From Fig. 5b, it is observed that proposed *PRSA* algorithm clearly outperforms *LSBM* and *ILSBM* against HCF COM of Adjacency Histogram based attack [8] which considers second order image statistics.

From Fig. 6a, it can be observed that the *PRSA* scheme also performs better than the *LSBM* and the *ILSBM* against Fangjun Huang et al.'s attack [7].

Jun Zhang et al.'s detector [25] performs well in case of *never compressed images*. It outperforms a blind attack based on wavelet moment steganalysis (WAM), when used for *never compressed images*. But this attack can not be successful against the proposed *PRSA* scheme since it depends only on image histogram for detection. Results from Fig. 6b verify this fact that the *PRSA* scheme performs much better than the *LSBM* and the *ILSBM* against Jun Zhang et al.'s attack.

From Fig. 7, it can be observed that the *PRSA* scheme also performs better than the *LSBM* and the *ILSBM* against Xiaolong Li et al.'s attack [9].

It is worth mentioning here that most spatial domain steganographic schemes including LSB Matching, Improved LSB Matching, PSSA, proposed *PRSA* scheme etc. assume that the communication channel is lossless and are susceptible to JPEG compression like attacks. It can be considered as one demerit of all these kind of spatial domain steganographic schemes.

#### 5.4 Security against blind steganalysis:

It is discussed in Section 4.2 that the *PSSA* scheme can be representative for any statistical restoration based techniques for noise addition. The proposed *PRSA* scheme is compared only with *PSSA* scheme to compare its performance against additive noise based blind attacks. For this purpose we have chosen the WAM steganalysis [5]. Images from the *Never Compressed Image Database* (NCID) [14] are used as test images. A typical ROC plot for embedding rates is given in Fig. 8. The detection performances of PSSA and PRSA(3,2) at different embedding rates are tabulated in Table 9. From Fig. 8 and Table 9, it is observed that the proposed *PRSA*(3, 2) is relatively less detectable than the *PSSA* scheme against WAM based blind attacks. These results conform with the theoretical analysis presented in Sections 4.2 and 5.1 showing that the proposed *PRSA* scheme adds less additive noise than the *PSSA* [20] scheme and thus less detectable against additive noise based blind attacks such as WAM [5].

## 6 Conclusion

In this paper, a pixel rearrangement based adaptive embedding scheme has been proposed which provides better performance than existing statistical restoration based schemes (like [20]). The main contribution of this paper is to reduce the

embedding noise for statistical restoration based scheme specially in spatial domain. An empirical noise analysis is given in favor of this claim. It is experimentally shown that due to this improvement, resulting PSNR between cover and stego images for the proposed *PRSA* scheme is reduced than the existing schemes. Moreover, the proposed *PRSA* scheme is less detectable against additive noise based blind attacks.

**Acknowledgements** The authors acknowledge the help received from the anonymous reviewers for the improvisation of the paper from its previous version.

## References

1. Chen CH, Shi YQ, Chen W, Xuan GR (2006) Statistical moments based universal steganalysis using JPEG 2-D array and 2-D characteristic function. In: Proceedings of IEEE international conference on image processing, pp 105–108
2. Dumitrescu S, Wu X, Wang Z (2002) Detection of LSB steganography via sample pair analysis. In: Proc. information hiding workshop, LNCS, vol 2578. Springer, pp 355–372
3. Eggers JJ, Bauml R, Girod B (2002) A communications approach to image steganography. In: Proc. SPIE security and watermarking of multimedia contents IV, vol 4675, pp 26–37
4. Fridrich J, Goljan M, Dui R (2001) Reliable detection of LSB steganography in color and grayscale images. In: Proc. ACM workshop on multimedia and security, Ottawa, CA, 5 Oct 2001, pp 27–30
5. Goljan M, Fridrich J, Holotyak T (2006) New blind steganalysis and its implications. In: Proceedings of SPIE for security, steganography, and watermarking of multimedia contents VIII, vol 6072, pp.1–13
6. Harmsen J, Pearlman W (2003) Steganalysis of additive noise modelable information hiding. In: Proc. security and watermarking of multimedia contents V, vol 5020, pp 131–142
7. Huang F, Li B, Huang J (2007) Attack LSB matching steganography by counting alteration rate of the number of neighbourhood gray levels. In: Proc. IEEE international conference on image processing, ICIP 2007, vol 1, pp 1401–1404
8. Ker AD (2005) Steganalysis of LSB matching in grayscale images. *IEEE Signal Process Lett* 12(6):441–444
9. Li X, Zeng T, Yang B (2008) A further study on steganalysis of LSB matching by calibration. In: Proc. IEEE international conference on image processing, ICIP 2008, pp 2072–2075
10. Mansouri A, Aznavah AM, Torkamani-Azar F, Kurugollu F (2010) A low complexity video watermarking in H.264 compressed domain. *IEEE T Inf Foren Sec* 5(4):649–657
11. Mielikainen J (2006) LSB matching revisited. *IEEE Signal Process Lett* 13(5):285–287
12. Provos N (2001) Defending against statistical steganalysis. In: Proc. 10th USENIX security symposium, vol 10. Washington DC, August 13–17 2001
13. Sallee P (2003) Model-based steganography. In: Proc. 2nd international workshop on digital watermarking, Seoul, Korea, 20–20 Oct 2003, pp 154–167
14. Schaefer G, Stich M (2004) UCID—an uncompressed colour image database. In: Proc. SPIE, storage and retrieval methods and applications for multimedia, vol 5307, pp 472–480
15. Shi YQ, Xuan GR, Yang CY, Gao JJ, Zhang ZP, Chai PQ, Zou DK, Chen CH, Chen W (2005) Effective steganalysis based on statistical moments of wavelet characteristic function. In: Proceedings of IEEE international conference on information technology: coding and computing, pp 768–773
16. Shi YQ, Xuan GR, Zou DK (2005) Image steganalysis based on moments of characteristic functions using wavelet decomposition prediction-error image and neural network. In: Proceedings of IEEE international conference on multimedia and expo, pp 269–272
17. Solanki K, Sullivan K, Madhow U, Manjunath BS, Chandrasekaran S (2005) Statistical restoration for robust and secure steganography. In: Proc. IEEE int. conf. on image processing, Genova, Italy, vol 2, 11–14 Sep 2005, pp 1118–1121
18. Solanki K, Sullivan K, Madhow U, Manjunath BS, Chandrasekaran S (2006) Probably secure steganography: achieving zero K-L divergence using statistical restoration. In: Proc. IEEE int. conf. on image processing, Atlanta, GA, USA, 8–11 Oct 2006, pp 125–128

19. Solanki K, Sarkar A, Manjunath BS (2007) YASS: yet another steganographic scheme that resists blind steganalysis. In: Proc. 9th int. workshop on information hiding, Saint Malo, Brittany, France, pp 16–31
20. Sur A, Goel P, Mukhopadhyay J (2008) A novel steganographic algorithm resisting targeted steganalytic attacks on LSB matching. In: International workshop on digital watermarking (IWDW 2008), 10–12 November 2008, Busan, Korea (presented)
21. Sur A, Goel P, Mukhopadhyay J (2009) A new statistical restoration method for spatial domain images. In: Proc. third international conference on pattern recognition and machine intelligence (PReMI'09), Delhi, India. Lecture notes in computer science, vol 5909/2009. Springer, Berlin/Heidelberg, pp 297–302
22. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: From error measurement to structural similarity. *IEEE Trans Image Process* 13(1):600–612
23. Xuan GR, Gao JJ, Shi YQ, Zou DK (2005) Image steganalysis based on statistical moments of wavelet subband histograms in DFT domain. In: Proceedings of IEEE international workshop on multimedia signal processing, pp 1–4
24. Xuan GR, Shi YQ, Gao JJ, Zou DK, Yang CY, Zhang ZP, Chai PQ, Chen CH, Chen W (2005) Steganalysis based on multiple features formed by statistical moments of wavelet characteristic functions. In: Proc. of seventh international information hiding workshop. Lecture notes in computer science, vol 3727. Springer, Berlin, pp 262–277
25. Zhang J, Cox IJ, Doerr G (2007) Steganalysis for LSB matching in images with high-frequency noise. In: Proc. IEEE 9th workshop on multimedia signal processing MMSP 2007, pp 385–388



**Arijit Sur** received his Ph.D. degree in Computer Science and Engineering from Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur. He has received his M.Sc. in Computer and Information Science and M. Tech in Computer Science and Engineering both from Department of Computer Science and Engineering, University of Calcutta. He is currently working as an Assistant Professor at Department of Computer Science and Engineering, Indian Institute of Technology Guwahati. He is a recipient of Infosys Scholarship during his Ph.D. tenure at IIT Kharagpur. He got Microsoft Outstanding Young Faculty Programme Award at Dept. of CSE, IIT Guwahati. His current research interest is Multimedia Security such as Image and Video Watermarking, Steganography & Steganalysis, Reversible Data Hiding and Network Security.



**Vignesh Ramanathan** is currently pursuing his B. Tech Dual Degree in Electrical Engineering at Department of Electrical Engineering, Indian Institute of Technology Kharagpur. His current research interests are image processing, image retrieval etc. He has visited Media Communication Lab, Dept. of Electrical Engineering, University of Southern California, LA, CA as a summer intern.



**Jayanta Mukherjee** received his B.Tech., M.Tech., and Ph.D. degrees in Electronics and Electrical Communication Engineering from the Indian Institute of Technology (IIT), Kharagpur in 1985, 1987, and 1990, respectively. He joined the faculty of the Department of Electronics and Electrical Communication Engineering at IIT, Kharagpur in 1990 and later transferred to the Department of Computer Science and Engineering where he is presently a Professor. He served as the head of the Computer and Informatics Center at IIT, Kharagpur from September 2004 to July 2007 and is presently serving as the head of the Department of Computer Science and Engineering and the School of Information and Technology. He was a Humboldt Research Fellow at the Technical University of Munich in Germany for one year in 2002. He also has held short term visiting positions at the University of California, Santa Barbara, University of Southern California, and the National University of Singapore. His research interests are in image processing, pattern recognition, computer graphics, multimedia systems and medical informatics. He has published over 150 papers in journals and conference proceedings in these areas. He received the Young Scientist Award from the Indian National Science Academy in 1992. Dr. Mukherjee is a Senior Member of the IEEE. He is a fellow of the Indian National Academy of Engineering (INAE).