◆◆ Scientific
◆◆ Research

Reversible Digital Image Watermarking Scheme Using Bit Replacement and Majority Algorithm Technique

Koushik Pal[1], Goutam Ghosh[1], Mahua Bhattacharya[2*]

[1]Institute of Radio Physics and Electronics, University of Calcutta, Kolkata, India; [2]Indian Institute of Information Technology and Management, Gwalior, India.
Email: [*]bmahua@hotmail.com

ABSTRACT

The current paper presents a new digital watermarking method through bit replacement technology, which stores multiple copies of the same data that is to be hidden in a scrambled form in the cover image. In this paper an indigenous approach is described for recovering the data from the damaged copies of the data under attack by applying a majority algorithm to find the closest twin of the embedded information. A new type of non-oblivious detection method is also proposed. The improvement in performance is supported through experimental results which show much enhancement in the visual and statistical invisibility of hidden data.

1. Introduction

Recent history has witnessed the rapid development in information technologies that has given an extended and easy access to digital information. Along with several developments it leads to the problem of illegal copying and redistribution of digital media. As a result, the integrity and confidentiality of the digital information has come under immense threat. Digital watermarking, an emerging technology, came in order to solve the problems [1]. Digital watermarking is a technique which allows an individual to add hidden copyright notices or other verification messages or even classified information into digital media [2,3].

Watermarks can either be visible or invisible. Here in this paper we utilize the invisible watermarking technique. This is used in public information settings such as digital image libraries, museums, and art galleries and also in defense communication where data security is of prime importance [4]. Watermark embedding utilizes two kinds of methods, one in the spatial domain and the other in the transform domain. In the spatial domain the watermark is directly embedded into the image pixels whereas in the frequency domain the image is decomposed into blocks and then mapped into the transform domain [5].

Watermarking is basically a process of hiding information in an image known as cover image. According to the behavior it is of two types: robust and fragile. Copyright protection is achieved by robust watermarking [6] while image authentication is usually achieved by fragile watermarking techniques [7-9]. Robust watermarking technique ensures the quality of the hidden information by several protection algorithms. In the fragile watermarking scheme if any alteration of the message is found then the message breaks up by itself and can be easily detected as tampered by the provider of the watermark [1]. Invertible watermarking is a new process which enables the exact recovery of the original image upon extraction of the embedded information.

In the present paper the work implements both authentication and confidentiality in a reversible [10] manner without affecting the image in any way. Security of images imposes three mandatory characteristics: confidentiality, reliability and availability [11]. *Confidentiality* means that only the entitled persons have access to the images. *Reliability* has two aspects: 1) Integrity—means the image has not been modified by a non-authorized person [12]; 2) Authentication corroborates that the image belongs indeed to the correct person and is issued from an authorized source. *Availability* is the capacity that an image is available to the entitled persons in the normal conditions of access and exercise [13].

1.1. LSB Watermarking and Its Limitation

The most straight-forward method of watermark embedding would be to embed the watermark into the least-

---

significant-bits (LSB) of the cover object [14]. Given the extraordinarily high channel capacity of using the entire cover for transmission in this method, a smaller object may be embedded multiple times [8]. Even if most of these are lost due to attacks, a single surviving watermark would be considered as a success. LSB substitution, however, despite its simplicity brings a lot of drawbacks. Although it may survive from all these transformations such as cropping, addition of noise or lossy compression etc. A better attack would be to simply set the LSB bits of each pixel to defeat the watermark with negligible impact on the cover object. Furthermore, once the algorithm is discovered, the embedded watermark could be easily modified by an intermediate party. An improvement on basic LSB substitution would be to use a pseudo-random number generator to determine the pixels to be used for embedding based on a given "seed" or key [1]. The algorithm however would still be vulnerable to replacing the LSB's with a constant. Even locations that were not used for the watermarking bits, the impact of the substitution on the cover image would be negligible. LSB modification proves to be a simple and fairly powerful tool for steganography [15], however, it lacks the basic robustness that watermarking applications require.

1.2. Attack and Distortion

In practice, a watermarked image may be altered either on purpose or accidentally. The watermarking system should be robust enough to detect and extract the watermark [16]. Different types of alterations or attacks can be done to degrade the image quality by adding distortions.

The distortions are limited to those factors which do not produce excessive degradations; otherwise the transformed object would be unusable. These distortions also introduce degradation on the performance of the watermark extraction algorithm [17,18]. Methods or a combination of methods, considered unintentional are used intentionally as an attack on a watermarked image in order to render the watermark undetectable.

Compression is a common attack, as data transferred via network is often compressed using JPEG. High quality images are often converted to JPEG to reduce their size. Another method is deletion or shuffling of blocks. In images rows or columns of pixels may be deleted or shuffled without a noticeable degradation in image quality. These may render an existing watermark undetectable.

Salt and pepper noise is another type of attack that replaces the intensity levels of some of the pixels of an image resulting in loss of information from those pixels. Some of the best known attacks are mentioned here; they may be intentional or unintentional, depending on the application.

In this paper we have taken two very popular attacks known as Salt and pepper noise and image compression.

2. Proposed Watermarking Technique for Data Authentication

Our proposed methodology for data hiding does not follow the conventional LSB technique because of the inherent limitations mentioned earlier. A new digital watermarking scheme described here, uses several bits of the cover image starting from the lower order to the higher order to hide the information logo. Several sets of the same data forming the information logo are hidden into the cover image. Thus, even if some of the information is lost due to attack, we can still collect the remaining information from the cover image and reconstruct the hidden information resembling the original one [19].

The detail algorithm for both the embedding scheme and the recovery scheme is given bellow.

2.1. Embedding the Digital Watermark

Step 1: Two images are taken as input: First of all, the cover image and the message or information logo are taken as inputs. The cover image is taken to be a gray scale image. The logo or information is a binary image, basically a sequence of 0's and 1's.

Step 2: The Size of the images is extracted: Next to make the program compatible to run for any size of the cover image and information logo keeping in mind the data carrying capacity of the cover image the dimensions of the respective images are extracted.

Mc=size(cover_image,1);Nc=size(cover_image,2);
Mm=size(logo,1);Nm=size(logo,2);

Step 3: Normalizing and reshaping the logo: After normalizing the information logo it is reshaped in one dimension.

logo_nor=logo/.256;

Step 4: Transforming the cover image into wavelet domain using dwt: The cover image is transformed to the wavelet domain using discrete wavelet transform. We use Haar transform to do the DWT. Here the 1st level DWT was used to obtain more capacity for hiding the information. The cover image is decomposed into 4 sub-domains as HH, HL, LH and LL according to the different frequencies of the cover image.

[HHi,HLi,LHi,LLi] = dwt2(cover_image,'haar');

Step 5: Calculating the length of the transformed cover image and 1d logo:

len_logo=length(logo_res);
*len_cover=Mc*Nc;*

Step 6: Calculating the size of each sub domain decomposed cover image and reshaping them in to 1d:

cover_HL_size_m=size(HLi,1);
cover_HL_size_n=size(HLi,2);
*HL1=reshape(HLi,1,cover_HL_size_m*cover_HL_size_n);*

Step 7: Determining the maximum coefficient value

of each of the 4 sub domains

max_HL1=max(HL1);

Step 8: Finding the position to hide the information logo into the transformed logo: The position for hiding the binary logo in each sub domain must be in between zero and the maximum coefficient value of that sub domain.

c=1;
* for i = 1: 1 : cover_HL_size_m*cover_HL_size_n*
*   if(HL1(i)<(max_ HL1) && HL1(i)>(0))*
*   pos_ HL1(c)=i;*
*   c=c+1;*
*   end*
* end*

Step 9: Hiding a number of sets of the same information logo in hl and lh domain: More than one set of the same information is being hidden in HL and LH band or domain for easier and good quality recovery. The hiding process in each of these domains follows a specific formula. The formula is that the black dots in each set of 1d information logo is hidden in a position of information logo from where a constant value is subtracted.

a=15
for i = 1: 1 : length_of_HL
*   if logo_res(i)==0*

w_HL1(pos_HL1_set_1(i))=HL1(pos_HL1_set_1(i))-a;
*   end*
* end*

Step10: Reshaping the decomposed image back to its normal dimension:

w_cH1_reshape=reshape(w_cH1,cover_cH_size_m,cover_cH_size_n);

Step11: Writing the watermarked image to a file and displaying it.

2.2. Recovery of the Embedded Watermark

We have assumed that the cover image hiding the watermark is available at the receiving end. So again, in the process of recovery we first take the original image used to hide the information. Along with it we also send the receiver of the message, 3 keys which essentially act as private keys. These keys are required to decrypt and to the extract the encrypted or embedded messages.

Step 1: The watermarked and original images are taken as inputs.

Step2: Finding the 1st level decomposition of both the inputs using DWT:

[HHw,HLw,LHw,LLw]=dwt2(watermarked_image,'haar');

[HHi,HLi,LHi,LLi] = dwt2(orig_image,'haar');

Step 3: Finding the size of each sub domain of both the decomposed input images:

orig_HL_size_m=size(HLi,1);
orig_HL_size_n=size(HLi,2);

watermarked_HL_size_m=size(HLw,1);
watermarked_HL_size_n=size(HLw,2);

Step 4: Reshaping each of the decomposition of both watermarked and original cover images into 1D:

*HLo=reshape(HLi,1,orig_HL_size_m*orig_HL_size_n);*

*HL1=reshape(HLw,1,watermarked_HL_size_m*watermarked_HL_size_n);*

Step 5: Taking the two input keys equal to the dimension of the logo to find the size of the 4 decompositions of the logo:

key_M=input('Enter the no of rows KEY 1 :');
key_N=input('Enter the no of cols KEY 2:');
logo_HL_size_m=key_M/2;
logo_HL_size_n=key_N/2;

Step 6: Determining the maximum coefficient values of the original cover image:

max_cHo=max(cHo);

Step 7: Finding the positions used to hide the logo for each decomposition:

c=1;
* for i = 1: 1 : orig_HL_size_m*orig_cA_size_n*
*   if(HLo(i)<(max_HLo) && HLo(i)>(0))*
*   pos_HLo(c)=i;*
*   c=c+1;*
*   end*
* end*

Step 8: Extracting the positional sets for different sets of the logo from each decomposition:

c=1;
for i = 1 : 1 : len_set_HLo
*   pos_HLo_set_1(i)=pos_HLo(c);*
*   c=c+1;*

Step 9: Recovery of the different sets of logo from each of the sub bands and construction of the final logo from the different recovered sets using majority algorithm:

*for i = 1: 1 : key_M*key_N*

if(rnd_HL1(pos_HLo_set_3(i))-rnd_HLo(pos_HLo_set_3(i)))==0
*   rec_wmark_V_set_3(i)=1;*
*   else*
*   rec_wmark_V_set_3(i)=0;*
*   end*
end

Step 10: After reshaping, displaying each of the recovered sets of logo and the final constructed logo:

res_rec_wmark_H_set_3=reshape(rec_wmark_H_set_3,key_M,key_N);
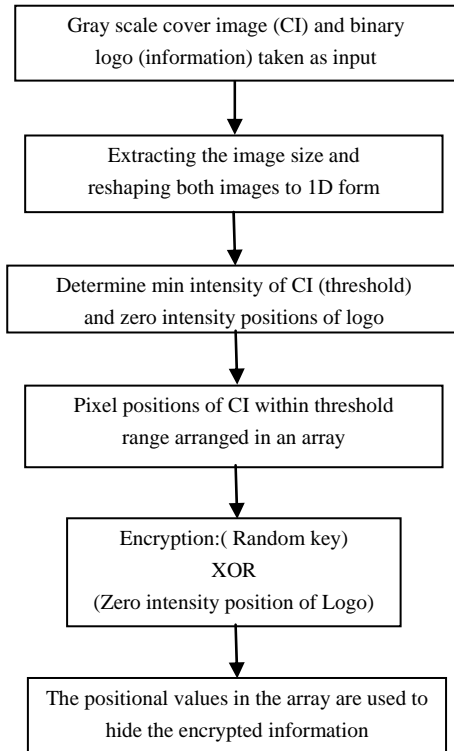*   if rec_wmark_V_set_3(i)==1*
*   val_1=val_1+1;*
*   end*
rec_wmark_H_FINAL(i)=val;

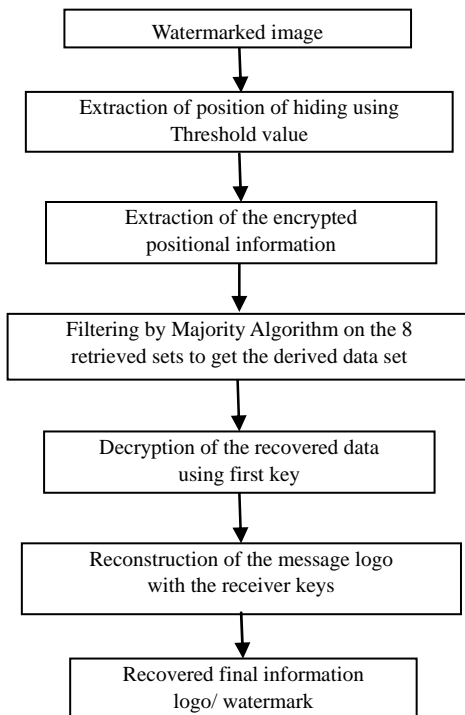*res_rec_wmark_H_FINAL=reshape(rec_wmark_H_FINAL,key_M,key_N);*

2.3. Design Flow of the Proposed Scheme

Embedding Technique:

```
┌─────────────────────────────────────┐
│ Gray scale cover image (CI) and binary│
│    logo (information) taken as input  │
└─────────────────────────────────────┘
                  │
                  ▼
┌─────────────────────────────────────┐
│      Extracting the image size and    │
│      reshaping both images to 1D form │
└─────────────────────────────────────┘
                  │
                  ▼
┌─────────────────────────────────────┐
│  Determine min intensity of CI (threshold)│
│   and zero intensity positions of logo│
└─────────────────────────────────────┘
                  │
                  ▼
┌─────────────────────────────────────┐
│    Pixel positions of CI within threshold│
│        range arranged in an array     │
└─────────────────────────────────────┘
                  │
                  ▼
┌─────────────────────────────────────┐
│         Encryption:( Random key)      │
│                  XOR                  │
│       (Zero intensity position of Logo)│
└─────────────────────────────────────┘
                  │
                  ▼
┌─────────────────────────────────────┐
│   The positional values in the array are used to│
│         hide the encrypted information│
└─────────────────────────────────────┘
```

Decoding Technique:

```
┌─────────────────────────────────────┐
│           Watermarked image           │
└─────────────────────────────────────┘
                  │
                  ▼
┌─────────────────────────────────────┐
│   Extraction of position of hiding using│
│             Threshold value           │
└─────────────────────────────────────┘
                  │
                  ▼
┌─────────────────────────────────────┐
│       Extraction of the encrypted     │
│          positional information       │
└─────────────────────────────────────┘
                  │
                  ▼
┌─────────────────────────────────────┐
│  Filtering by Majority Algorithm on the 8│
│     retrieved sets to get the derived data set│
└─────────────────────────────────────┘
                  │
                  ▼
┌─────────────────────────────────────┐
│      Decryption of the recovered data │
│             using first key           │
└─────────────────────────────────────┘
                  │
                  ▼
┌─────────────────────────────────────┐
│     Reconstruction of the message logo│
│          with the receiver keys       │
└─────────────────────────────────────┘
                  │
                  ▼
┌─────────────────────────────────────┐
│       Recovered final information     │
│            logo/ watermark            │
└─────────────────────────────────────┘
```

3. Image Quality Metrics

To measure the amount of visual quality degradation between original and watermarked images, different types of image quality metrics are used. In the present work we have used *peak signal-to-noise ratio* (*PSNR*) and structural similarity index measure (*SSIM*).

3.1. Peak Signal-to-Noise Ratio (*PSNR*)

It is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. *PSNR* is usually expressed in terms of dB for a wide range of signals The *PSNR* is most commonly used as a measure of quality of reconstruction for lossy compression. The cover image in this case is the original data, and the information logo is the error introduced by watermarking. When comparing the deformed image with the original one an *approximation* to human perception of reconstruction quality is made. Therefore in some cases one reconstruction may appear to be closer to the original than another, even though it has a lower *PSNR*. So a higher *PSNR* would normally indicate that the reconstruction is of higher quality.

It is most easily defined via the mean square error (*MSE*) which for two $m \times n$ monochrome images I and $K,$ where one of the images is considered a noisy approximation of the other, is defined as:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left[I(i,j) - K(i,j) \right]^2$$

The *PSNR* is defined as:

$$PSNR = 10 \log_{10} \left(\frac{MAX_1^2}{MSE} \right)$$

$$= 20 \log_{10} \left(\frac{MAX_1}{\sqrt{MSE}} \right)$$

here, MAX_1 is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255.

3.2. Structural Similarity Index Measure (*SSIM*)

It is a method for measuring the similarity between two images. The *SSIM* index is a full reference metric, where the measure of the image quality is based on an initial distortion-free image as reference. *SSIM* is designed to improve on traditional methods like *PSNR* and *MSE*, which have proved to be inconsistent with human eye perception. The resultant *SSIM* index is a decimal value between –1 and 1. The value 1 is only reachable in the case of two identical sets of data. The *SSIM* metric is calculated on various windows of an image. The measure between two windows x and y of common size $N \times N$ is:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

where μ_x is the average of x; μ_y the average of y; σ_x^2 the variance of x; σ_y^2 the variance of y; σ_{xy} the covariance of x and y; $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ are two variables to stabilize the division with weak denominator; L the dynamic range of the pixel-values (typically this is $2^{\text{\# bits per pixel}} - 1$); $k_1 = 0.01$ and $k_2 = 0.03$ by default.

4. Results and Discussions

In this section several experimental results are given to show the outcome of the proposed watermarking technique.

In Section 4.1 three sets of cover image along with three information logos are taken as input. The watermarked image is shown after embedding. The computed value of the quality metrics are also given to find the image quality.

In Section 4.2, the watermarked images and the recovered information logos are given.

In Section 4.3, the outcome for the same recovery technique is shown but under two attacks known as salt and pepper noise and image compression. For salt and pepper noise the percentage is varied up to 40% and compression up to 5%. The required noisy watermarked images and the recovered logo from those images are presented.

In Section 4.4, eight different sets of the recovered logo and the final constructed logo using majority algorithm are given.

4.1. Embedding of Watermark into Cover Image and Quality Metrics

It can be observed from **Table 1** that the results obtained from the quality metrics are very satisfactory and hence we can conclude from the obtained data that the watermarked image is not very much different from the original cover image that is being used. Also, the difference between the watermarked image and the original appears almost the same to the human eye.

Higher *PSNR* value indicates good quality of picture. After embedding the information logo in the cover image for *Lena*, *To*wer and *Fruit* images we find that the *PSNR* value is quite high.

Similarly *SSIM* is another measuring metric used for finding the similarity between the two images. Here we observe that after embedding the information logo the similarity between the cover image and watermarked image is 0.98 which describes a good *structural similarity* between these two images.

4.2. Recovery of Watermark from Watermarked Image without Any Attack

Here the hidden watermarked image *i.e.* the information logo is successfully recovered from the un-attacked

Table 1. Watermark embedding and image quality metrics.

Cover Image (dim-256 × 256)	Message Image (dim-16 × 16)	Watermarked Image (dim-256 × 256)	PSNR in dB	SSIM
Lena	S logo	Watermarked Lena	42.343	0.9889
Tower	K logo	Watermarked Tower	41.806	0.9781
Fruit	Max Payne logo	Watermarked Fruit	41.506	0.9853

watermarked image which is given in **Table 2**. Primarily, we have considered the communication is ideal and hence no external interference or attack has been included.

In practice, however, we have to consider noise, which is dealt with in the next section.

4.3. Recovery of Watermark from Watermarked Image under Attacks

In watermarking terminology, an attack is any processing that may impair detection of the watermark or communication of the information conveyed by the watermark. The processed watermarked data is then called attacked data. There are two kinds of watermark attacks: Non-intentional attacks, such as compression of a legally obtained, watermarked image or video file, and intentional attacks, such as an attempt by a multimedia pirate to destroy the embedded information and prevent tracing of illegal copies of watermarked digital video.

The present work describes the following two types of attack:

(1) *Salt and pepper noise*.
(2) *JPEG Compression*.
(1) *Salt and pepper noise*:

In this section we have demonstrated the proposed watermarking technique after using the salt and pepper noise to corrupt the watermarked images up to 40%.

From the above set of results in **Table 3**, it is clear that that the proposed algorithm can withstand 40% salt and

Table 2. Recovered watermark or hidden message from un-attacked watermarked image.

Watermarked Image (dimension 256 × 256)	Recovered Message Image (dimension 16 × 16)
Lena	S logo
Tower	K logo
Fruit	Max Payne logo

Table 3. Recovered watermark or hidden message from salt and pepper noise attacked watermarked image.

Watermarked Image (dimension 256 × 256)	Attacked Image (Salt and Pepper Noise)	Recovered Message Image (dimension 16 × 16)
Lena	10%	S logo
Lena	20%	S logo
Lena	30%	S logo
Lena	40%	S logo

pepper attack with ease and the information logo that is derived from the watermarked image closely resembles the information logo that was embedded into the image. Hence we can say that the proposed algorithm efficiently handles salt and pepper noise.

Similarly in the next **Table 4**, the strength of the proposed algorithm is demonstrated against the salt and pepper attack with a different set of data.

(2) *JPEG Compression*:

The performance of the proposed algorithm is also demonstrated against JPEG compression attack in **Table 5**. This algorithm also demonstrates its strength against compression attack as well.

4.4. Construction of Different Recovered Logo Using Majority Algorithm Technique

Tables 6 and **7** describe the outcome of majority algorithm technique. Here final constructed S logo and K logo from 8 different recovered sets have been shown. It is clearly visible that the proposed majority algorithm technique is strong enough to construct the information logo from some distorted sets of recovered logo.

　　　JILSA

Table 4. Recovered watermark from salt and pepper noise attacked watermarked image.
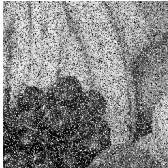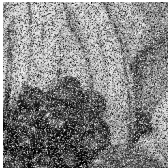
Watermarked Image	Attacked Image (Salt and Pepper Noise)	Recovered Message Image
 Fruit	 10%	 Max Payne
 Fruit	 20%	 Max Payne
 Fruit	 30%	 Max Payne
 Fruit	 40%	 Max Payne

Table 5. Recovered watermark from JPEG compressed watermarked image.

Watermarked Image	Attacked Image (JPEG Compression)	Recovered Message Image
 Fruit	 Quality = 99%	 Max Payne
 Tower	 Quality = 98%	 K logo
 Lena	 Quality = 95%	 S logo

Table 6. Derived S logo from 8 sets of recovered noisy logo using majority algorithm.

Recovered S logo from 1st set	Recovered S logo from 2nd set	Recovered S logo from 3rd set	Recovered S logo from 4th set	
Recovered S logo from 5th set	Recovered S logo from 6th set	Recovered S logo from 7th set	Recovered S logo from 8th set	Derived S logo from these 8 set using Majority Algorithm

Table 7. Derived K logo from 8 sets of recovered noisy logo using majority algorithm.

Recovered K logo from 1st set	Recovered K logo from 2nd set	Recovered K logo from 3rd set	Recovered K logo from 4th set	
Recovered K logo from 5th set	Recovered K logo from 6th set	Recovered K logo from 7th set	Recovered K logo from 8th set	Derived K logo from these 8 set using Majority Algorithm

SSIM is used for finding the similarity between the two images. **Table 8** describes the quality of the recovered logo. The similarity between the original logo and the recovered logo from the watermarked image is measured using *SSIM*. The following results describe that the proposed algorithm is quite efficient for salt and pepper noise up to 40% and JPEG compression resulting in image quality distortion = 95%.

5. Conclusions

In present paper the proposed algorithm for digital watermarking aims at obtaining a solution to the several problems of digital communication and also for data hiding. It is seen that the proposed algorithm is robust against compression and "salt and pepper" noise attacks where a private key is required for the recovery of the hidden information and which enhances security to the algorithm. Since digital watermarking has many applications in the digital world today it can be thought of as a digital communication scheme where an auxiliary message is embedded in digital multimedia signals and is available wherever the latter signals move.

The results obtained show satisfactory statistics of the performance of the proposed algorithm. The obtained *PSNR* and *SSIM* values support the quality of the encryption method. It is also seen that the embedded informa-

Table 8. SSIM values for different sets of recovered watermark or information logo under salt and pepper noise and JPEG compression attack.

Used Logo (dimension 16×16)	Types of Attack	Amount of Distortion	SSIM
S Logo	Salt and Pepper Noise	20%	0.9554
		30%	0.9032
		40%	0.7954
S Logo	JPEG Compression	Quality = 99%	0.9687
		Quality = 98%	0.8654
		Quality = 95%	0.7496

mation is successfully recovered from the watermarked image by using the majority algorithm technique. We can conclude by stating that the proposed algorithm provides a method for secure data hiding.

6. Acknowledgements

REFERENCES

[1] P. W. Wong, "A Public Key Watermark for Image Verification and Authentication," *Proceedings of the IEEE International Conference on Image Processing*, Chicago, 4-7 October 1998, pp. 455-459.

[2] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," Prentice Hall, Upper Saddle River, 2002.

[3] I. J. Cox, M. Miller and J. Bloom, "Digital Watermarking," Morgan Kaufmann Publishers, Burlington, 2002.

[4] M. Chandra, S. Pandey and R. Chaudhary, "Digital Watermarking Technique for Protecting Digital Images," 3rd *IEEE International Conference* on *Computer Science and Information Technology* (*ICCSIT*)," Chengdu, 9-11 July 2010, pp. 226-233. doi:10.1109/ICCSIT.2010.5565177

[5] M. Kutter and F. A. P. Petitcolas, "A Fair Benchmark for Image Watermarking Systems," *Proceedings of SPIE Security and Watermarking of Multimedia Contents*, San Jose, January 1999, pp. 226-239.

[6] R. M. Prasad and S. Koliwad, "A Robust Wavelet-Based Watermarking Scheme for Copyright Protection of Digital Images," *International Conference on Computing Communication and Networking Technologies* (*ICCCNT*), Karur, 29-31 July 2010, pp. 1-9.

[7] M. L. Miller, I. J. Cox, J. M. G. Linnartz and T. Kalker,

"A Review of Watermarking Principles and Practices," In: K. K. Parhi and T. Nishitani, Eds., *Digital Signal Processing for Multimedia Syatems*, Marcel Dekker Inc., New York, 1999, pp. 461-485.

[8] D. Osborne, D. Abbott, M. Sorell and D. Rogers, "Multiple Embedding Using Robust Watermarks for Wireless Medical Images," *IEEE Symposium on Electronics and Telecommunications*, Timisoara, October 2004, pp. 245-250.

[9] N. V. Dharwadkar, B. B. Amberker, Supriya and P. B. Panchannavar, "Reversible Fragile Medical Image Watermarking with Zero Distortion," *International Conference on Computer and Communication Technology* (*ICCCT*), 17-19 September 2010.

[10] D. Coltuc, "Low Distortion Transform for Reversible Watermarking," *IEEE Transactions on Image Processing*, Vol. 21, No. 1, 2012, pp. 412-417.

[11] J. Fridrich, *et al.*, "Lossless Data Embedding for All Image Formats," *Proceedings of SPIE Photonics West*, *Security and Watermarking of Multimedia Contents*, San Jose, January 2002, pp. 572-583.

[12] H. Huang, G. Coatrieux, H. Z. Shu, L. M. Luo and C. Roux, "Medical Image Integrity Control and Forensics Based on Watermarking—Approximating Local Modifications and Identifying Global Image Alterations," *Annual International Conference of the IEEE on Engineering in Medicine and Biology Society*, Boston, 30 August-3 September 2001, pp. 8062-8065.

[13] M. A. Dorairangaswamy and B. Padhmavathi, "An Effective Blind Watermarking Scheme for Protecting Rightful Ownership of Digital Images," *IEEE Region* 10 *Conference*, *TENCON* 2009, Singapore, 23-26 January 2009, pp. 1-6.

[14] L. N. Hu and L. G. Jiang, "Blind Detection of LSB Watermarking at Low Embedding Rate in Grayscale Images," In: M. Celik, G. Sharma, E. Saber and A. Tekalp, Eds., *Hierarchical Watermarking for secure Image Authentication with Localization*, *IEEE Transactions on Image Process*, Vol. 11, No. 6, 2002, pp. 585-595.

[15] N. F. Johnson, Z. Duric and S. Jajodia, "Information Hiding: Steganography and Watermarking—Attacks and Countermeasures," Kluwer Academic Press, Norwell, 2001,

[16] P. Wong, "A Watermark for Image Integrity and Ownership Verification," *Image Processing*, *Image Quality, Image Capture Systems Conference*, Portland, May 1998, pp. 374-379.

[17] Y. Q. Shi, "Reversible Data Hiding," *Digital Watermarking*, Vol. 3304, 2005, pp. 1-12. doi:10.1007/978-3-540-31805-7_1

[18] E. T. Lin and E. J. Delp, "A Review of Fragile Image Watermarks," *Proceedings of the Multimedia and Security Workshop at ACM Multimedia*'99, Orlando, October 1999, pp. 35-39.

[19] K. Pal, G. Ghosh and M. Bhattacharya, "A Novel Digital Image Watermarking Scheme for Data Security Using Bit Replacement and Majority Algorithm Technique," *Watermarking*, Vol. 1, 2012, pp. 117-132.