

Sliding Window used for Robustness Optimization Employing Neighborhood Concept and Genetic Algorithm

Sachin Goyal and Roopam Gupta

Department of Information Technology, U.I.T, R.G.P.V, Bhopal

Abstract

Fast distribution of digital contents through open networks is not posing a significant problem due to digital media revolution. Modern technologies have also reduced the reproduction time of digital media and its fast distribution. However, this facility has also the darker side where unauthorized users can tamper its contents and manipulate the digital data thus giving rise to serious security concerns. This problem has to be addressed very seriously. Digital watermarking techniques have recently evolved to address the above problems. The usage of these digital watermarks prevent illegal reproduction and usage of digital data as well as help in identifying the origin, author, owner etc even after various manipulations or attack on the digital data. A number of watermarking techniques in spatial and frequency domain were given by various researchers which suffered from problems robustness. Genetic algorithm provides an alternative way of creating watermarks with Promising values of robustness aspect of watermarking. This paper deals with design and development of a new watermarking technique which uses genetic algorithm to identify locations within the cover image for watermark insertion in spatial domain and then apply the average neighborhood concept for the purpose of watermark insertion and extraction ensuring higher robustness and resilience to several possible image attacks. Genetic search often produces same watermark locations in different populations for watermark insertion resulting in poor value of robustness, which need to be checked. Sliding window concept introduced in this paper uses a set of a few genes which are serially shuffled to get new set of locations for watermarking during each population generation and helps in enhancing robustness aspect of watermarking. Roulette-wheel selection has been used while using the genetic algorithms developed in the paper.

Keywords: *Digital watermarking, Genetic Algorithm, Robustness, Sliding window concept, Neighborhood concept*

1. Introduction

There are many watermarking methods which have been developed to embed a watermark into the cover image. Some of the important contributions of various proposed methods of digital watermarking during initial research period in digital watermarking are presented here. A watermark was generated using the least significant bit of the original image by Schyndel, *et al.*, [1] to produce the watermarked image. The watermark was extracted from a suspected image by taking the least significant bits at the proper locations. Cross-correlation of the original and extracted watermark was made by Schyndel, which showed that the resulting image contained an invisible watermark with simple extraction procedures. Cox, *et al.*, [2] pointed that, in order for a watermark to be robust to attack, it must be placed in perceptually significant areas of the image. Yen and Huey, *et al.*, [3] chose pair of position with same quantization scale in standard JPEG quantization table as cover image and it use the concept of discrete cosine Transformation. Huang, *et al.*, [4] paper used a concept of DCT and proposed a Progressive watermarking Techniques with genetic algorithm. B.sikander, *et al.*, [5]

presents a new method for adaptive watermark strength optimization in discrete cosine transform domain. In this paper GA is used as to intelligently select watermark strength. Proposed method use genetic algorithm to optimize the strength of the watermark for mid band DCT coefficients. M.J Anwar, *et al.*, [6] this method is more robust against JPEG compression. In this method genetic algorithm is used to find best position in image block that the positions have a mathematical relationship. Somying Promcharoen and Yuttapong Ranganseri [7] this approach used a fuzzy C-mean algorithm to classify 8x8 DCT block as texture or non-texture region. In this paper, a digital image watermarking by using genetic algorithm to optimize parameters used in block based DCT watermark embedding. M. Rafiq [8] have proposed a watermarking algorithm in the DCT domain using an evolutionary algorithm. Wang, Yulin and Pearmain Alan [9] present two kinds of watermarking techniques. One is based on estimation of the luminance value of the central pixel with it's adjacent pixels in one area in special domain and another is based on DCT domain.

Z.wai [10], *et al.*, was proposed new embedding and extracting method with genetic algorithm. The proposed embedding and extracting method was employed to accelerate the genetic watermarking. Gu Tianming and Wang Yanjie [11] proposed a digital watermarking algorithm based on DWT coefficient.

Section 2 discusses the Algorithm for optimization of Robustness a using genetic algorithms. Section 3 and 4 discusses the Neighborhood and sliding window concept respectively. Section 5 show Experimental Result and Conclusion is given in section 6.

2. Algorithm Optimization of Robustness with Sliding Window Neighborhood Concept Using Genetic Algorithm

The following conventions and assumptions apply to the embedding algorithm and extraction algorithms given as below.

- (1) *cimage* used as a two dimensional cover image and *w* used as two dimensional watermark bit.
- (2) *fitness [i]* return the value of *i*th of the chromosome Robustness in the population.
- (3) Chromosome [*k*] returns *k*th chromosome in the population.
- (4) *new_cimage* defined as a modified image after inserting a watermark in specified location, *i.e.*,
watermarked image.
- (5) Population [*i,j*] defined *j*th genes of *i*th population.
- (6) *new_population (i,j)* is obtained after performing crossover and mutation.

Step 1: Let the cover image used to embed the watermark be given as

$$cimage = [c_{11}, c_{12}, \dots, c_{ij}, \dots, c_{mc \times nc}], \quad \text{where } 1 \leq i \leq mc \text{ and } 1 \leq j \leq nc \quad (1)$$

Step 2 : *cimage* is now reshaped into row vector containing *mc*nc* entries

$$cimage((i-1)*nc+j) = cimage(i,j), \quad \text{where, } 1 \leq i \leq mc \text{ and } 1 \leq j \leq nc \quad (2)$$

This produces a row vector

$$cimage(c_1, c_2, \dots, c_i, \dots, c_{mc*nc}) \quad (3)$$

The size of the cover image is given as *s = mc*nc*

Step 3: Let the watermark be given as

$$w = [w_{11}, w_{12}, \dots, w_{ij}, \dots, w_{mn}], \quad \text{where } 1 \leq i \leq m \text{ and } 1 \leq j \leq n \quad (4)$$

Step 4: Now the array *fitness* is initialized with all zeros.

$$fitness = [0, 0, 0, 0 \dots pop_size \text{ times}] \quad (5)$$

Step 5 : The array *population* is created randomly as

$$Population = [p_{11}, p_{12}, \dots, p_{ij}, \dots, p_{popsize*no_of_genes}] \quad (6)$$

where $1 \leq i \leq \text{pop_size}$ and $i < j \leq \text{no_of_genes}$ And $p_{ij} = 0$ or 1 (randomly assigned)

Step 6 : Now, the watermark is converted into a single dimension vector W_{m1} as

$$W_{m1} = [w_1, w_2, \dots, w_i, \dots, w_{m \times n}] \quad (7)$$

The total number of watermark locations is given by $m \times n$.

Step 7: Now the individual chromosomes from the population are extracted.

$$\text{Chromosomes } [k] = [p_{k1}, p_{k2}, \dots, p_{ki}, \dots, p_{kno_of_genes}] \quad (8)$$

where, $1 \leq k \leq \text{pop_size}$ and $p_{k1}, p_{k2}, \dots, p_{ki}$ are the genes of the population.

Step 8: Now, the individual chromosomes are brought into variable_range as following.

First of all, the watermark locations in the cover image $x(i)$ are found.

$$x(i) = \sum_{k=0}^{i+1} \{ \text{chromosome}(2i - 1 + k) \} 2^{-(k+1)} \quad (9)$$

where $1 \leq i \leq \text{watermark_location}$

then

$$x(i) = 1 + \text{fix}((\text{variable_range} - 1) * x(i) / (1 - 2^{-\text{no_of_genes/watermark_location}})) \quad (10)$$

where $1 \leq i \leq \text{watermark_location}$

Step 9 : : Now the Pixel intensity value of the cover image as modified according to average neighborhood pixel intensity.

Firstly calculate the average pixel intensity of selected location $x(i)$ by using neighbourhood concept.

$$\begin{aligned} &\text{if } \text{cimage}(x(i)) < \text{average_pixe_intensity} \\ &\text{new_cimage}(x(i)) = \text{average_pixel_intensity} + E_s \end{aligned} \quad (11)$$

$$\begin{aligned} &\text{where } 1 \leq i \leq \text{watermark_location} \text{ and } W_{m1}(i) = 1, \\ &\text{if } \text{cimage}(x(i)) > \text{average_pixel_intensity} \\ &\text{new_cimage}(x(i)) = \text{average_pixel_intensity} - E_s \end{aligned} \quad (12)$$

where $1 \leq i \leq \text{watermark_location}$ and $W_{m1}(i) = 0$, E_s is embedding strength

Step 10: Now Calculate fitness function based on Normalized Correlation(NC). NC is the correlation between the embedded watermark w and the extracted watermark w' which has been normalized by the energy watermark.

$$\text{NC}(w \text{ } w') = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} W(i, j) \times W'(i, j)}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} W^2(i, j)} \quad (13)$$

Step 11: Now maxfitness and bestindividual index k is derived

$$\text{fitness}(i) < \text{fitness}(k) \quad \forall i : 1 \leq i \leq \text{pop_size} \text{ and } i \neq k, \quad (14)$$

Step 12 :Now apply roulette-wheel selection strategy.

(a)Find the total fitness(F) of current generation as follow

$$F = \sum_{i=1}^{\text{pop_size}} \text{fitness}(i) \quad (15)$$

(b) calculate the probability of a selection P_i for each chromosome

$$P_i = \text{fitness}(i) / F \quad (16)$$

(c) Calculate the cumulative probability q_i for each chromosome.

$$q_i = \sum_{j=1}^i P_j \quad (17)$$

where i is varying from 1 to pop_size

The selection process is based on spinning the roulette-wheel pop_size times.

Step 13 : Now the crossoverpoint is selected randomly such that

$$1 \leq crossoverpoint < no_of_genes \quad (18)$$

Now crossover between chromosome1 & chromosome2 is performed as per following equation and new chromosome pair called newChromosome_pair is generated

$$newChromosome_pair(1,j) = chromosome1(j) \quad (19)$$

$$newChromosome_pair(2,j) = chromosome2(j) \quad (20)$$

$$\forall j \therefore, 1 \leq j \leq crossoverpoint$$

OR

$$newChromosome_pair(1,j) = chromosome2(j) \quad (21)$$

$$newChromosome_pair(2,j) = chromosome1(j) \quad (22)$$

$$\forall j \therefore, crossoverpoint < j \leq no_of_genes$$

Step 14 : Now new_population is defined as

$$new_population(i,j) = newChromosome_pair(1,j) \quad (23)$$

$$new_population(i+1,j) = newChromosome_pair(2,j) \quad (24)$$

$$\forall j \therefore, 1 \leq j \leq no_of_genes \text{ and } r < crossoverprobability, \text{ where}$$

r is random number generated between $0 \leq r < 1$

If $r > crossoverprobability$ then crossover is not performed and new_population is given by

$$new_population(i,j) = chromosome1 \text{ and} \quad (25)$$

$$new_population(i+1,j) = chromosome2 \quad (26)$$

$$\forall j \therefore, 1 \leq j \leq no_of_genes$$

Step 15: Now the new_population is redefined such that each chromosome in the new_population is mutated depend on the mutation probability.

$$new_population(i,j) = 1 - (new_population(i,j)) \quad (27)$$

$$\forall i, j \therefore, 1 \leq i \leq pop_size \text{ and } 1 \leq j \leq no_of_genes \text{ and } r < mutation_probability,$$

where r is random number generated between $0 \leq r < 1$

Step 16: $gen_count = gen_count + 1$

Step 17: $population = new_population$

Repeat all step 7 to 16 for $gen_count \leq no_of_generation$

Now the maxfitness and best watermark location is obtained.

Watermark Extraction : : For the purpose of watermark extraction following formula is adapted in average neighborhood pixel insertion method.

Firstly we calculate average pixel intensity of selected location

$$\begin{aligned} \text{Sum_intensity} = & cimage(loc) + cimage(loc+1) + cimage(loc-1) + \\ & cimage(loc+ncol) + cimage(loc+ncol+1) + cimage(loc+ncol-1) + cimage(loc-ncol) + \\ & cimage(loc-ncol+1) + cimage(loc-ncol-1) \end{aligned} \quad (28)$$

$$\text{average_pixel_intensity} = \text{sum_intensity} \setminus 3 * 3 \quad (29)$$

If average pixel intensity value of watermarked image is greater than cover image pixel intensity then watermark bit 1 is obtained otherwise 0.

$$\text{Watermark bit} = \begin{cases} 1 & \text{average_pixel_intensity}(loc) > \\ & cimage(loc) \end{cases}$$

$$\begin{cases} 0 & \text{otherwise} \end{cases}$$

$$\text{Where } 1 \leq loc < mc * nc \quad (30)$$

3. Neighborhood Concept

In this concept, first of all a suitable location is selected by genetic algorithm for watermark insertion. Now, for the purpose of insertion of watermark, the neighborhood of the concerned location is utilized. A neighborhood of 3*3 pixels is chosen surrounding the concerned pixel and the average of the pixel intensity values of the neighborhood pixel is compared with the pixel intensity where the watermark has to be inserted. If the pixel intensity value is higher than this average a binary 1 is inserted else a binary 0 is inserted in the concerned pixel.

There are two Case occur when be inserted watermark bit in to cover image.

In first case, when watermark mark bit 1 is inserted if the value of selected location pixel is less then average intensity value then increase the intensity of selected location according to equation(31).

$$SL = API + Es \quad (31)$$

In second case when watermark bit 0 is inserted if the value of selected location pixel is grater then average intensity value then decrease the value of selected location according to equation (32)

$$SL = API - Es \quad (32)$$

Where SL- Selected location, API-Average Pixel Intensity and Es embedding strengths

The advantage of using neighborhood concept for watermark is reflected in terms of higher robustness. This is due to the reason that during image attacks, the average of surrounding pixels is not significantly affected and thus it prevents the watermark bit from getting modified.

4. Sliding Window Concept

When the set of number of genes remain same for each location they may leads to same set of bits giving rise to similar watermark location after decoding hence the starting location of the watermark bit along with the number of genes used for creating the location both is varied to create dissimilar location each time this will avoid the redundancy of the watermark location and help in unique and global search in watermark location. This unique concept termed as sliding window is introduced here, where the sliding window consist of the set of bits used for creating watermark location and the set is shifted horizontally to propose a new starting location and set of length increase each time. This ensures that different watermark location to be created each time after decoding.

So, mathematically expressed each value $x(i)$ can be expressed as

$$x(i) = \sum_{K=0}^{i+1} \{chromosom(2i - 1 + k)\} 2^{-(k+1)} \quad (33)$$

This can be decoded to obtain the corresponding watermark location.

$$x(i) = 1 + fix((variable_range-1)*x(i)/(1-2^{-no_of_genes/watermark_location})) \quad (34)$$

where $1 \leq i \leq watermark_location$

Equation (33) can be further analytically examined to show that the probability of repetition of same location is minimized.

- (1) Case 1 Consecutive Watermark location.
- (2) Case 2 Non Continuous Watermark location.

Case 1 Consecutive Watermark Location

So, i th location $x(i)$ is given by

$$x(i) = \sum_{K=0}^{i+1} \{chromosom(2i - 1 + k)\} 2^{-(k+1)}$$

Similarly (i+1)th location

$$x(i+1) = \sum_{K=0}^{i+2} \{chromosome(2(i+1) - 1 + k)\}2^{-(k+1)}$$

Difference between equation ii and iii is given by

$$\begin{aligned} &= x(i+1) - x(i) \\ &= \sum_{K=0}^{i+2} \{chromosome(2(i+1) - 1 + k)\}2^{-(k+1)} \\ &\quad - \sum_{K=0}^{i+1} \{chromosome(2i - 1 + k)\}2^{-(k+1)} \\ &= \sum_{K=0}^{i+2} \{chromosome(2i - 1 + k + 2)\}2^{-(k+1)} \\ &\quad - \sum_{K=0}^{i+1} \{chromosome(2i - 1 + k)\}2^{-(k+1)} \end{aligned} \quad (35)$$

For example putting i=1 , it produce 1st and 2nd location

$$\begin{aligned} &= \sum_{K=0}^3 \{chromosome(k + 3)\}2^{-(k+1)} - \sum_{K=0}^2 \{chromosome(k + 1)\}2^{-(k+1)} \\ &= [chromosome(3)*2^{-1} + chromosome(4)*2^{-2} + chromosome(5)*2^{-3} + \\ &\quad chromosome(6)*2^{-4}] \\ &\quad - [chromosome(1)*2^{-1} + chromosome(2)*2^{-2} + chromosome(3)*2^{-3}] \\ &= [chromosome(3) - chromosome(1)]*2^{-1} + [chromosome(4) - \\ &\quad chromosome(2)]*2^{-2} + [chromosome(5) - \\ &\quad chromosome(3)]*2^{-3} + chromosome(6)*2^{-4} \end{aligned} \quad (36)$$

To make above expression zero necessary condition required

- chromosome(3) = chromosome(1)(A)
- chromosome(4) = chromosome(2)(B)
- chromosome(5) = chromosome(3) (c)
- chromosome(6) = 0(D)

It is extremely difficult and very less probability that all these condition can be satisfied simultaneously. So probability of happen of simultaneously occur condition (A)(B)(C)(D) together is given by

$$S(H)=P(A)*P(B)*P(C)*P(D) \quad (37)$$

Where P(A) = Probability of occurrence of condition(A) , P(B) = Probability of occurrence of condition(B),P(C)= Probability of occurrence of condition(C), P(D) = Probability of occurrence of condition(D)

The above product much smaller than each individual value of P(A), P(B), P(C), P(D).

Case 2 Non Continuous Watermark Location

So, (i+2)th location x(i+2) is given by

$$x(i) = \sum_{K=0}^{i+3} \{chromosome(2(i+2) - 1 + k)\}2^{-(k+1)}$$

similarly (i+5)th location

$$x(i+5) = \sum_{K=0}^{i+5} \{chromosome(2(i+5) - 1 + k)\}2^{-(k+1)}$$

We take difference between equation ii and iii is given by

$$= x(i+5) - x(i+2)$$

$$= \sum_{K=0}^{i+5} \{chromosome(2(i+5) - 1 + k)\} 2^{-(k+1)} - \sum_{K=0}^{i+3} \{chromosome(2(i+2) - 1 + k)\} 2^{-(k+1)} \quad (38)$$

Example putting i=1 , it produce 4th and 6th location

$$= \sum_{K=0}^6 \{chromosome(11+k)\} 2^{-(k+1)} - \sum_{K=0}^4 \{chromosome(k+5)\} 2^{-(k+1)}$$

$$= [chromosome(11)*2^{-1} + chromosome(12)*2^{-2} + chromosome(13)*2^{-3} + chromosome(14)*2^{-4} + chromosome(15)*2^{-5} + chromosome(16)*2^{-6} + chromosome(17)*2^{-7}] - [chromosome(5)*2^{-1} + chromosome(6)*2^{-2} + chromosome(7)*2^{-3} + chromosome(8)*2^{-4} + chromosome(9)*2^{-5}]$$

$$= [chromosome(11) - chromosome(5)]*2^{-1} + [chromosome(12) - chromosome(6)]*2^{-2} + [chromosome(13) - chromosome(7)]*2^{-3} + [chromosome(14) - chromosome(8)]*2^{-4} + [chromosome(15) - chromosome(9)]*2^{-5} + chromosome(16)*2^{-6} + chromosome(17)*2^{-7} \quad (39)$$

To make above expression zero necessary condition required

- chromosome(11) = chromosome(5)(E)
- chromosome(12) = chromosome(6)(F)
- chromosome(13) = chromosome(7) (G)
- chromosome(14) = chromosome(8)(H)
- chromosome(15) = chromosome(9)(I)
- chromosome(16) = chromosome(17)=0(J)

it is extremely difficult and very less probability that all these condition can be satisfied simultaneously. So probability of happen of simultaneously occur condition (E)(F)(G)(H)(I)(J) together is given by

$$S(H) = P(E)*P(F)*P(G)*P(H)*P(I)*P(J) \quad (40)$$

Where P(E) = Probability of occurrence of condition(E), P(F) = Probability of occurrence of condition(F),P(G)= Probability of occurrence of condition(G), P(H) = Probability of occurrence of condition(H), P(I)= Probability of occurrence of condition(G), P(J) = Probability of occurrence of condition(H)

The above product much smaller than each individual value of P(E), P(F), P(G), P(H),P(I),P(J).

From the analysis of difference between two continuous and non continuous location as shown by equation (35) and (38) . There is a very less probability that the difference will yield to zero, there shall be at most of time some finite value of difference.

Thus all x(i) shall have some difference and this will yield to some different watermark location each time when they are decoded. This help in avoidance of redundancy of location and prevents the genetic algorithm to find local maxima while conducting search of proper locations for watermark introduce to obtain optimize in term of robustness as decided by the fitness function.

The sliding window concept can be illustrated by the following example. Let us take, for instance the number of watermark bits = 5 and the number of genes = 15

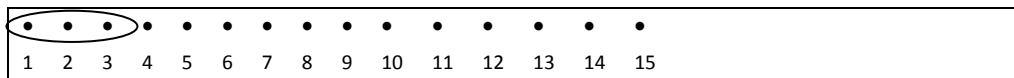
Let us take watermark bits (w)	=	5 bit	
No. of Genes	=	15 (5x3)	
Watermark_genes_size (z)	=	15/5	= 3 bit

Watermark_genes_size represent a number of genes construct a watermark but in sliding window concept ,size of watermark genes size increase by one every time when you insert a watermark bit and change the starting location by 2, while taking a window size as 3. So in this way we remove a repeated location this can illustrated below.

For 1st Location

Starting location = 1

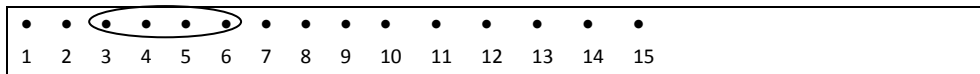
Watermark_genes_size = 3



For 2nd Location

Starting location = 3

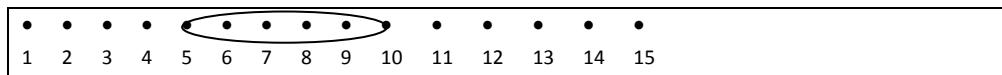
Watermark_genes_size = 4



For 3rd Location

Starting location = 5

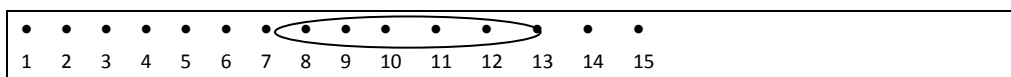
Watermark_genes_size = 5



For 4th Location

Starting location = 7

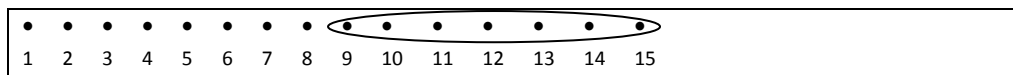
Watermark_genes_size = 6



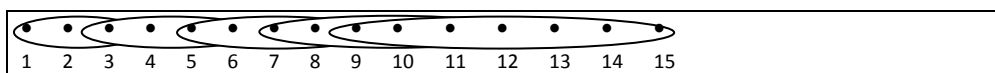
For 5th Location

Starting location = 9

Watermark_genes_size = 7



In General

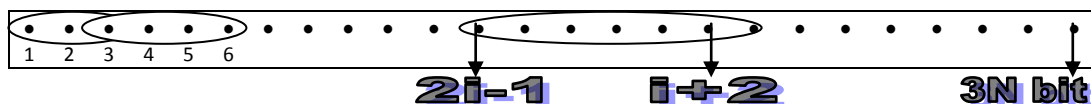


Generalization of sliding window

let us taken N watermark bit

for ith location Starting location = $2i - 1$

watermark_gene_size = $i+2$



5. Experimental Results

All experiments were conducted on genuine Intel (R) CPU T-2050 @1.60GHZ, 1GB of RAM. The operating system used was Microsoft Windows Vista, Version 2007. In all of this experiment simulation take Lena (512 x 512) as a cover image (Figure 1) and

different payload of binary watermark bits are embedded into cover image. Table 1 represents the various parameters setting used in the experiments for genetic algorithm. Roulette-wheel selection strategy is used to find out the desired locations for watermark insertion for Robustness optimization. Table 2 show value of NC under different type of attack. (*i.e.*, Median filter , Mean filter, Sharpening , Salt & Pepper, Gaussian attack, Low Pass Filter , JPEG Compression). Figure 2-5 show different type of simulation result. Table 3 show simulation result of extracted binary image watermark. For Comparison with other method, The NC value of Lena image (512*512) with payload 4096bits shown in Table 4. Figure 6 and 7 show watermarked image of 4096bits. Figure 8 different type of attack with varying payloads.



Figure 1. Lena Image (512X512)

Table 1. Genetic Algorithm Parameters

Parameters	Values
Generation	20-50
Population Size	6
Crossover Probability	0.90
Mutation rate	0.025
Window Size	3

Table 2. Variation of NC with Payload

S.No.	Payload (bits)	Median Filter Attack	Mean Filter Attack	Salt & Pepper Attack	Sharpening Attack	Gaussian Noise	Low Pass Filter	JPEG QF=50%
1	512	0.9968	1	0.9935	0.9745	0.6419	0.9417	0.9781
2	1024	0.9903	0.9887	0.9920	0.9632	0.6096	0.9289	0.9722
3	2048	0.9880	0.9840	0.9832	0.9480	0.6000	0.9125	0.9691
4	4096	0.9856	0.9820	0.9744	0.9412	0.5900	0.8971	0.9686
5	16384	0.9641	0.9767	0.9581	0.9280	0.5634	0.8615	0.8912

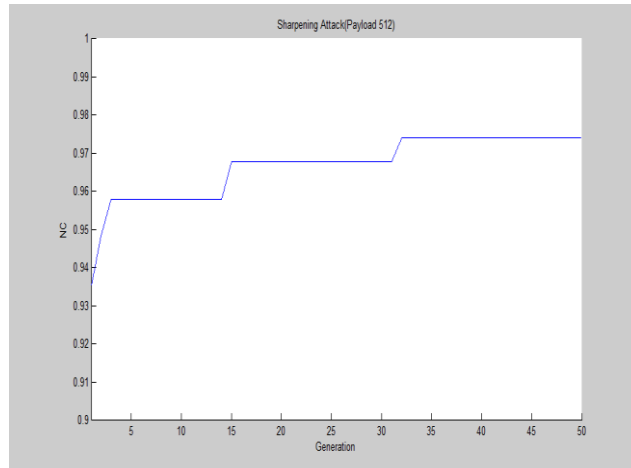


Figure 2. Sharpening Attack with Payload 512 Bits

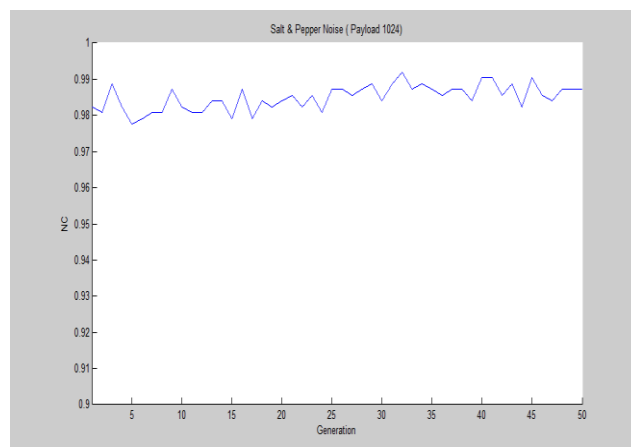


Figure 3. Salt & Pepper Noise with Payload 1024 Bits

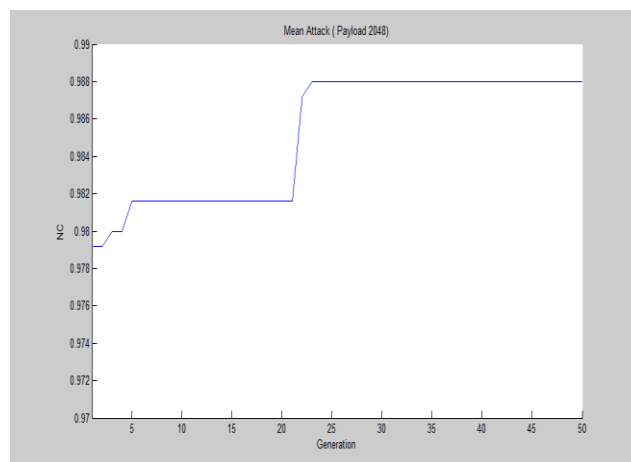


Figure 4. Mean Filter Attack with Payload 2048 bits

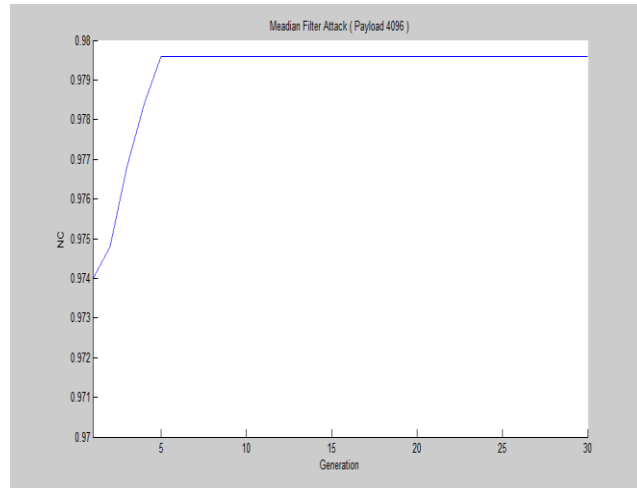


Figure 5. Median Filter Attack with Payload 4096 Bits












Figure 6. Watermarked Image (Median Filter Attack)



Figure 7. Watermarked Image (Salt & Pepper Noise)

Table 3. Extracted Watermark after Different Type of Attack

S.No	Cover Image (512x512)	Watermark Image(64x64)	Different Type of Attack	Extracted Watermark
1			Median Attack NC= 0.9826	

			Mean Filter Attack NC=0.9715	
			Sharpening NC=	
			Salt & Pepper Noise NC=0.9751	
			Gaussian Noise NC=0.6034	
			Low Pass Filter NC=0.8890	
			JPEG QF=50% NC=0.9687	

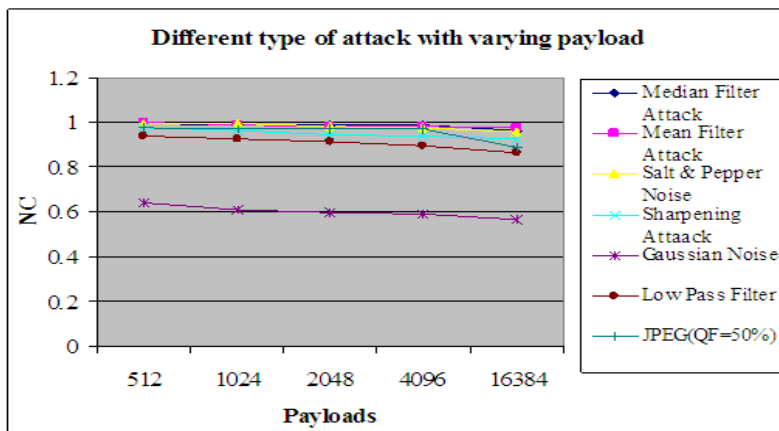


Figure 8. Different Type of Attack with Varying Payloads

Table 3. Comparison of Robustness our Method with other Method

S.N o.	Types of Attack	Our Method (NC)	Yen Method[3] (NC)	Somying Method[7] (NC)	Rafiq Method[8] (NC)	Wang Method[9]
1	Salt & Pepper	0.9744	0.97	-	0.97	0.95
2.	Median Filter	0.9856	0.97	-	0.99	0.97
3	Mean Filter	0.9820	0.94	0.6965	0.96	0.97
4	Sharpening Filter	0.9412	0.96	-	0.98	0.98
5	JPEG (QF=50%)	0.9686	0.96	0.8631	1.0	0.81
6	Low Pass Filter	0.8971	-	0.6680	-	-

6. Conclusion

In this paper, we have developed and demonstrated the technique of employing genetic algorithm along with average neighborhood and sliding window concept with Roulette-wheel strategy to insert a binary watermark into the gray scale cover image in spatial domain is fruitful for maintaining a higher value of robustness. The results shown in the paper indicates that the technique proposed in the paper meets its objectives in terms of providing a satisfactory value of robustness while maintaining a reasonable value of robustness and a moderate payload. The experimental results showed that the proposed method can improve the quality of the watermarked image and give more robustness of the watermark as compare with the other previous method.

Reference

- [1] R. G. Van Schyndel, A. Z. Tirkel and C. F. Osborn, "A Digital Watermark", in Proc. IEEE International Conf. Image processing, vol. 2, (1994), pp. 86-92.
- [2] J. Cox and J. Kilian, "A Secure Robust Watermark for Multimedia", in Proc. First International Workshop, Lecture notes in computer science, vol. 1174, pp. 185-206.
- [3] S. H. Yen, *et al.*, "A Watermarking Technique based on JPEG quantization Table", Taipei, Taiwan: Proceeding of the 2006 International Computer Symposium ICS, (2006).
- [4] Huang, Pan, Shieh and Wang, "Progressive watermarking techniques with genetic Algorithms", IEEE, (2002).
- [5] B. Sikander, M. Iahiaq, M. A. Jaffer, M. Tariq and A. Mirza, "Adaptive digital watermarking of images using Genetic Algorithm", in Proc IEEE.
- [6] M. J. Anwar, M. Ishtiaq, M. Iqbal and M. Jaffar, "Blocked-based digital image watermarking using Genetic Algorithm", in Proc IEEE international conference on Emerging Technologies, (ICET), (2010), pp. 204-208.
- [7] S. Promchareon and Y. Rangsaneri, "Genetic watermarking with Block-Based DCT Clustering", in IEEE International Symposium on Communication and Information Technologies ISCIT, (2008), pp. 346-351.
- [8] M. Rafiq and M. Ebrahimi, "A Robust Evolutionary Based Digital Image Watermarking Technique in DCT Domain", in Proc IEEE Seventh International Conference on Computer Graphics, Imaging and Visualization, (2010), pp. 105-109.
- [9] Y. Wang and P. Alan, "Blind image data hiding based on self reference", Pattern Recognition Letters, vol. 25, (2004).
- [10] Z. Wai, J. Dai and J. Li, "Genetic watermarking based on DCT domains techniques", IEEE CCECE/CCGEL, Ottawa, (2006) May, pp. 2365-2368.
- [11] G. Tianming and W. Yanjie, "DWT-based Digital Image Watermarking Algorithm", proc IEEE Tenth International Conference on Electronics Measurement and Instruments, ICEMI, (2011), pp. 163-166.

Authors



Roopam Gupta, She is a member of IETE and ISTE society. She was born in India in October 1971. She has done her BE in Electronics and Communication from GEC Bhopal, M.Tech. in Digital Communication from MANIT Bhopal and PhD from Barkatullah University. Her area of interest includes watermarking, ad hoc networks and security. She is presently working as head in Department of Information Technology, UIT, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal.



Sachin Goyal, He was born in India in March 1979. He has done his BE in Computer Science & Engg From ITM, RGPV, Bhopal, M. Tech in Artificial Intelligence from SATI, RGPV, Bhopal and pursuing PhD in IT from RGPV, Bhopal. His area of Interest includes Digital watermarking, Theoretical Computer science and computer network. He is presently working as a Assistant professor in Department of Information Technology, UIT, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal.