Authentication of JPEG Images Based on Genetic Algorithms

Venkata Gopal Edupuganti and Frank Y. Shih*

Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA

Abstract: This paper presents an efficient authentication method for JPEG images based on Genetic Algorithms (GA). The current authentication methods for JPEG images require the receivers to know the quantization table beforehand in order to authenticate the images. Moreover, the quantization tables used in the JPEG compression are different for different quality factors, thus increasing the burden on the receivers to maintain several quantization tables. We propose a novel GA-based method which possesses three advantages. First, the computation at the receiver end is simplified. Second, it is no more required for the receivers to maintain quantization tables. Third, the method is resistant against Vector Quantization (VQ) and Copy-Paste (CP) attacks by generating the authentication information which is unique with respect to each block and each image. Furthermore, we develop a two- level detection strategy to reduce the false acceptance ratio of invalid blocks. Experimental results show that the proposed GA-based method can successfully authenticate JPEG images under variant attacks.

Keywords: Watermarking, genetic algorithm, authentication, and vector quantization.

1. INTRODUCTION

Digital images are transmitted more often over the Internet now than ever before. Unfortunately, free-access digital multimedia communication provides virtually unprecedented opportunities to pirate copyrighted material. Due to wide availability of image manipulation software, digital images suffer from different kinds of attack, such as modification and removal of the content. Therefore, the tasks of detecting and tracing copyright violations have stimulated significant interests among engineers, scientists, lawyers, artists, and publishers, to name a few. As a result, the research in watermark authentication has become very active in recent years, and the developed techniques have grown and been improved a great deal.

Authentication of digital images can be carried out in two ways: *digital signature* and *digital watermarking* [1-3]. A digital signature, such as hash value, which is a cryptographic technique, has two drawbacks in authenticating images. First, in order to check the authenticity, the verifier must know the digital signature beforehand, so the sender has to send it through a separate secure communication channel. This generates an overhead on bandwidth. Second, although digital signatures authenticate the images, they fail to locate which parts of the images were altered. One way of achieving tamper localization is dividing the image into blocks and sending the digital signature associated with each block to the receiver. However, this increases the overhead on bandwidth.

*Address correspondence to this author at the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA; Tel: 973-596-5654; Fax: 973-596-5777; E-mail: frank.y.shih@njit.edu

Digital watermarking is categorized into three types: *robust, fragile,* and *semi-fragile*. Robust watermarking [1, 3] is used to protect the copyright of digital media. Such a watermark stands with the image even after signal processing operations, such as compression, translation, and rotation, are applied on the image. Fragile watermarking [1, 3-6] on the other hand becomes invalid even if the image is slightly modified. This feature allows us to verify the authenticity of digital images. Semi-fragile watermarking [1, 3, 7-9], which comes in between robust and fragile, can distinguish between malicious (such as cropping, modification, etc.) and non-malicious attacks (such as compression, smoothing, etc.).

In general, there are two ways to perform digital watermarking: one is in spatial domain and the other is in frequency domain [1, 3]. Spatial-domain techniques are simple as they embed the watermark directly into the pixels, such as Least-Significant-Bit (LSB) modification. The disadvantage is that the attacker can easily detect the watermark. On the other hand, frequency-domain approaches involve complex calculations, but they enhance the data security as compared against spatial-domain techniques. Such approaches first apply Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), or Discrete Wavelet Transform (DWT) to transform the image data from spatial domain to frequency domain, and then embed the watermark into the frequency-domain coefficients.

Most of fragile watermarking methods for image authentication have been designed for authenticating uncompressed images [4-6, 9]. Even though they have the advantage of recovering the tampered blocks in addition to tamper localization, their storage space and bandwidth required to transfer an uncompressed image are very high. As JPEG com-

1874-0618/09

pression is a widely-used image format, several authentication methods for JPEG images [7, 8, 10, 11, 13] have been proposed.

Lin and Chang [8] designed a semi-fragile watermarking system for authenticating JPEG visual content using two invariant properties of DCT coefficients. Ho and Li [7] proposed a semi-fragile watermarking scheme for authentication of JPEG images using nine-neighborhood block consideration to provide block dependency. This scheme is tolerant to JPEG compression to a pre-determined lowest quality factor. Li [10] proposed a fragile watermarking scheme for authentication of JPEG images by considering all the DCT coefficients of each block. Wang *et al*. [11] proposed a fragile watermarking method for authentication of JPEG images that generates the watermark using the initial values of the quantized DCT coefficients as an input to the chaotic system and embeds the watermark into the LSB of the quantized DCT coefficients. As each block is watermarked individually (i.e. using only the information present in that block), the method suffers from Copy-Paste (CP) [4] and Vector Quantization (VQ) [12, 14] attacks. Note that all the above methods require the receivers to know the quantization tables beforehand for image authentication.

Fig. (**1**) shows the often-used JPEG procedure to compress an image. It involves splitting an input image into blocks of size 8×8, DCT, quantization, entropy coding, etc [11]. As aforementioned, the existing fragile and semi-fragile watermarking methods [7, 8, 10, 11] need the receiver to know the quantization table beforehand to authenticate the image. As different JPEG quality factors use different quantization tables, the receiver has to maintain a lot of quantization tables. We intend to overcome this problem to allow the receiver to authenticate the image irrespective of the quality factor and the quantization table used in creating the watermarked image. In this paper, we propose a GA-based method that can adjust the image such that the modified image after JPEG compression contains the authentication information in the DCT coefficients of each block to authenticate the image. Furthermore, the generation of authentication information guarantees the uniqueness with respect to each block (depending on block position) and each image for thwarting the CP and VQ attacks.

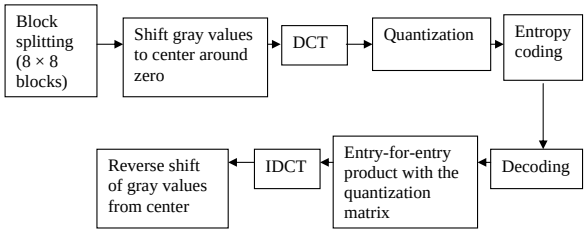

Fig. (1). The JPEG procedure.

The rest of this paper is organized as follows. Section 2 presents the GA-based watermark embedding method. Section 3 describes the authentication procedure. Section 4 shows experimental results. Finally, conclusions are drawn in Section 5.

2. THE GA-BASED WATERMARK EMBEDDING METHOD

2.1. The Overall Watermark Embedding Procedure

The block diagram of the GA-based watermark embedding procedure is shown in Fig. (**2**). Each step is briefly described below and will be explained in more details in later sections. Let *A* be an original grayscale image of size $N \times N$, where *N* is a multiple of 8. Let the quality factor of JPEG compression be denoted as *QF*.

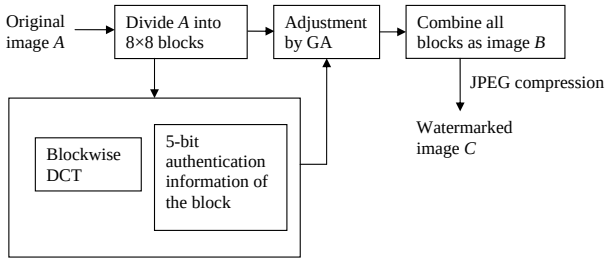

Fig. (2). The overall watermark embedding procedure.

Step$_e$ 1. As JPEG uses blocks of size 8 × 8, we divide the image *A* into a set of 8×8 blocks, named $\{b(i, j) \mid i, j = 1, 2, \cdots, N/8\}$, where i and j respectively denote the row and column numbers of the block. Mark each block $b(i, j)$ with a number X by traversing the blocks in the image from top-to-bottom and left-to-right as

$$X(i, j) = (i-1) \times (N/8) + (j-1). \quad (1)$$

Step$_e$ 2. Apply a chaotic map [4] to transform the block number *X* into a new mapping block number X' as

$$X'(i, j) = \{(k \times X(i, j)) \bmod M\} + 1, \quad (2)$$

where $M = N/8 \times N/8$ as the total number of blocks in the image, X, $X' \in [0, M-1]$, and $k \in [2, M-1]$ (a secret key) is a prime number. Note that if *k* is not a prime number, the one-to-one mapping cannot be achieved.

Step$_e$ 3. For each block $b(i, j)$, we generate the authentication information which will be described in Section 2.2.

Step$_e$ 4. For each block $b(i, j)$, we adjust its values using GA [15], so the modified block $b'(i, j)$ after JPEG compression contains the authentication information generated in step 3 in the LSB of integer part of five upper-left DCT coefficients as shown in Fig. (**3**).

Step$_e$ 5. Combine all modified blocks $b'(i, j)$ to generate a modified image *B*.

Step$_e$ 6. Compress the modified image *B* using JPEG compression with *QF* to obtain the watermarked image *C*.

2.2. Authentication Information Generation

In order to thwart the CP and VQ attacks, the authentication information needs to be unique per block and per image. Fig. (**4**) illustrates the authentication information generation. The chaotic-map block number $X'(i, j)$, computed in Sec-

tion 2.1, is unique to each block. We considered 16 bits for $X'(i, j)$ to support larger images i.e. with 8 bits we can only number 256 blocks which is suitable for images of size up to 128×128. The 11-bit clock information is obtained by using the bit-XOR ' $\oplus$ ' (i.e., bitwise exclusive-or) operation of six elements of image capture identity (which was recorded in the image header): year, month, day, hour, minute, and second. We attach the 11-bit information to the end of 16-bit stream of $X'(i, j)$ to generate the 27-bit block signature. Finally, the 5-bit authentication information is computed by subjecting the 27-bit block signature to CRC-5 checksum. As we are going to embed the authentication information in the low-level DCT coefficients, the choice of more than five bits will increase the computation load on GA adjustment process. On the other hand, the choice of fewer bits i.e. less than five will decrease the discriminative power of the authentication information. The secret key *k* of the chaotic map is shared with the receiver using the public key cryptographic algorithm [6].

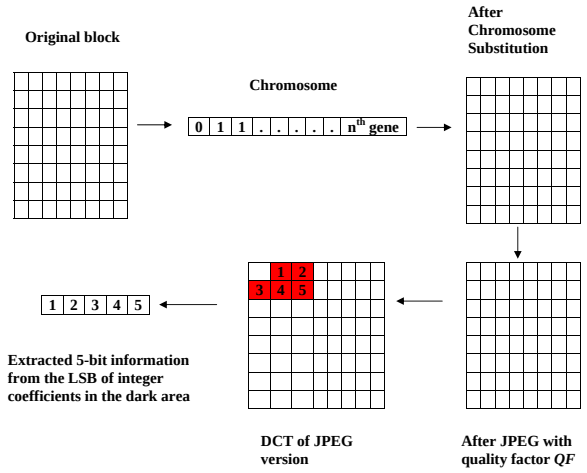

Fig. (3). GA procedure to adjust the block.

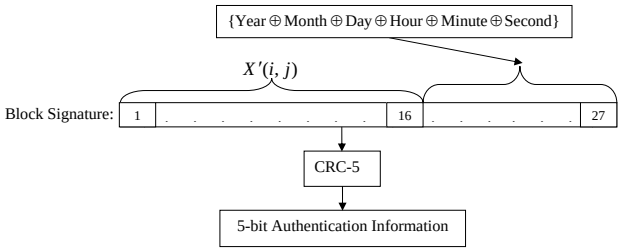

Fig. (4). The generation of the authentication information.

2.2.1. Cyclic Redundancy Checksum (CRC)

In the computation of CRC [6, 16], a k-bit string is represented by the coefficients of a k–1 degree polynomial; i.e., the coefficients of x^0 to x^{k-1}. The Most Significant Bit (MSB) of the k-bit string becomes the coefficient of x^{k-1}, the second MSB becomes the coefficient of x^{k-2}, and the Least Significant Bit (LSB) becomes the coefficient of x^0. For example, the polynomial representation of the bit string 1101 is x^3+x^2+1. To compute the CRC-r authentication information of a bit string, we pad 'r' zeros to the end of the bit string. Then we divide the resulting bit string with an r-degree polynomial. We subtract the remainder from the bit string padded with zeros. The resultant bit string after subtraction is the authentication information. The subtraction is done in modulo-2 arithmetic. To check if a CRC-r generated bit string is modified or not, we divide the bit string with the r-degree polynomial used in the generation stage. If there is a non-zero remainder after division, it means that the bit string is modified.

2.3. Adjustment by GA

Genetic algorithm (GA) is a randomized, parallel, and global search approach based on the mechanics of natural selection and natural genetics to find solutions of problems [15]. During last decade, GA has been widely applied in the digital watermarking community [17, 18]. Generally, GA starts with some randomly-selected genes in the first generation, called *population*. Each individual, called *chromosome*, in the population corresponds to a solution in the problem domain. An objective (or fitness function) is used to evaluate the quality of each chromosome. The chromosomes with high quality will survive and form the population of the next generation. By using the reproduction, crossover and mutation operations, a new generation is recombined to find the best solution. This process will repeat until a pre-specified condition is satisfied, or a constant number of iterations are reached.

We apply GA to adjust the pixel values in each block $b(i, j)$, such that the adjusted block $b'(i, j)$ after compression with quality factor *QF* contains the authentication information (generated as in section 2.2) in the LSB of the integer part of the five DCT coefficients as indicated in Figure 3. This algorithm is divided into three steps: substitution of chromosome in the original block, chromosome evaluation, and genetic algorithm. Section 2.3.3 presents the GA adjustment process in two different approaches. The two approaches use the details in sections 2.3.1 and 2.3.2 in the evaluation of each chromosome to find a suitable adjusted block.

2.3.1. Substitution of Chromosome in the Original Block

The idea is to modify the *p* LSBs of each pixel in the block, where *p* is equal to the chromosome length divided by 64. For example, if the chromosome length is 64, then only one LSB is modified in each pixel. We traverse the pixels from left-to-right and top-to-bottom, and substitute the first gene in the LSB of the first pixel, the second gene in the LSB of the second pixel, and so on as shown in Fig. (**5**). For another example, if the chromosome length is 64*p*, then the first *p* genes are used to replace the *p* LSBs of the first pixel, the next *p* genes are used to replace the *p* LSBs of the second pixel, and so on.

2.3.2. The Objective Function for Chromosome Evaluation

The objective function is the bit difference between the five extracted LSBs from the integer part of the DCT coefficients (as shown in Fig. **3**) and the 5-bit authentication in-

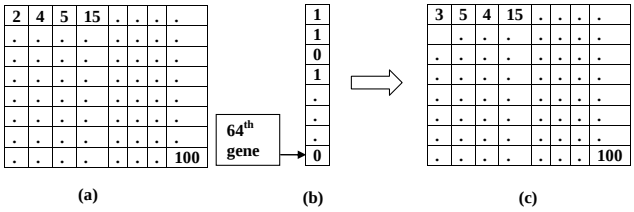

Fig. (5). (a) original 8×8 block, (b) 64-bit chromosome, and (c) adjusted block after substituting (b) in (a).

formation (as described in Section 2.2). We intend to minimize the difference, with '0' being treated as the best solution. The objective function can be expressed as

$$Objective\_function = \sum_{i=1}^{5} | a(i) - r(i) |, \quad (3)$$

where a is the 5-bit authentic information, and r is the retrieved LSBs from the integer part of the five DCT coefficients.

2.3.3. Genetic Algorithm to Adjust the Block

We present two types of block adjustment: *fixed-length chromosome* and *variable-length chromosome*. It is a trade-off between speed and watermarked image quality. The fixed-length chromosome algorithm runs faster, but produces a lower-quality watermarked image as compared to the variable-length chromosome algorithm.

(a) Fixed-Length Chromosome

In the fixed-length chromosome algorithm, the chromosome length is fixed and determined by the quality factor *QF* of JPEG compression. The chromosome length designed for the corresponding *QF* range is listed in Table **1**. The chromosome length increases with the decreasing *QF*, this is because if the quality factor is low we will lose more information i.e. there is a large difference between the original block and the compressed block, so we need to adjust a lot of information in each pixel compared to the images with high quality factor. On experimenting with different images, we came up with the numbers in Table **1** for better performance. For example if we choose chromosome length less than 192 in the case of quality factor greater than 80, some blocks take larger number of iterations to meet the stopping criteria. It aims at adjusting the block with acceptable image quality under reduced time consumption of the GA adjustment process. The GA adjustment process by the fixed-length chromosome algorithm is presented as follows:

Step$_f$ 1. Define the chromosome length, population size, crossover rate, replacement factor, and mutation rate. The initial population is randomly assigned with 0's and 1's.

Step$_f$ 2. Substitute each chromosome in the original block as explained in section 2.3.1 and evaluate the objective function for each corresponding chromosome by eq. (3).

Step$_f$ 3. Apply reproduction, crossover and mutation operators to generate the next generation of chromosomes.

Step$_f$ 4. Repeat steps 2 and 3 until the objective function value is equal to zero.

Step$_f$ 5. Substitute the final chromosome in the original block to obtain the modified block.

Table 1. The Chromosome Lengths Associated with Different *QF* Values

Quality Factor, *QF*	Chromosome Length
QF>80	192
65≤ *QF*≤*80*	256
55≤ *QF*≤*65*	320
45≤ *QF*≤*55*	384
QF<*45*	448

(b) Variable-Length Chromosome

In the variable-length chromosome algorithm, for improving the watermarked image quality we start the adjustment process with a chromosome size of 64. If a solution cannot be obtained before the termination condition is satisfied, we increase the chromosome size by 64 and execute the adjustment process again. This process is repeated until the termination condition is satisfied. The final chromosome size is selected based on the *QF* of the JPEG compression in Table1. The termination condition for the final chromosome size is to obtain an objective function value of zero and that of the intermediate chromosome sizes is a generation limit of 100. The GA adjustment process by the variable-length chromosome algorithm with $QF = 65$ is presented as follows and its flowchart is shown in Fig. (**6**).

Step$_v$ 1. Define the objective function, number of genes $n = 64$, population size, crossover rate, replacement factor, and mutation rate. The initial population is randomly assigned with 0's and 1's.

Step$_v$ 2. If $n < 256$, the termination condition is the maximum of 100 generations; otherwise, the termination condition is the objective function value being zero.

Step$_v$ 3. Substitute each chromosome in the original block as explained in section 2.3.1 and evaluate the objective function for each corresponding chromosome by eq. (3).

Step$_v$ 4. While the termination condition is not satisfied, apply reproduction, crossover, and mutation operators to generate the next generation of chromosomes, and compute the objective function for each corresponding chromosome.

Step$_v$ 5. If the objective function value of population is zero, substitute the chromosome in the original block to obtain the modified block; otherwise, $n = n + 64$ and go to step 2.

3. AUTHENTICATION

Authentication is the procedure that validates the authenticity of the received image. The main objective of authentication is to detect the tampered blocks with low false acceptance ratio. Our authentication algorithm is presented as follows:

Step$_a$ 1. Let *D* be a testing image, which is divided into blocks of size 8 × 8. Let the group of blocks be denoted as *attack_blocks*.

Step$_a$ 2. Retrieve the secret key *k*, the polynomial used in CRC checksum, and the clock values.

Step$_a$ 3. Compute the block numbers and the corresponding mapping block number by the same way as in Section 2.1.

Step$_a$ 4. (Level-1 detection) For each block *attack_blocks(i, j)* $\in D$, where $1 \leq i, j \leq N/8$, compute the authentication information as in Section 2.2 and denote it as *auth*. Compute the DCT of *attack_blocks*(*i, j*). Retrieve the 5-bit information from the LSB of integer part of the five DCT coefficients as shown in Fig. (**3**), and denote it as *retv*. Follow the same order of DCT coefficients used in the embedding stage while retrieving. If $\sum_{k=1}^{5} |auth(k) - retv(k)| \neq 0$, mark *attack_blocks(i ,j)* as invalid; otherwise, mark *attack_blocks(i, j)* as valid.

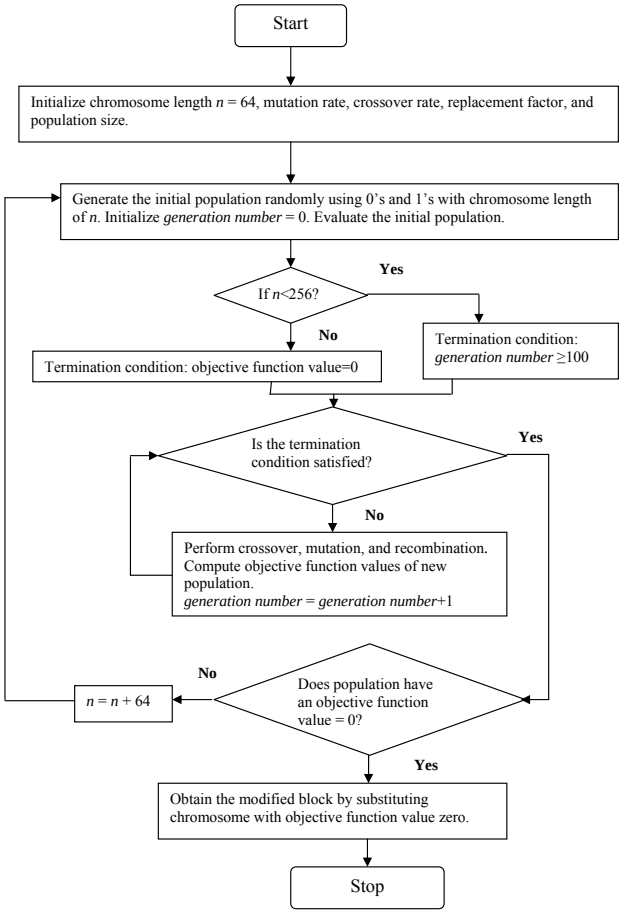

Fig. (6). The variable- length chromosome procedure.

Step$_a$ 5. (Level-2 detection) This level is used to reduce the false acceptance ratio of invalid blocks. For each valid block after level-1 detection, if it has five or more invalid blocks in its 3×3 neighborhood, mark the block as invalid; otherwise, mark the block as valid.

The proposed authentication algorithm not only can detect the tampered blocks, but also can reduce the burden on the verifier as it does not require quantization of the DCT coefficients in the authentication process. Furthermore, the two-level detection strategy can reduce the false acceptance ratio of invalid blocks.

4. EXPERIMENTAL RESULTS

To evaluate experimental results, we use the Peak Signal-to-Noise Ratio (*PSNR*) to measure the quality of the watermarked image. Let the image size be $m \times n$. The *PSNR* in decibels (db) of an image *A* with respect to an image *B* is defined by

$$PSNR = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right), \quad (4)$$

where *MSE* is the mean square error defined by

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [A(i, j) - B(i, j)]^2 . \quad (5)$$

Note that the higher the *PSNR* is, the less distortion there is to the host image and the retrieved one.

We use the False Acceptance Ratio (*FAR*) as a metric for performance evaluation of the authentication system. The *FAR* is defined as the ratio of total number of undetected blocks to the total number of attacked blocks. Its value varies in between 0 and 1. The lesser the *FAR* value, the better the performance of the authentication system is.

$$FAR = \frac{total\ number\ of\ undetected\ blocks}{total\ number\ of\ attacked\ blocks} \quad (6)$$

Fig. (7). (a) The original Lena image, (b) the watermarked image by fixed-length chromosome algorithm with *PSNR*=36.67, and (c) the watermarked image by variable-length chromosome algorithm with *PSNR*=45.12.

We use the following GA parameters in all the experiments: recombination rate=0.7 and generation gap=0.9. Fig. (**7**) shows the results of Lena images before and after the watermarking algorithms by fixed-length chromosome and variable-length chromosome with *QF* = 75, secret key *k* = 13, and clock = 2008/11/27 12:45:40. As expected, the variable-length chromosome algorithm provides better watermarked image quality than the fixed-length one. More results of the *PSNR* values by variable-length chromosome on Barbara, Baboon, and Cameraman images are listed in Table **2**.

Table 2: PSNR Values for Watermarked Images

	Lena	Barbara	Baboon	Cameraman
PSNR	**45.12**	**35.34**	**45.12**	**38.13**

We conduct CP, cropping, and VQ attacks on the watermarked image. In the CP attack [4], a portion of a given image is copied and then used to cover some other objects in the given image. If the splicing is perfectly performed, human perception would not be able to notice that identical (or virtually identical) regions indeed exist in an image. The ideal regions for the CP attack are textured areas with irregular patterns, such as grass. Because the copied areas will likely blend with the background, and it will be very difficult for the human eye to detect any suspicious artifact. Another fact which complicates the detection is that the copied regions come from the same image. They therefore have similar properties, such as the noise component or color palette. It makes the use of statistical measures to find irregularities in different parts of the image impossible. In the case of blockwise authentication systems, if the system has no blockwise dependency or unique signature that differentiates each block, then this kind of attack is not detected.

Holliman and Memon [14] presented a counterfeiting attack on blockwise independent watermarking schemes, referred to as vector quantization (VQ) or collage attack. In such a scheme the attacker generates a forgery image by the collage of authenticated blocks from different watermarked images. Because of the blockwise nature of embedding and authentication processes, the forgery image is authenticated as valid. Additionally, if the database of watermarked images is huge, the attacker can easily generate a collage image that is identical to the unwatermarked image.

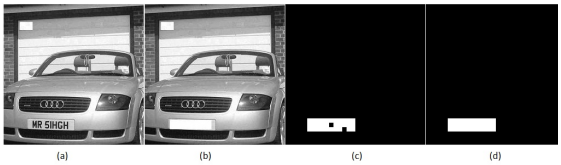

Fig. (8). (a) The watermarked image by the variable-length chromosome algorithm with *PSNR*=43.36, (b) the CP-attacked watermarked image with plate number removed, (c) the authentication image after level-1 detection, and (d) the authentication image after level-2 detection.

Fig. (**8**) shows the results of a watermarked car image along with the CP-attacked version and the authentication images after level-1 and level-2 detections. The white portion in the authentication image indicates the modified part. Fig. (**9**) shows the results of a watermarked donkey image. We also test the proposed algorithm under the cropping attack using eleven different single-chunk cropping criteria, as shown in Fig. (**10**), and their resulting *FAR* values are listed in Table **3**. Fig. (**11**) shows the resulting images under different cropping patterns. Fig. (**12**) shows the performance of fixed-length and variable-length chromosome algorithms on different images. At last, Table **4** shows the statistical comparison of variable-length chromosome algorithm with the methods in [4] and [11] on different images.

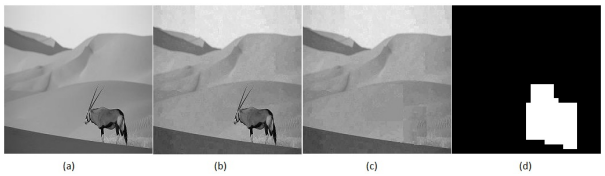

Fig. (9). (**a**) The original donkey image, (**b**) the watermarked image using variable-length chromosome algorithm with *PSNR*=41.14, (**c**) the CP-attacked image, and (**d**) the authentication image.

Table 3. The FAR Values Under Attacks Using Different Cropping Criteria in Fig. (12)

Attack_Type	*FAR*	Attack_Type	*FAR*
a	**0**	**g**	**0**
b	**0**	**h**	**0**
c	**0**	**i**	**0.038**
d	**0.0039**	**j**	**0**
e	**0**	**k**	**0**
f	**0**		

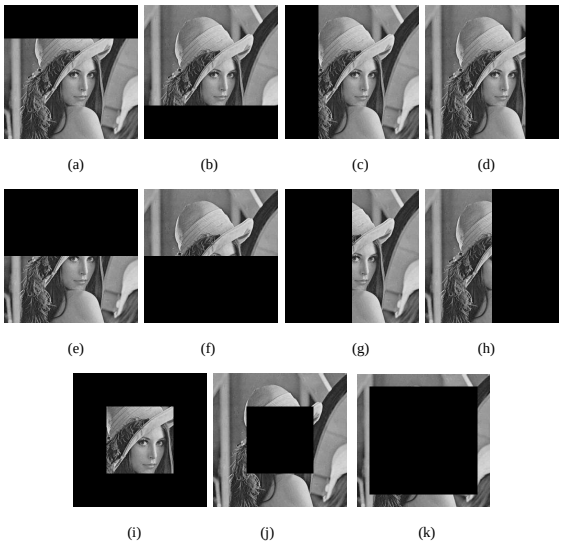

Fig. (10). Different cropping criteria: (a) top-64, (b) bottom-64, (c) left-64, (d) right-64, (e) top-128, (f) bottom-128, (g) left-128, (h) right-128, (i) outer, (j) center-25%, and (k) center-63.7%.

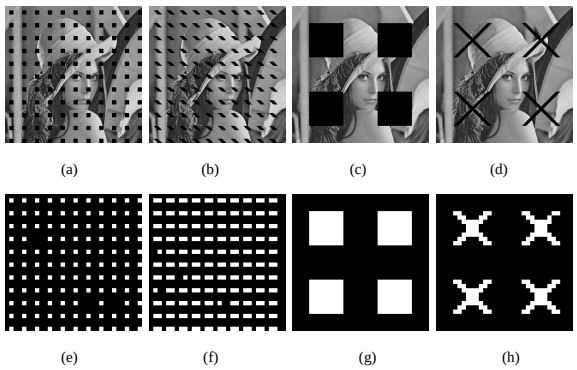

Fig. (11). (a), (b), (c), and (d) Different cropping patterns; (e), (f), (g), and (h) corresponding detection images.

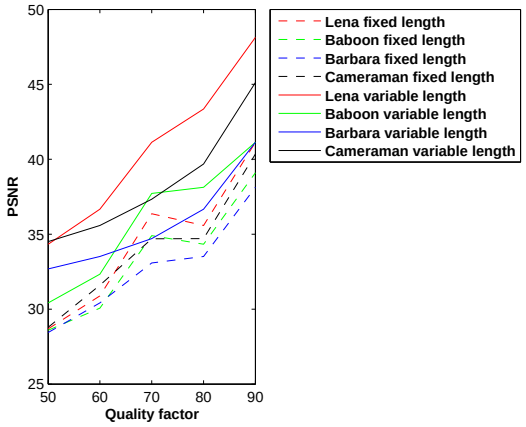

Fig. (12). Comparison of variable-length and fixed-length chromosome algorithms on different images.

Table 4. Comparison of *PSNR* Values of the Variable-Length Chromosome Algorithm with the Methods in [4], and [11]

Images	*PSNR* using variable-length chromosome		*PSNR* using [11]		*PSNR* using [4]
	QF 80	*QF* 90	*QF* 80	*QF* 90	
Lena	43.36	48.13	43.36	48.13	40.68
Barbara	36.67	41.14	36.67	41.14	40.72
Baboon	39.68	45.12	39.68	45.12	40.73

5. CONCLUSIONS

This paper presents an efficient authentication method for JPEG images based on GA, which adjusts the image so that the modified image contains authentication information in the DCT coefficients after JPEG compression. In the watermark extraction stage, the receiver obtains the authentication information without quantizing the DCT coefficients; this reduces the burden on the receiver in terms of computation and space required for storing quantization tables. In addition, with the generation of authentication information that is unique to each block and each image, the presented GA-based method is superior to the existing methods as it thwarts the VQ and CP attacks.

This paper also presents the variation of image adjustment by fixed-length and variable-length chromosome to meet different application requirements, such as time and watermarked image quality. Furthermore, the two levels of detection in the authentication procedure reduce the false acceptance ratio of invalid blocks.

REFERENCES

[1] I. J. Cox, M. L. Miller, and J. A. Bloom, *"Digital watermarking principles & practice"*, Morgan Kaufmann, 2002.

[2] M. U. Celik, G. Sharma, E. Saber, and A. M. Tekalp, "Hierarchical watermarking for secure image authentication with localization", *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 585-594, June 2002.

[3] F. Y. Shih, *"Digital watermarking and steganography: fundamentals and techniques"*, Florida: CRC Press, 2007.

[4] T. Y. Lee and S. D. Lin, "Dual watermark for image tamper detection and recovery", *Pattern Recognit.*, vol. 41, no. 11, pp. 3497-3506, Nov. 2008.

[5] P. L. Lin, C. K. Hsieh, and P. W. Huang, "A hierarchical digital watermarking method for image tamper detection and recovery", *Pattern Recognit.*, vol. 38, no. 12, pp. 2519-2529, Dec. 2005.

[6] P. L. Lin, P. W. Huang, and A. W. Peng, *"A fragile watermarking scheme for image authentication with localization and recovery"*, in Proc. IEEE Sixth Intl. Symp. Multimedia Software Engineering, Miami, FL, Dec. 2004, pp. 146-153.

[7] C. K. Ho, and C. T. Li, *"Semi-fragile watermarking scheme for authentication of JPEG images"*, in Proc. Intl. Conf. Information Technology: Coding and Computing, Las Vegas, NV, 2004, vol. 1, pp. 7-11.

[8] C. Y. Lin, and S.-F. Chang, *"Semi-fragile watermarking for authenticating JPEG visual content"*, in Proc. SPIE Security and Watermarking of Multimedia Content II, San Jose, CA, Jan. 2000, vol. 3971, pp. 140-151.

[9] J. Fridrich, *"Image watermarking for tamper detection"*, in Proc. IEEE Intl. Conf. Image Processing, Chicago, IL, Oct. 1998, vol. 2, pp. 404-408.

[10] C. T. Li, "Digital fragile watermarking scheme for authentication of JPEG images", *IEEE Proc. Vis Image Signal Process.*, vol. 151, no. 6, pp. 460-466, 2004.

[11] H. Wang, K. Ding, and C. Liao, *"Chaotic watermarking scheme for authentication of JPEG images"*, in Proc. Intl. Symp. Biometric and Security Technologies, Apr. 2008, pp. 1-4.

[12] Z. -M. Lu, D. -G. Xu, and S. -H. Sun, "Multipurpose image watermarking algorithm based on multiscale vector quantization", *IEEE Trans. Image Process.*, vol. 14, no. 6, pp. 822-831, June 2005.

[13] M. Suhail and M. Obaidat, "Digital watermarking-based DCT and JPEG model", *IEEE Trans. Instr Meas.*, vol. 52, no. 5, Oct. 2003.

[14] M. Holliman, and N. Memon, "Counterfeiting attacks on oblivious block wise independent invisible watermarking schemes", *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 432-441, Mar. 2000.

[15] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press, 1975.

[16] A. S. Tanebaum, *Computer Networks*, 4th ed. The Netherlands Pearson Education International, 2003.

[17] C. -S. Shieh, H. -C. Huang, F. -H. Wang, and H. -S. Pan, *"Genetic watermarking based on transform-domain techniques"*, *Pattern Recognit.*, vol. 37, no. 3, pp. 555-565, Mar. 2004.

[18] F. Y. Shih and Y. -T. Wu, *"Enhancement of image watermark retrieval based on genetic algorithms"*, *J. Vis. Commun. Image Represent.*, vol. 16, no. 2, pp. 115-133, Apr. 2005.

Received: December 21, 2009 Revised: February 11, 2010 Accepted: February 16, 2010