Journal of Theoretical and Applied Information Technology

www.jatit.org

COMPARISON BETWEEN TWO WATERMARKING ALGORITHMS USING DCT COEFFICIENT, AND LSB REPLACEMENT

Mona M. El-Ghoneimy
Associate Professor, Elect. & Comm. Dept., Faculty of Engineering, Cairo University, Post code 12316
E-mail: mmriad@link.net

ABSTRACT

Digital watermarking is a method through which we can authenticate images, videos and even texts. Watermarking functions are not only authentication, but also protection for such documents against malicious intentions to change such documents or even claim the rights of such documents. In this paper two watermarking algorithms are simulated. The first algorithm is based on the Discrete Cosine Transform (DCT)(in the frequency domain) and the second algorithm is based on the least significant bit (LSB)replacement (in the spatial domain). The results are shown and compared under different kinds of attacks.

Keywords: *Image Processing (I.M.), Digital Watermarking (D.W.), DCT Coefficient, LSB Replacement.*

1. INTRODUCTION

Digital watermarking is a technique which allows an individual to add hidden copyright notices or other verification messages to digital audio, video, or image signals and documents. Such a message is a group of bits describing information pertaining to the signal or to the author of the signal (name, place, etc.). The technique takes its name from watermarking of paper or money as a security measure. Digital watermarking can be a form of steganography [1], in which data is hidden in the message without the end user's knowledge.

Requirements of Image Watermarking:

An image watermarking system needs to have at least the following two components:

1. A watermark embedding system.
2. A watermark extraction (recovery) system.

The watermark embedding system takes as input the watermark bits, the image data, and optionally a secret or public key. The output of the watermark embedding system is the watermarked image.

The watermark extraction system takes as input an image that possibly contains a watermark and possibly a secret or public key. Depending on the type of watermarking system used, it may also take as input the original image or the watermark [2, 3].

2. THE FIRST PROPOSED WATERMARKING SCHEME BASED ON DCT:

Several techniques can transform an image into frequency domain, such as DCT, DFT and wavelet transform [4,5,6]. Each transform has its advantages. Here the DCT approach will be discussed.

The most common DCT definition of a 1-D sequence of length N [7, 8] is:

$$C(u) = \alpha(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{\pi(2x+1)}{2N}\right] \quad (1)$$

for u= 0,1,2,…,N− 1. Similarly, the inverse transformation is defined as

$$f(x) = \sum_{u=0}^{N-1} \alpha(u)C(u)\cos\left[\frac{\pi(2x+1)u}{2N}\right] \quad (2)$$

for x= 0,1,2,…,N− 1 . In both equations (1) and (2) α(u) is defined as

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & for \quad u = 0 \\ \sqrt{\frac{2}{N}} & for \quad u \neq 0 \end{cases} \quad (3)$$

The 2-D DCT is a direct extension of the 1-D case and is given by:

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right]$$

(4)

Where: u, v = 0,1,2,…,N −1 and. The inverse transform is defined as:

$$f(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\alpha(u)\alpha(v)C(u,v)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right]$$

(5)

Where: x, y = 0,1,2,…,N −1.

This technique embeds the watermark in the DCT domain to increase the robustness of the watermarking scheme against JPEG compression. The watermark bits are embedded in each nxn DCT block of the image. The embedding algorithm needs to carefully choose where to embed the watermark bits in the n x n block. It is not wise to embed the watermark bits in the low frequency components of the DCT block, because these coefficients are subject to heavy quantization during JPEG compression. Hence, it is better to embed the watermark in mid or high frequency DCT components. If the embedding factor M is chosen small, embedding the watermark in lowest frequency components will be more desirable, because these components are the ones that are least likely to be quantized in JPEG compression [9]. The flow chart of the algorithm is shown in figure (1).

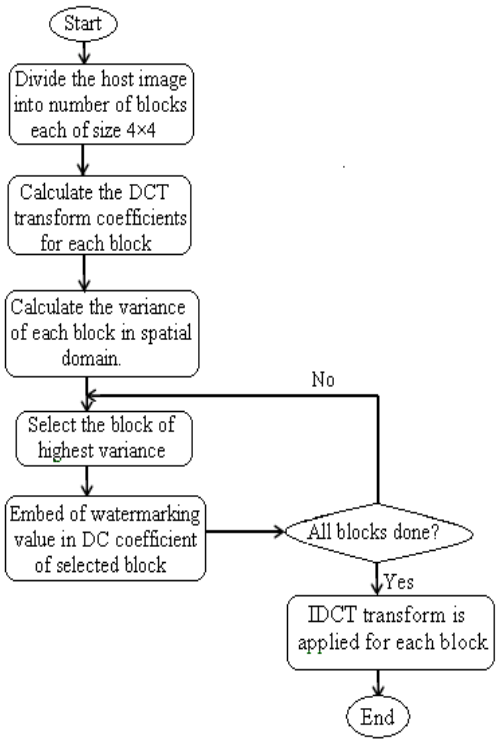


Figure (1): The DCT algorithm flow chart

2.1 Watermark Embedding Process

In this proposed approach, the embedded binary watermark image must be invisible to human eyes and robust to most image processing operations. To meet these requirements, each binary watermark pixel value (0 or 1) is embedded in one block of the host image. The Embedding algorithm can be described in following steps:

Step 1. Load the image to be watermarked (original image). The size of the original image is 512 × 512.

Step 2. Load the watermark image. The size of the watermark is 64×64 .

Step 3. The host image is divided into a number of blocks, the size of each block is 4×4 .

Step 4. Guarantee that the number of host image blocks is equal to or greater than the number of watermark pixels.

Step 5. Calculate the variance of each block in the host image in spatial domain.

Step 6. For each host image block compute the DCT transform coefficients.

Step 7. Select the DC component of blocks which has highest variance, and each watermark pixel W_q (0 or 1) is embedded in the DC component X_{dc} in order as follow:

$X'_{dc} = X_{dc} + M$ if $W_q = 1$

$X'_{dc} = X_{dc} - M$ if $W_q = 0$ (6)

Where q=1,2,3,……..rc, Where: rc= size of the watermarked image , M is the embedding watermark strength.

Step 8. After embedding the watermark, IDCT transform is applied for each block, then the watermarked image is reconstructed [10, 11].

2.2 Watermark Extraction Process:

To obtain the extracted watermark from watermarked image, the following procedure was performed:

Step 1. Original image is used for watermark retrieval, as in the embedding process, original image and watermarked image are divided into a number of 4×4 blocks.

Step 2. Calculate the DCT transform coefficients for each block in both original image and watermarked image.

Step 3. Watermark extraction process is done by comparing the DC coefficient of each two corresponding blocks with the same embedding order (of maximum block variance) as follow :

$W'_q = 1$ if $X_w - X_o \geq 0$

$W'_q = 0$ if $X_w - X_o \leq 0$ (7)

,where X_w is the DC coefficient of the watermarked image, X_o is the DC coefficient of

the original image, and W'_q is the extracted watermark pixel, then the extracted watermark will be as follow :

After extracting the watermark, the normalized cross-correlation (NCC) is calculated to evaluate the effectiveness of our scheme. The normalized cross-correlation is calculated between the original watermark W(i,j), and the extracted one W'(i,j) from this relation :

$$NCC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left[W(i,j).W^{'}(i,j)\right]}{\sum_{i=1}^{M}\sum_{j=1}^{N}\left[W(i,j)\right]^2} \qquad (8)$$

As NCC can take values from 0 to 1, and as long as NCC more closed to 1, this means that the extracted watermark is more similar to the original watermark.

2.3 Simulation Results:

2.3.1 Simulation results without attacks:

The watermarking system must embed the watermark in the image such that the visual quality of the image is not perceptibly distorted, thus, to study the embedding effect, we should calculate the peak signal to noise ratio (PSNR) from the following relation:

$$PSNR = \frac{[255]^2}{\sum_{i=1}^{M}\sum_{j=1}^{N}\left[X(i,j) - X^{'}(i,j)\right]^2} \qquad (9)$$

,as X is the original image and X' is the watermarked image. From the proposed scheme when the embedding strength M=10, the PSNR of the watermarked image without any attacks is about 46.1926 dB and the mean square error is 1.5625 which are an acceptable values. Also by calculating the NCC without any attacks it will be 1, which means that the extracted watermark is similar to the original watermark.

2.3.2 Simulation results with different attacks:

- **JPEG Compression:**

JPEG is a commonly used standard method of compression for photographic images. JPEG compression is applied with different quality factors to indicate the robustness of the proposed scheme against JPEG compression. Table (1) illustrates how NCC, mean squared error (mse), and PSNR change with different quality factors (QF) when M=10 :

From table (1), and figure (2), we find that the quality factor is directly proportional to both PSNR and NCC, as increasing the quality factor means low compression rate and more details preserved by the compression algorithm, also the watermark can be survive with quality factor down to 55% ,as in this case, the watermark can be retrieved correctly .

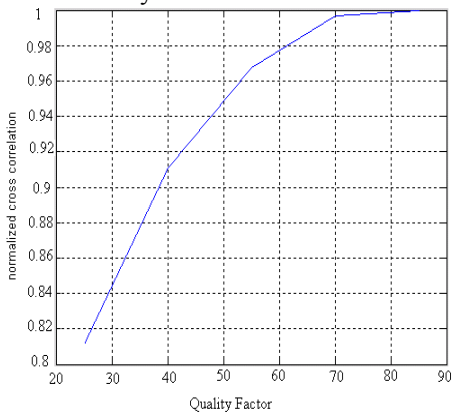


Figure (2): JPEG quality factor versus NCC

- **Gaussian Noise:**

In this section, Gaussian noise is applied over the watermarked image with zero mean and different variances, where the variance of the noise is a function of the image intensity values in the watermarked image. From table (1), and figure (3), it is found that as long as variance increases the PSNR decreases and so the NCC, also when the variance increases up to 0.001 the PSNR decreases down to 29.9212 but the extracted watermark still can be distinguished.

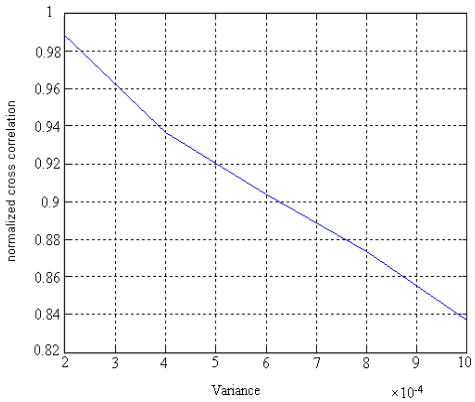


Figure (3): variance versus the NCC

- **Median Filtering:**

Reducing noise in an image by blending the

brightness of pixels within a selection, the filter searches the radius of a pixel selection for pixels of similar brightness, discarding pixels that differ too much from adjacent pixels, and replaces the center pixel with the median brightness value of the searched pixels. Median filtering is applied on watermarked image with PSNR = 31.6174 and mse = 44.8061 using 3×3 neighborhood.

- **Blur Filter (Gaussian smoothing) :**

The blur filter smoothen an image, and are useful for retouching. It smoothes transitions by averaging the pixels next to the hard edges and shaded areas in an image. The degree of smoothing is determined by the standard deviation. Here we apply Gaussian smoothing.

- **Cropping:**

In this section, a square at the image center is cropped out from the watermarked image then the watermark is extracted after cropping with different dimensions, table (1) illustrates the effect of applying cropping with different dimensions on the watermarked image.

It is clear that the NCC gradually decreases which means that this algorithm is robust against cropping, as the recognition of a degraded watermark is easy.

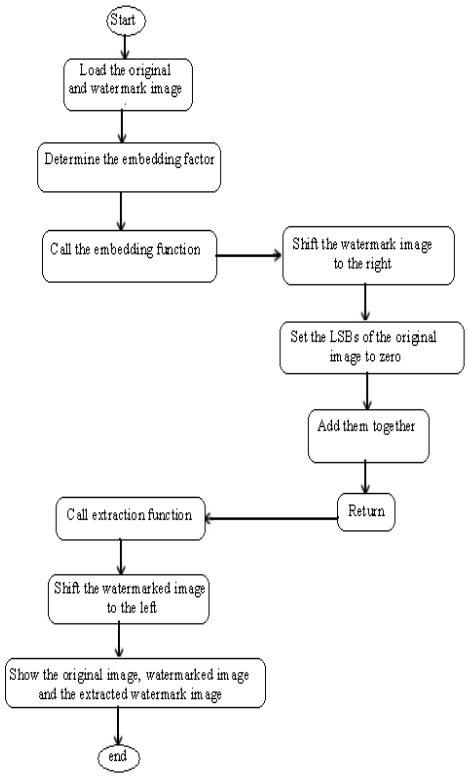
- **Global Geometrical distortion (Rotation):**

Applying a small degree of rotation on the watermarked image will lead to a full damage to the watermark information when applying a 3 degree rotation at the watermarked image center, as the resulting PSNR = 12.9709 and the NCC of the corresponding extracted watermark will be equal 0.5121.

3- THE SECOND PROPOSED ALGORITHM USING LSB REPLACEMENT

In this algorithm we embed the most significant bits of each pixel of the watermark in the least significant bits' places of the original image. The embedding of the watermark is performed choosing a subset of image pixels and substituting the least significant bit of each of the chosen pixels with watermark bits.

In the extraction we extract the most significant bits of the watermark that we embedded in the original image. Or in other words extraction of the watermark is performed by extracting the least significant bit of each of the selected image pixels. If the extracted bits match the inserted bits, then the watermark is detected. The extracted bits do not have to exactly match with the inserted bits. A correlation measure of both bit vectors can be calculated. If the correlation of extracted bits and inserted bits is above a certain threshold, then the extraction algorithm can decide that the watermark is detected. The algorithm flow chart is shown in figure (4).



Figure(4) Flow chart of LSB algorithm

The following steps show clearly how such an algorithm can be implemented:

Step 1. Load the original image.

Step 2. Load the watermarked image.

Step 3. Determine the value of the embedding factor (factor that represents how many bits of the watermark image are embedded in the LSB's of the original image).

Step 4. Call the embedding function to embed the most significant bits of the watermark image; whose number is equal to the embedding factor; in the least significant bits of the original image.

Step 5. Show the original image and the watermarked image.

Step 6. Use the extraction function to extract the watermark.

3.1 Simulation results:

3.1.1 Simulation results without attacks

From the relation we used in last section we have the value of PSNR=27.8952.

3.1.2 Simulation results with attacks:

- **JPEG compression:**

We applied JPEG compression to the image with different qualities where the quality means the amount of degradation in the image (i.e., amount of compression applied)
The resulting mse, PSNR and NCC are shown in table (2), and the relation between the quality factor and the NCC is shown in figure (5).

- **Gaussian noise:**

We applied Gaussian noise insertion with zero mean and variance=
[0.0002 0.0004 0.0006 0.0008 0.001].
Table (2) shows the values for each of NCC, mse and PSNR. The relation between the variance and the NCC is as shown in figure (6).

- **Median filtering:**

2D median filtering is applied using 3x3 neighborhoods. Table (2) shows the values for mse, PSNR and NCC after applying median filtering.

- **Blur filter (Gaussian Smoothing):**

After applying the attack the mse, PSNR and NCC values are as shown in table(2).

- **Cropping:**

The resulting mse, PSNR and NCC are shown in table (2), and the relation between NCC and block size is shown in figure (7).

- **Global geometrical distortion (Rotation):**

Rotation for the image with various angles in degrees [3 6 9 12 15] is applied as a global geometrical distortion. The resulting mse, PSNR and NCC are shown in table (2), and the relation between the angle of rotation and the NCC is as shown in figure (8).

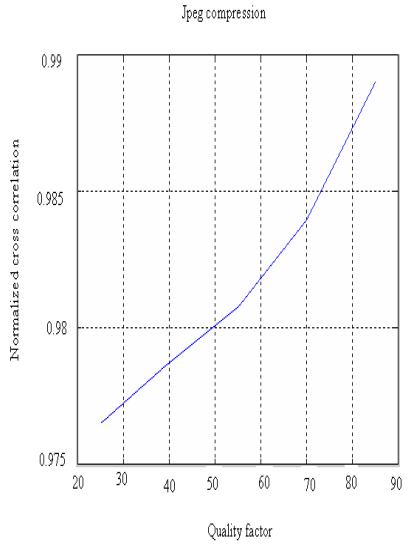


Figure (5) The relation between the NCC and the QF for JPEG compression.

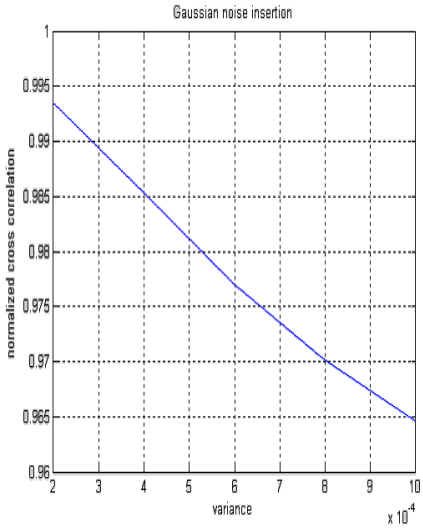


Figure (6) Results of Gaussian noise

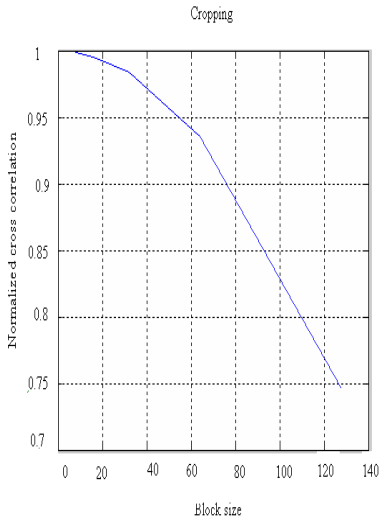


Figure (7) Results after cropping

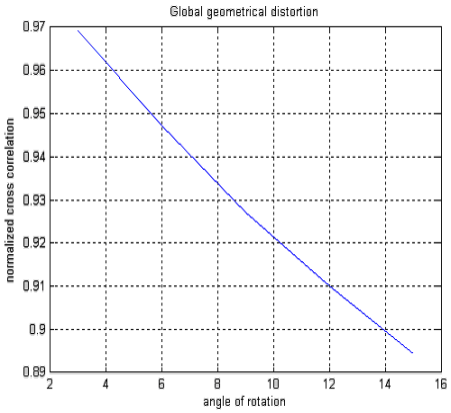


Figure (8) Results after rotation

3.2 Imperceptibility and Robustness of the Algorithm:

The visual quality of the image does not change significantly because the watermark bits only change the least significant bits of some pixels. Hence, the addition of the watermark to an image using this algorithm is quite imperceptible. On the other hand, this algorithm is not very robust, due to the same reason. As the least significant bits of pixels do not contribute to the image much, some attacker can possibly zero out several least significant bits of all pixels of the image and hence clear the watermark. This suggests that it may not be a good idea to insert the watermark bits to non significant parts of the image. This algorithm also will not be robust against JPEG compression because it is performed in the spatial domain and involves least significant bits of the image pixels. It will be shown that DCT domain based watermarking techniques are more robust to JPEG compression.

4. COMPARISON BETWEEN THE DCT ALGORITHM AND THE LSB REPLACEMENT ALGORITHM

For DCT algorithm under no attacks, the PSNR=46.1926 and NCC=1.
For LSB replacement algorithm under no attacks, the PSNR=27.8952 and NCC=1.
From tables (1), and (2) we can find that:

- With no attacks we can see that the correlation (similarity) between the watermark image and the extracted watermark is 100% for both algorithms; the DCT and the LSB; since there was no attacks (i.e no change) applied on the watermarked image. Concerning the peak signal to noise ratio, from the table its value for the DCT algorithm is higher than that of the LSB algorithm, which means that embedding a watermark using the DCT algorithm doesn't change the original image as the LSB algorithm does.

- For the median filtering, from the table we notice that the NCC value for the LSB algorithm is higher than that of the DCT algorithm.
- For the Gaussian noise insertion attack, from the table we notice that the NCC values for the LSB algorithm are higher than those of the DCT algorithm.
- For the cropping attack, from the table we notice that the values of the NCC are higher for the DCT than those of the LSB since when cropping for example 8x8 block, in the DCT algorithm we damage only 4 bits of the watermark image since we embed one pixel in each 4x4 block, but for the LSB we damage 64 bits of the watermark image since we embed one pixel of the watermark image in every pixel of the original image.
- For the global geometrical distortion, the LSB algorithm gives higher values than DCT algorithm since the rotation make the detection fail in the DCT domain by disturbing the synchronization of the DCT coefficients, as geometric attacks do not actually remove the embedded watermark itself, but intend to distort the watermark detector synchronization with the embedded information. The detector

could recover the embedded watermark information when perfect synchronization is regained, so the DCT domain technique is weak to the geometric attack, especially such as rotation since it uses only the DCT domain detection in extracting the watermark.

- For the JPEG compression attack, for the NCC, the DCT algorithm gives higher values than LSB algorithm since JPEG mainly depends on the DCT transformation.
- For the blur filtering, the LSB algorithm gives higher value for the NCC than the DCT algorithm.

REFERENCES:

[1] C.Cachin, "An Information-Theoretic Model for Steganography", Proceedings of 2nd Workshop on Information Hiding, MIT Lab. for Computer Science, May 1998

[2] R.Gonzalez and R.Woods, "Digital Image Processing", 1998

[3] Q. Zhang, Z. Ji, W. Zhu, J. Lu, and Y.-Q. Zhang, "Joint power control and source-channel coding for video communication over wireless," IEEE VTC'01, New Jersey, October 2001

[4] J.G. Cao, J.E. Fowler, and N.H. Younan, "An image-adaptive Watermark based on a Redundant Wavelet Transform", Proceedings of IEEE International Conference on Image Processing, Greece, Oct. 2001, pp.277-28

[5] C.S. Woo, J. Du, and B. Pham, "Performance factors analysis of a Wavelet-based watermarking method", Australian Information Security Workshop AISW2005, New Castle, Australia, vol. 44, 2005

[6] M. Sharkas, B . Youssef, and N. Hamdy, "An adaptive image watermarking algorithm employing the DWT", The 23rd National Radio Science Conference NRSC2006, Menoufiya, Egypt, March 2006

[7] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Transactions on Computers, vol. C-32, pp. 90-93, Jan. 1974

[8] Haque, "A Two-Dimensional Fast Cosine Transform," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-33 pp. 1532-1539, December 1985

[9] F. Hartung and M. Kutter, "Multimedia watermarking techniques," Proc. of IEEE, vol.87, pp. 1079–1107, 1999

[10] S. Liu and A. C. Bovik, "Efficient DCT-domain Blind Measurement and Reduction of Blocking Artifacts," IEEE Transactions on Circuits and Systems for Video Technology, April 2001

[11] H. Radha, M. van der Schaar, Y. Chen, "The MPEG-4 Fine-Grained Scalable Video Coding Method for Multimedia Streaming over IP," IEEE Transactions on Multimedia, March, 2001

Table (1): Results of the first algorithm:

Attack	Attack description	NCC	mse	%PSNR
JPEG Compression	QF=85	1	12.8356	80.2
	QF=70	0.9969	21.2970	75.44
	QF=55	0.9673	28.0442	72.85
	QF=40	0.9110	34.8381	70.81
	QF=25	0.8116	46.9147	68.01
Gaussian noise	var.=0.0002	0.9846	14.3951	79.12
	var.=0.0004	0.9443	27.2264	72.71
	var.=0.0006	0.8993	40.4381	69.41
	var.=0.0008	0.8701	53.3960	66.79
	var.=0.001	0.8393	66.2157	64.77
2D Median Filtering	3x3 neighborhood	0.6848	44.8061	68.48
Blur Effect	Gaussian	0.9188	21.8060	75.21
Cropping	8×8	0.9997	2.3058	96.34
	16×16	0.9978	6.5281	86.55
	32×32	0.9950	27.2343	73.12
	64×64	0.9796	121.4375	59.07
	128×128	0.9179	678.3977	42.90
Rotation	3 degree	0.5121	3.2809e+003	51.21

Table (2): Results of the second algorithm:

Attack	Attack description	NCC	mse.	%PSNR
JPEG Compression	QF=85	0.989	117.6819	98.3098
	QF=70	0.9839	123.1704	97.6001
	QF=55	0.9808	127.7164	97.0358
	QF=40	0.9787	135.8606	96.0734
	QF=25	0.9765	155.0862	94.0128
Gaussian noise	var.=0.0002	0.9935	118.4778	98.2048
	var.=0.0004	0.9853	130.7045	96.6758
	var.=0.0006	0.977	145.1285	95.046
	var.=0.0008	0.9702	156.7024	93.8514
	var.= 0.001	0.9646	168.5493	92.7168
2D Median Filtering	3x3 neighborhood	0.9999	226.7551	88.0984
Blur Effect	Gaussian	0.9901	156.5405	93.8675
Cropping	size 8×8	0.9988	108.3494	99.5961
	size 16×16	0.9956	142.8667	95.2906
	size 32×32	0.9834	280.6382	84.7792
	size 64×64	0.9355	856.2363	67.4126
	size 128×128	0.7461	3243.2277	46.6784
rotation	3 degrees	0.9693	1898.1442	55.0185
	6 degrees	0.9471	3006.99	47.8558
	9 degrees	0.927	3898.95	43.8116
	12 degrees	0.91	4591.9937	41.2644
	15 degrees	0.8942	5149.3768	39.4808