International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 10, Issue 11 (November 2014), PP.64-69

A New hybrid method in watermarking using DCT and AES

[1]Yasir Mustafa Wais [2]Ardm Hasseb Mohammed Ali
[1,2]Department of Electronic and Communication University of Cankaya Ankara - Turkey

Abstract:- In this paper I'm trying to make a combination between the encryption by using one of the most powerful algorithm called Advanced Encryption Standard (AES) to encrypt a secret message another word logo and then embed it in the digital image in frequency domain by using the Discrete Cosine Transform (DCT) in low frequency to increase the robustness and then applying some attacks to check it.

Index Terms:- Advanced Encryption Standard, Discrete Cosine Transform**.**

I. INTRODUCTION

Basically to protect any kind of secrete message transmitted over the network from playing on it or eavesdrops by attackers, there are two approaches have been imposed. One is a traditional one cryptography approach, which used to convert a secrete message or to unrecognizable form by using secret Key beside of algorithm. Second one is watermark which used to hide your secrete message inside digital media (images, video, audio etc.) this project am trying to make a combination between cryptography by using AES algorithm and watermark by using DCT.

II. DISCRETE COSINE TRANSFORM

The DCT is a very popular transform function used in signal processing. It transforms a signal from spatial domain to frequency domain. Due to good performance, it has been used in JPEG standard for image compression. DCT has been applied in many fields such as data compression, pattern recognition, image processing, and so on. The DCT transform and its inverse manner can be expressed as follows.

$$F(u,v)=\frac{4C(u)C(v)}{n^2}\sum_{j=0}^{n-1}\sum_{k=0}^{n-1}f(j,k)\cos[\frac{(2j+1)u\pi}{2n}]\cos[\frac{(2k+1)v\pi}{2n}], \qquad (1)$$

$$f(j,k)=\sum_{u=0}^{n-1}\sum_{v=0}^{n-1}C(u)C(v)F(u,v)\cos[\frac{(2j+1)u\pi}{2n}]\cos[\frac{(2k+1)v\pi}{2n}], \qquad (2)$$

Where

$$C(w)=1/\sqrt{2} \quad \text{when } w=0$$
$$C(w)=1 \quad \text{when } w=1,2,3,\ldots n-1$$

As an image transformed by the DCT, it is usually divided into non-overlapped m x m block. In general, a block always consists of 8x8 components. The block coefficients are shown in figure 1. The left-top coefficient is the DC value while the others stand for AC components. The zigzag scanning permutation is implied the energy distribution from high to low as well as from low frequency to mid then high frequency with the same manner.

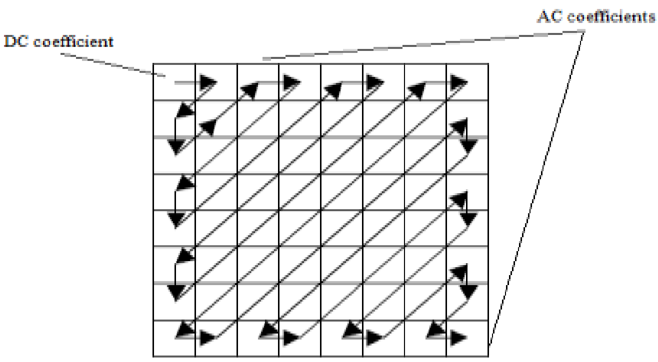


Figure 1 DCT coefficients and zigzag

The human eyes are more sensitive to noise in lower-frequency band than higher frequency. The energy of natural image is concentrated in the lower frequency range. The watermark hidden in the higher frequency band might be discarded after a lossy compression. Therefore, the watermark is always embedded in the lower-band (lower frequency) range of the host image that transformed by DCT is perfect selection.

III. DESCRIBING THE PROPOSED ALGORITHM

A. Embedding algorithm using DCT

Embedding process has been done by using cover image with size 512x512 and logo or watermark image with size 128x128 in low frequency according of below steps

1- Initial step of embedding process consist of breakout the original image NxN in to 8x8 blocks and the logo or watermark image into 4x4 blocks.
2- Make a DCT for each original and logo block.
3- Convert each original and logo block to vector form by using Zigzag order
4- Apply below embedding equation, which a_{ij} represent a block from original image , b_{ij} represent a block from logo and α represent the scaling factor.

$$w_{ij} = a_{ij} + \alpha . b_{ij}$$

5- After embedding process apply a Re-zigzag order to convert the vector to block.
6- Make inverse DCT to convert it to spatial domain
7- Repeat the steps until all blocks run out.

The figure 2 explains the embedding process for two kinds of logo first normal one and second encrypted one

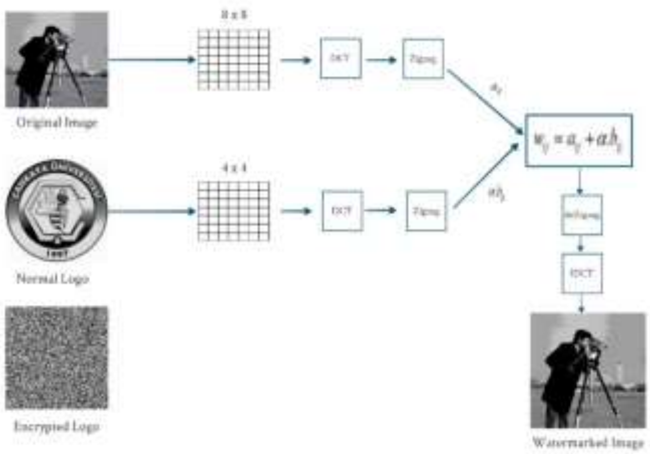


Figure 2 Embedding process

B. Extracting algorithm

Extraction process it's kind of inverse operation for embedding process which done by using cover or original image 512x512 and watermarked image with size 512x512 according of below steps

1- Breakout the watermarked image and original image into 8x8 blocks
2- Make a DCT for each watermarked and original block
3- Apply Zigzag order for each watermarked and original block to convert it to vector
4- Apply below extracting equation.

$$b_{ij} = w_{ij} - a_{ij} / \alpha .$$

5- After extracting process apply Re-zigzag order to convert vector to block
6- Make inverse DCT to convert it to spatial domain
7- Repeat the steps until all blocks run out .

The figure 3 explains the extraction process for one kind of logo the normal one same outcomes for encrypted one

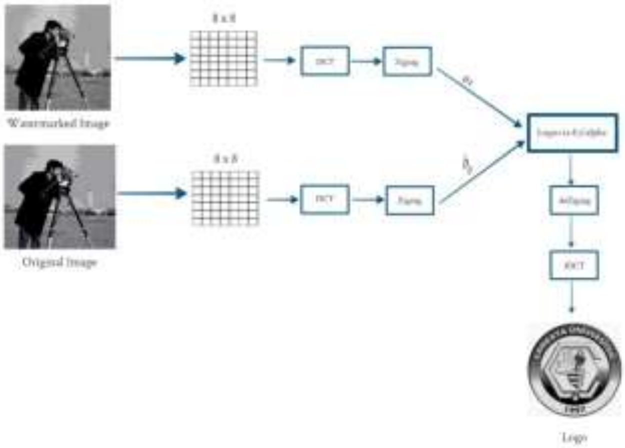


Figure 3 Extraction process

IV. ADVANCED ENCRYPTION STANDARD

In January 1997, researchers world-over were invited by National Institute of Standards and Technology (NIST) to submit proposals for a new standard to be called Advanced Encryption Standard (AES)in order to take a place of Data Encryption Standard (DES), which was published in 1977.

Thus after the selection process the two Belgian cryptographers, Joan Daemen and Vincent Rijmen, win the contest therefore some AES called the Rijndael algorithm.

In June 2003, the U.S. Government announced that AES could be used to protect classified information "Top Secret". the Rijndael algorithm support supported plaintext sizes of 128, 192 and 256 bits, as well as, key-lengths of 128,192 and 256 bits. But in my approach I used 128 bit's because of gray level image which each pixel represent 1 byte.
So to descript how the algorithm works to encrypt or decrypt an image

A. Encryption process

In encryption process am going to pick up each 4x4 logo block as a state matrix together with cipher key as a input to encryption process and repeat the operation until the blocks run out in logo image the below steps explain the encryption process for each block.

1- Key Expansion - The algorithm for generating the 10 rounds of the round key is as follows: The 4th column of the key is rotated such that each element is moved up one row.

2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

→

cf
4f
3c
09

Then puts this result through a forwards Sub Box algorithm which replaces each 8 bits of the matrix with a corresponding 8-bit value from S-Box.

cf
4f
3c
09

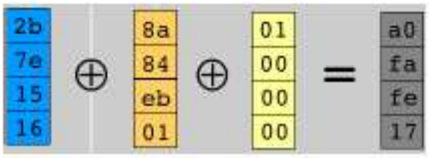
→

8a
84
eb
01

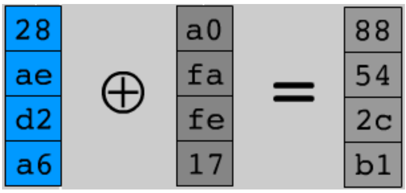
To generate the first column of the key, this result is XOR -ed with the first column of the i-1th key as well as a constant (Row constant or Rcon) which is dependent on i.

Rcon=

01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00



Then select the second column of the key and then continues iteratively for the other two columns in order to generate round key 1.



2. Initial Round
   1. Add Round Key : each byte of the state is combined with the round key using bitwise xor.
3. Rounds
   1. Sub Bytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
   2. Shift Rows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
   3. Mix Columns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
   4. Add Round Key
4. Final Round (no MixColumns)
   1. SubBytes
   2. ShiftRows
   3. AddRoundKey

B. Decryption process

It's kind of inverse process that used same 10 round keys but with the inverse SubBytes, inverse ShiftRows and inverse MixColumns.

V. EXPERIMENTAL RESULTS

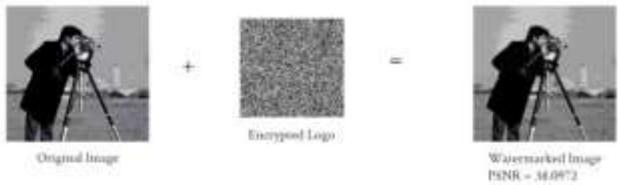
The test has been done by using some of the popular images like (Cameraman) as an original image with regards of the two kinds of logo one is encrypted one as below .

Logo Encrypted Logo

The second one is normal logo (My university logo)

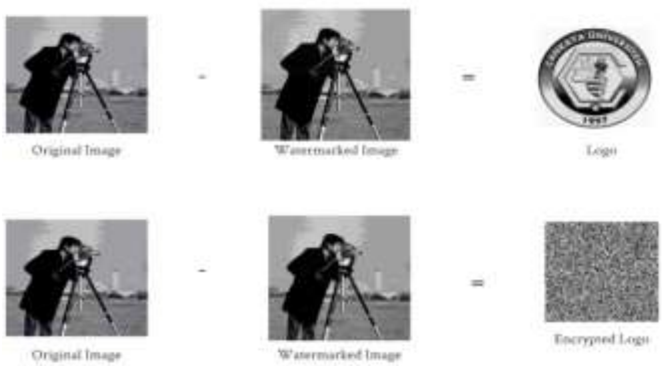
Normal Logo

To embed these two logos with scale factor 0.08 the outcomes like below with the PSNR.



To apply some attacks and then extract the logo the outcomes like below

Regarding of outcomes for extraction process like below



VI. CONCLUSIONS

The combination idea work just normal case when you extract an encrypted logo and then decrypt it but when you apply any kind of attacks this idea does not stand, take a look for extract normal logo it's contain a little bit noise depend of type of attack this will be same impact for encrypted logo and to decrypted it according to AES will be impossible to get a secret message. Therefore I think if there is noise reduction for each attack used to eliminate impacted logo and then apply decryption process or find out another encryption algorithm with can decrypt an impacted logo.

REFERENCES

[1]. A Digital Image Watermarking Method in the Discrete Cosine Transformation Domain, Mohammad Reza Khammar, Yunusa Ali Saied, M. H Marhaban .

[2]. A Digital Image Watermarking Method in the Discrete Cosine Transformation Domain, Mohammad Reza Khammar, Yunusa Ali Saied, M. H Marhaban.

[3]. Intelligent Multimedia Analysis for Security Applications edited by Husrev T. Sencar, Sergio Velastin, Nikolaos Nikolaidis, Shiguo Lian.

[4]. Digital Watermarking Using DCT Transformation, Wen Yuan Chen and Shih Yuan Huang.

[5]. A DCT-domain system for robust image watermarking Mauro Barni, Franco Bartolini, Vito Cappellini, Alessandro Piva.

[6]. http://en.wikipedia.org/wiki/Advanced_Encryption_Standard.

[7]. Understanding AES Mix-Columns Transformation Calculation , Kit Choy Xintong.