Selective Image Encryption Using DCT with Stream Cipher

Sapna Sasidharan
TIFAC CORE in Cyber Security
Amrita Vishwa Vidyapeetham
Coimbatore, India
sapnapv@gmail.com

Jithin R
TIFAC CORE in Cyber Security
Amrita Vishwa Vidyapeetham
Coimbatore, India
jithinr550@gmail.com

Abstract—Encryption is used to securely transmit data in open networks. Each type of data has its own features; therefore different techniques should be used to protect confidential image data from unauthorized access. In this paper, selective image encryption using DCT with Stream Cipher is done. In the DCT method, the basic idea is to decompose the image into 8×8 blocks and these blocks are transformed from the spatial domain to the frequency domain by the DCT. Then, the DCT coefficients correlated to the lower frequencies of the image block are encrypted using the RC4 Stream Cipher. The resulted encrypted blocks are shuffled using the Shuffling Algorithm. Selective encryption is a recent approach where only parts of the data are encrypted to reduce the computational requirements for huge volumes of images.

Keywords- DCT; Stream Cipher; Shuffling Algorithm; Selective Encryption

I. INTRODUCTION

Currently, information security is becoming more essential in data storage and transmission. Images are widely used in several processes. Therefore, the protection of image data from unauthorized access is important. Image encryption plays a significant role in the field of information hiding. Image hiding or encrypting methods and algorithms range from simple spatial domain methods to more complicated and reliable frequency domain [1] ones. It is argued that the encryption algorithms, which have been originally developed for text data, are not suitable for securing many real-time multimedia applications because of large data sizes. A major recent trend is to minimize the computational requirements for secure multimedia distribution by "selective encryption" where only parts of the data are encrypted. Selective encryption aims at avoiding the encryption of all bits of a digital image and yet ensuring a secure encryption. The key point is to encrypt only a small part of the bit stream to obtain a fast method [2].

Several selective encryption methods have been proposed for DCT compressed images. Droogenbroeck and Benedett [3] selected AC coefficients from compressed images for encryption. In their method the DC coefficients are not ciphered because they carry important visible information and they are highly predictable. The compression and encryption stages are separated in this approach and this requires an additional operating cost. Fitch et al. [4] have proposed a partial image encryption where the data are organized in a scalable bitstream form [5]. The variety of applications for secure multimedia requires either full encryption or selective encryption. However, there is a huge spectrum of applications that demands security on a lower level, as for example that ensured by selective encryption (SE). Such approaches reduce the computational requirements in networks with diverse client device capabilities. The goal of SE is to encrypt a well defined range of parameters or coefficients. The security level of SE is always lower when compared with the full encryption. However, SE decreases the data size to be encrypted and consequently requires lower computational time. Confidentiality is very important for lower powered systems such as for example wireless devices. Always, when considering image processing applications on such devices we should use minimal resources. However, the classical ciphers are usually too slow to be used for image and video processing in commercial low powered systems. The selective encryption (SE) can fulfill the application requirements without the overhead of the full encryption. In the case of SE, only the minimum necessary data are ciphered [6]. In [7] a technique was proposed, called zigzag permutation applicable to DCT-based videos and images. On one hand this method provides a certain level of confidentiality, while on the other hand it increases the overall bit rate. Combining SE and image/video compression using the set partitioning in hierarchical trees was used in [8]. However, this approach requires a significant computational complexity. A method that does not require significant processing time and which operates directly on the bit planes of the image was proposed in [9]. An approach that turns entropy coders into encryption ciphers using statistical models was proposed in [10]. In [11] it was suggested a technique that encrypts a selected number of AC coefficients. The DC coefficients are not ciphered since they carry important visual information and they are highly predictable. In spite of the constancy in the bit rate while preserving the bit stream compliance, this method is not scalable. Moreover, the compression and the encryption process are separated and consequently the computational complexity is increased [6].

II. DISCRETE COSINE TRANSFORM

The DCT is a mathematical transformation that takes a signal and transforms it from spatial domain into frequency domain. Many digital image and video compression schemes use a block-based DCT, because this algorithm minimizes the amount of data needed to recreate a digitized image. In particular, JPEG and MPEG use the DCT to concentrate image information by removing spatial data redundancies in two-dimensional images [12]. In the standard JPEG encoding, the representation of the colors in the image is converted from RGB to YCbCr, then the image is decomposed in 8×8 blocks, these blocks are transformed from the spatial to the frequency domain by the DCT. Then, each DCT coefficient is divided by its corresponding constant in a standard quantization table and rounded down to the nearest integer. After this step, the DCT quantized coefficients are scanned in a predefined zigzag order to be used in the final step, the lossless compression as illustrated in Fig. 1. In each block the 64 DCT coefficients are set up from the lowest upper left corner) to the highest frequencies (lower right corner) [13].

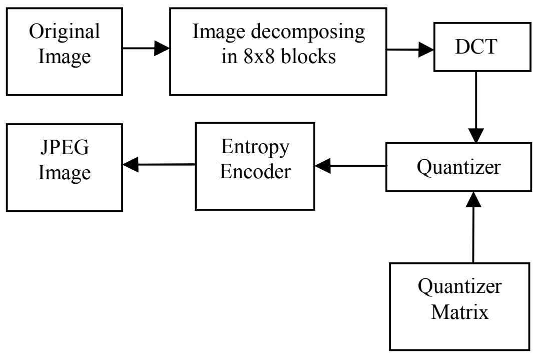

Fig. 1: JPEG Compression Algorithm

III. STREAM CIPHER

RC4 is a stream cipher, symmetric key algorithm. The same algorithm is used for both encryption and decryption as the data stream is simply XORed with the generated key sequence. The key stream is completely independent of the plaintext used. It uses a variable length key from 1 to 256 bit to initialize a 256-bit state table. The state table is used for subsequent generation of pseudo-random bits and then to generate a pseudo-random stream which is XORed with the plaintext to give the ciphertext.

The algorithm can be broken into two stages: initialization, and operation. In the initialization stage the 256-bit state table, **S** is populated, using the key, **K** as a seed. Once the state table is setup, it continues to be modified in a regular pattern as data is encrypted.

The initialization process can be summarized by the pseudo-code:

```
j = 0;
for i = 0 to 255:
S[i] = i;
for i = 0 to 255:
j = (j + S[i] + K[i]) mod 256;
swap S[i] and S[j];
```

It is important to notice here the swapping of the locations of the numbers 0 to 255 (each of which occurs only once) in the state table. The values of the state table are provided. Once the initialization process is completed, the operation process may be summarized as shown by the pseudo code below;

```
i = j = 0;
for (k = 0 to N−1) {
i = (i + 1) mod 256;
j = (j + S[i]) mod 256;
swap S[i] and S[j];
pr = S[ (S[i] + S[j]) mod 256]
output M[k] XOR pr
}
```

Where M [0..N−1] is the input message consisting of N bits. This algorithm produces a stream of pseudo-random values. The input stream is XORed with these values, bit by bit. The encryption and decryption process is the same as the data stream is simply XORed with the generated key sequence.

IV. THE PROPOSED METHOD

In the proposed method, a comparative study of selective image encryption using DCT with Stream Cipher is done. In the DCT method, the basic idea is to decompose the image into 8×8 blocks and these blocks are transformed from the spatial domain to the frequency domain by the DCT. Then, the DCT coefficients correlated to the lower frequencies of the image block are encrypted using the RC4 Stream Cipher. The concept behind encrypting only some selective DCT coefficients (the coefficients [0,0], [0,1], [0,2], [1,0], [2,0], [1,1]) is based on the fact that the image details are situated in the lower frequencies and the human is most sensitive to the lower frequencies than to the higher frequencies. An extra security has been provided to the resulted encrypted blocks by shuffling the resulted blocks using the Shuffling Algorithm. Fig. 4 shows the general block diagram of the proposed method of selective image encryption.

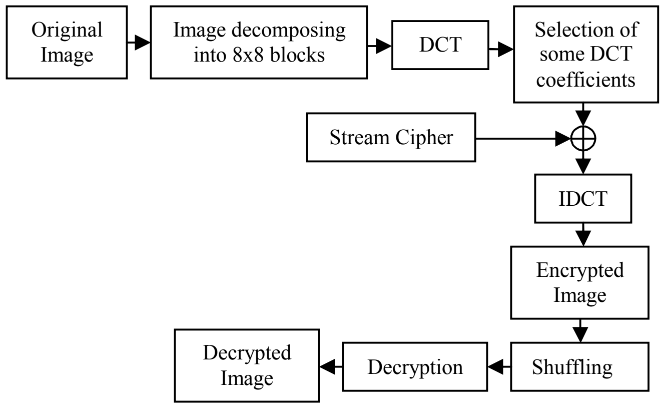

Fig. 4: Block Diagram of the Proposed Method

In the following, the encryption, decryption and shuffling of the images are illustrated.

Algorithm to Encrypt Image

Input : Target Image to be encrypted and the stream RC4 Key values.

Output: Encrypted Image

Begin

Step 1: Read the image header, save the height of the image in variable height & the width in variable width and save the body image in an array imagbody.

Step 2: Obtain how many blocks exist in an image row and how many ones in the column, by dividing the width and height of the image by N, where N is equal to 8 (the required block size).

NoRowB = Image Height / N;

NoColB = Image Width / N;

Step 3: For all blocks in the image perform the following:

- Get_block (row_no, col_no)
- Perform a DCT on the block and save the resulted coefficients in an array.
- Round the selected coefficients, convert the selected coefficients to 11 bits; the 12th bit is used to save the sign of the coefficient.
- Encrypt the selected coefficients by XORing the generated bit stream from the RC4 + Key with the coefficient bits, the sign bit of the selected coefficients will not be encrypted.
- Perform an Inverse Discrete Cosine Transform (IDCT) and get the new block values and the resulted values could be positive or negative values due to the encryption step.

Step 4: Apply the proposed shuffling algorithm on the resulted blocks to obtain the encrypted image.

End

Algorithm to Decrypt Image

Input : Target Image to be decrypted and the Encryption Key

Output: Original Image

Begin

Step 1: Read the image header, save the height of the image in variable height & the width in variable width and save the body image in an array imagbody.

Step 2: Obtain how many blocks exist in an image row and how many ones in the column, by dividing the width and height of the image by N, where N is equal to 8 (the required block size).

NoRowB = Image Height / N;

NoColB = Image Width / N;

Step 3: For all blocks in the image perform the following:

- Get_block (row_no, col_no)
- Perform a DCT on the block and save the resulted values in an array.
- Round the selected coefficients, convert the selected coefficients to 11 bits; the 12th bit is used to save the sign of the coefficient.
- Decrypt the resulted bits by using the generated bit stream from the RC4 + Key, by performing an XOR operation, the sign bit of the selected coefficients will remain.
- Convert the resulted bits into integer values, and join the sign (from the step above) with each integer, if the coefficient is negative multiply it by -1.
- Perform an IDCT and get the new blocks.

Step 4: Reshuffle the block, since the shuffling algorithm generates the same row and column numbers to return the shuffled blocks into their original locations.

Step 5: Reconstruct the image to get the original Image.

End

Shuffling Algorithm

Input : Key, number of blocks in the row (**NoRows**), number of blocks in the column (**NoCols**) and the resulted encrypted image saved in an array.

Output: A new shuffled image

Begin

```
        for i = 0 to (NoRows × NoCols)
        NewVal [i] = (Key × i) mod (NoRows × NoCols)
        endfor
        k = 0
        for i = 0 to (NoRows × NoCols)
        MoveBlock (ImageBlk (NewVal [i]), ImageBlk [k])
        k++
        endfor
    End
```

V. EXPERIMENTAL RESULTS

The performance analysis of selective image encryption using DCT with Stream Cipher is measured using the Peak Signal to Noise Ratio (PSNR), histogram analysis and entropy. Fig. 5.1 shows the Original Image used in the DCT method. Fig. 5.2 shows the Selective Encryption of the original image. The Encrypted Image after applying the shuffling algorithm is shown in Fig. 5.3 and in Fig 5.4, the Decrypted Image is shown. Here, the number of coefficients/ block selected is 6.

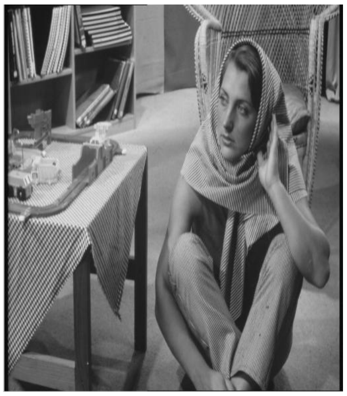
Fig. 5.1: Original Image

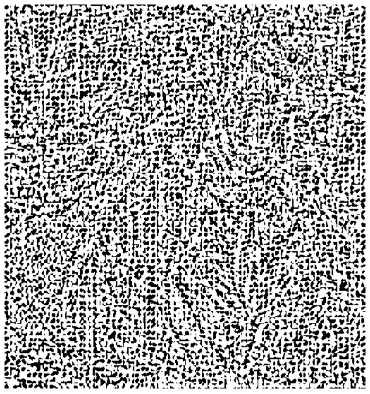
Fig. 5.2: Selective Encryption

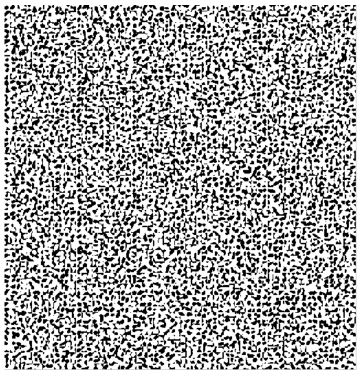
Fig. 5.3: Encrypted Image
(After Shuffling)

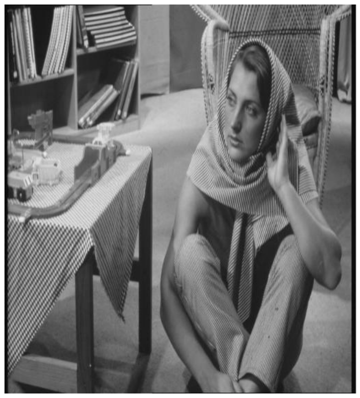
Fig. 5.4: Decrypted Image

Table I shows the Performance Analysis of the DCT method. When the number of coefficients/block selected is 6, we obtain a lower PSNR in the case of Encrypted Image and a higher PSNR in the case of Decrypted Image. Higher PSNR value shows a better quality of the image.

Table I

Performance Analysis of DCT Method

Number of coefficients/ block	PSNR of Encrypted Image	PSNR of Decrypted Image
3	32.5619	52.9979
6	29.2989	75.0756
10	29.1731	74.9530
15	28.9983	74.8003

Table II shows the performance analysis of encrypted and decrypted images in terms of PSNR when tested with different test images of size 512×512. A lower PSNR in obtained in the case of Encrypted Images and a higher PSNR is obtained in the case of Decrypted Image. Higher PSNR value shows better quality of the images.

Table II

Performance Analysis of DCT Method with different test images

Test Images	PSNR of Encrypted Image	PSNR of Decrypted Image
Barbara	20.5784	65.6641
House	20.7056	65.4996
Lena	20.8768	65.5393
Airplane	20.6219	65.4215
Baboon	20.7354	65.3072

To demonstrate that our proposed algorithm has strong resistance to statistical attacks, test is carried out on the histogram of enciphered image. Several gray-scale images of size 512×512 are selected for this purpose and their histograms are compared with their corresponding ciphered image. One typical example is shown below. The histogram of the original image contains large spikes as shown in Fig. 5.5 but the histogram of the cipher image as shown in Fig. 5.6, is more uniform. It is clear that the histogram of the encrypted image is, significantly different from the respective histogram of the original image and bears no statistical resemblance to the plain image. Hence statistical attack on the proposed image encryption procedure is difficult.

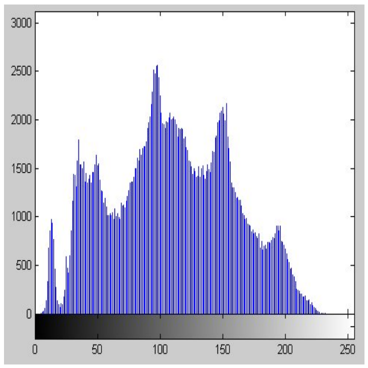

Fig. 5.5: Histogram of Original Image

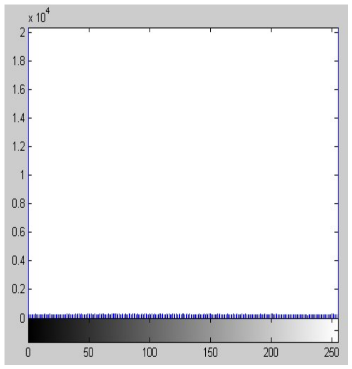

Fig. 5.6 Histogram of Encrypted Image (after shuffling)

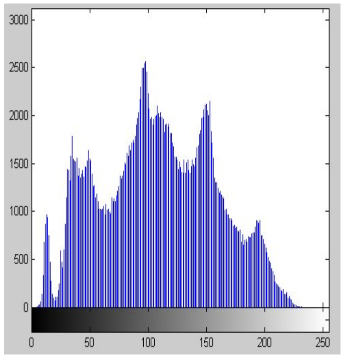

Fig. 5.7 Histogram of Decrypted Image

Entropy is a statistical measure of randomness. Table III shows the entropy of different test images of size 512×512.

Table III

Entropy of different test images

Test Images	Entropy of Encrypted Image
Barbara	3.9693
House	3.9525
Lena	3.9654
Airplane	3.9571
Baboon	3.9470

VI. CONCLUSION

The proposed encryption method uses the Selective Encryption approach where the DC coefficients and some selective AC coefficients are encrypted, hence the DC coefficients carry important visual information, and it's difficult to predict the selective AC coefficients, this give a high level of security in comparison with methods mentioned above. The algorithm will not encrypt bit by bit the whole image but only selective DCT coefficients will be encrypted, and extra security has been added to the resulted encrypted blocks by using Shuffling method. The algorithm considered as a fast image encryption algorithm, due to the selective encryption of certain portion of the image (the DC and some AC coefficients). PSNR values of the encrypted images are low and are resistant to statistical attacks. Hence, better security has been provided.

REFERENCES

[1] Lala Krikor, Sami Baba, Thawar Arif, Zyad Shaaban, “Image Encryption Using DCT and Stream Cipher”, European Journal of Scientific Research, ISSN 1450-216X, Vol.32, No.1 (2009), pp.47-57.

[2] Xiliang Liu, “Selective Encryption of Multimedia Content in Distribution Networks: Challenges and New Directions”, Proceedings of Communications, Internet, and Information Technology (CIIT 2003), Scottsdale, AZ, USA, Nov. 2003.

[3] M. Van Droogenbroeck and R. Benedett, “Techniques for a Selective Encryption of Uncompressed and Compressed Images”, in Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS) 2002, Ghent, Belgium, Sept. 2002.

[4] M. M. Fisch, H. Stgner, and A. Uhl, “Layered Encryption Techniques for DCT-Coded Visual Data”, in European Signal Processing Conference (EUSIPCO) 2004, Vienna, Austria, Sep., 2004.

[5] Rodrigues, J.M. Puech, W. Bors, A.G. "Selective Encryption of Human Skin in JPEG Images", IEEE International Conference on Image Processing, 2006.

[6] Puech, W.; Rodrigues, J.M.; Bors, A.G., "Analysis and Cryptanalysis of a Selective Encryption Method for JPEG Images", Eighth International Workshop on Image Analysis for Multimedia Interactive Services, 2007. WIAMIS07, June 2007.

[7] L. Tang., “Methods for Encrypting and Decrypting MPEG Video Data Efficiently”, In Proc. ACM Multimedia, volume 3, pages 219–229, 1996.

[8] H. Cheng and X. Li., “Partial Encryption of Compressed Images and Videos”, IEEE Trans. on Signal Processing, 48(8):2439–2445, Aug. 2000.

[9] R. Lukac, K. Plataniotis, “Bit-Level Based Secret Sharing for Image Encryption”, Pattern Recognition, 38(5):767–772, May 2005.

[10] C. Wu, C. Kuo. “Design of Integrated Multimedia Compression and Encryption Systems”, IEEE Trans. on Multimedia, 7(5):828–839, Oct. 2005.

[11] M. V. Droogenbroeck, R. Benedett, “Techniques for a Selective Encryption of Uncompressed and Compressed Images”, In Proc. of Advanced Concepts for Intelligent Vision Systems (ACIVS) 2002, Ghent, Belgium, pages 90–97, Sept. 2002.

[12] C. Coconu, V. Stoica, F. Ionescu, and D. Profeta, "Distributed Implementation of Discrete Cosine Transform Algorithm on a Network of Workstations", Proceedings of the International Workshop Trends & Recent Achievements in IT, Romania, pp. 116-121, May 2002.

[13] JPEG, jpeg.org.