A Hybrid Lossy Compression Using 2-D Discrete Cosine Transform and Visual Cryptographic Technique for Security of Multimedia Image Data Communications in Internet of Things

Quist-Aphetsi Kester[1,2,3,4(✉)]

[1] Ghana Technology University, Accra, Ghana
[2] Lab-STICC (UMR CNRS 6285), University of Brest, Brest, France
kester.quist-aphetsi@univ-brest.fr
[3] Satellite Communications and Navigation Systems Department, CRITAC, Accra, Ghana
[4] Directorate of Information Assurance and Intelligence Research, CRITAC, Accra, Ghana

Abstract. Security in internet of things is very critical. The computer systems that use image acquisition and processing from onboard or remote systems need to secure image data over the internet. Security of multimedia image data transmitted over secured and unsecured communications channel in today's cyberspace is of paramount consideration due to malicious activities over these channels and there has been a high demand especially for end to end security applications for mobile devices. This is to ensure safety and security as well as privacy to transmitted data and stored data in the clouds. In this paper, we proposed a hybrid approach for securing digital images for devices by engaging lossy compression using 2-D discrete cosine transform and Visual Cryptographic Technique. The DCT was used for the lossy compression process and the visual cryptography was used to encrypt the image for confidentiality. The results showed to be very effective and the implementation was done using MATLAB.

Keywords: Internet of things · Digital images · Security · Encryption · Compression · Discrete cosine transforms

1 Introduction

Communications of multimedia data over secured and unsecured channels require encryption techniques in securing its content. Over the decade, as computer processing power increase more live video transmission and streaming has also increased over the internet. These have led to tremendous efforts and many technical innovations in supporting real-time video streaming. Cost-effective large-scale video broadcast has remained an elusive goal. Internet protocol (IP) multicast represented an earlier attempt to tackle this problem but failed largely due to concerns regarding scalability, deployment, and support for higher level functionality [1]. These live data transmission over secured and unsecured communications channels have also call for concerns on

T.F. Bissyande and O. Sie (Eds.): AFRICOMM 2016, LNICST 208, pp. 292–303, 2018.
https://doi.org/10.1007/978-3-319-66742-3_28

data security as malicious activities by hackers over the internet pose a lot of challenge to ensuring privacy and security for its users [2]. Disparate devices accessing one data source poses a lot of challenges to data compression servers since different data formats have to be made available for easy access and effective interoperability between the various disparate devices. Normally, data is compressed before encrypted for transmission but in our approach, we proposed a situation in which we provided a two security layer for the data by first engaging pixel displacement approaches to visually conceal the content of the data before compression using discrete cosine transform.

2 Review

Image compression is of key importance in transmission of images. This helps in saving bandwidth in data communications as well as reducing the rate of power usage at terminals during transmission from one point to the other. Compressed data stored on storage devices also help saves storage space cost. Security to compressed data provides confidentiality and maintains the integrity of the transmitted data until it is verified or authenticated by the receiver. In the protection of the privacy of original content of images, Qian et al. in their work of JPEG encryption for image rescaling in the encrypted domain proposed a novel protocol of encrypting the JPEG image suitable for image rescaling in the encrypted domain. In their approach, the image owner perturbs the texture and randomizes the structure of the JPEG image by enciphering the quantized Discrete Cosine Transform (DCT) coefficients. After receiving the ciphered JPEG image by a service provider, the service provider generates a rescaled JPEG image by down-sampling the encrypted DCT coefficients. They achieved this by engaging an encryption key in the process. Upon receiving the sent data via the service provider to the recipient, the recipient on the other side, the encrypted JPEG image rescaled by the service provider can then be decrypted to a plaintext image with a lower resolution with the aid of encryption keys by the receiver. Qian et al. in their work [3] of JPEG encryption for image rescaling but also provided privacy and security [3]. In securing digital videos and images, several cryptographic techniques have been developed to encrypt images and videos in recent years. Some of the encryption schemes are based on the principles of cryptography that are designed to ensure the confidentiality of multimedia data [4–7]. In most cases, such solutions may not be suitable for the effective real-time secure applications since their encryption processes require vast resources including computing complexity, time and power. Yicong Zhou et al. in their work of encryption using Discrete Parametric Cosine Transform introduced a new effective image encryption algorithm based on the Discrete Parametric Cosine Transform (DPCT). Their new algorithm transforms images into the frequency domain using the DPCT with a set of parameters, and then converts images back into the spatial domain using the inverse DPCT with a different set of parameters to obtain the encrypted images [8]. Their new algorithm transforms the original images into the frequency domain using the new 2D DPCT with the parameters P with This is shown in the Fig. 1 below. It then uses an inverse 2D DPCT with different parameters P′ to convert the images back into the spatial domain to obtain the encrypted images. Its security keys are the combination of the parameters of the DPCT and inverse DPCT.

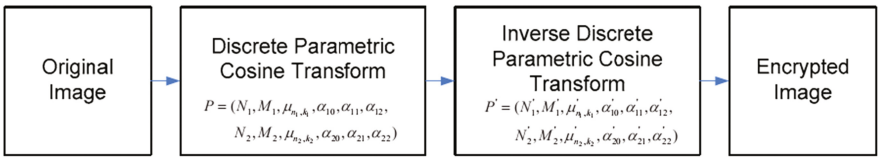

Fig. 1. Block diagram of the image encryption algorithm. In the above diagram

The experimental results of their procedure showed that the algorithm can fully or partially encrypt different types of digital images with efficiency while preserving the quality of the images. The Discrete Cosine Transform is widely used for compression of digital images and it possesses a good energy compaction property. These properties made it suitable for coding and compression, feature extraction, filtering, and image encryption. The Discrete Cosine Transform approaches for encryption schemes have been developed to protect multimedia data in the frequency domain by encrypting the DCT coefficients/blocks [5], the quantization table [9], or Huffman table [10]. These approaches significantly lower computational cost, has the encryption speed for multimedia contents in real-time applications such as mobile computing and server-end computing [11]. The Discrete Cosine Transform based encryption methods are usually combined with the data compression process; they may not be suitable for applications requiring high quality data. In our work, we proposed a hybrid approach for securing digital images for devices by engaging lossy compression using 2-D discrete cosine transform and Visual Cryptographic Technique. The DCT was used for the lossy compression process and the visual cryptography was used to encrypt the image for confidentiality. The results showed to be very effective and the implementation was done using MATLAB.

3 Methodology

In our approach we combined lossy compression technique using 2-D Discrete Cosine Transform and visual cryptographic technique for security of multimedia image data communications. But we first encrypt the image using the visual cryptographic

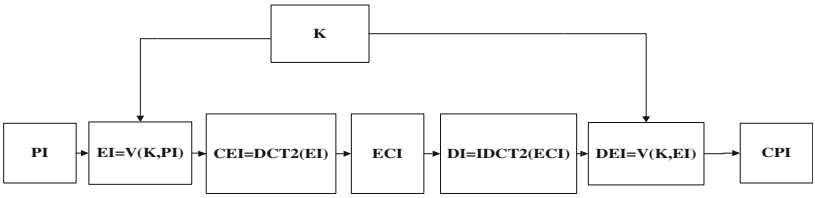

Fig. 2. Block diagram of the proposed approach: PI = Plain Image, K = Engaged key, EI = Encryption Process, V (K, PI) = the function which is the image encryption process that engages the key, and the plain image in the ciphering process. CEI = Compression of the encrypted Image process, DCT2 (EI) = the 2-D Discrete Cosine Operation on the Encrypted Image, ECI = Encrypted but Compressed Image, DI = Decompression of the compressed encrypted image, IDCT2 = Inverse of the 2-D Discrete Cosine Operation on the Encrypted Image, DEI = Decrypted decompressed ciphered Image, V (K, EI) = the function that operated on the decompressed image and the encrypted, CPI = Compressed but decrypted image

technique first, which is based on pixel shuffling and displacement, before finally applying the 2-D Discrete Cosine Transform. The Fig. 2 below illustrates the procedure.

There was no pixel loss during the encryption process due to the pixel displacement technique engaged. This means that there was pixel conservation in the process. But there was pixel loss during the compression and decompression process. The decrypted image at the end was good since critical features were still visible. Below is the explanation of the processes engaged, that is the image encryption and the 2-D discrete cosine transform process. Some image values were cut-off during the engagement of the DCT process.

3.1 The Image Encryption Process

Step 1. Start
Step 2. Extraction of data from a plain image,
Let I = an image = f (R, G, B)
I is a color image of m × n × 3 arrays

$$\begin{pmatrix} R & G & B \\ r_{i1} & g_{i2} & b_{i3} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ r_{n1} & g_{n2} & b_{n3} \end{pmatrix} \tag{1}$$

(R, G, B) = m x n
Where R, G, B ∈ I
(R o G) i j = (R) ij. (G) ij
Where r_11 = first value of R
r= [ri1] (i=1, 2… m)
x ∈ r_i1 : [a, b]= {x ∈ I: a ≤ x ≥ b}
a=0 and b=255
R= r= I (m, n, 1)
Where g_12 = first value of G
g= [gi2] (i=1, 2... m)
x ∈ g : [a, b]= {x ∈ I: a ≤ x ≥ b}
a=0 and b=255
G= g= I (m, n, 1)
And b_13 = first value of B
g= [bi3] (i=1, 2... m)
x ∈ b_i1 : [a, b]= {x ∈ I: a ≤ x ≥ b}
a=0 and b=255
B=b= I (m, n, 1
Such that R= r= I (m, n, 1)

Step 3. Extraction of the red component as 'r'
Let size of R be m × n [row, column] = size (R) = R (m × n)

$$\text{rij} = \text{r} = \text{I}\ (\text{m},\ \text{n}, 1)\ = \begin{pmatrix} R \\ r_{i1} \\ \vdots \\ \vdots \\ \vdots \\ r_{in} \end{pmatrix} \tag{2}$$

Step 4. Extraction of the green component as 'g'
Let size of G be m × n [row, column] = size (G)

$$\text{gij} = \text{g} = \text{I}\ (\text{m},\ \text{n}, 1)\ = \begin{pmatrix} G \\ g_{i2} \\ \vdots \\ \vdots \\ \vdots \\ g_{n2} \end{pmatrix} \tag{3}$$

Step 5. Extraction of the blue component as 'b'
Let size of B be m × n [row, column] = size (B) = B (m × n)

$$\text{bij} = \text{b} = \text{I}\ (\text{m},\ \text{n}, 1)\ = \begin{pmatrix} B \\ b_{i3} \\ \vdots \\ \vdots \\ \vdots \\ b_{n3} \end{pmatrix} \tag{4}$$

Step 6. Getting the size of r as [c, p]
Let size of R be [row, column] = size (r) = r (c × p)
Step 7. Engagement of K which is the symmetric secret key generated. The key is then engaged to iterate the step 8 to 14.
Step 8. Let r = Transpose of rij

$$\text{r} = \begin{pmatrix} R & & & & \\ r_{i1} & \ldots & \ldots & \ldots & r_{n1} \end{pmatrix} \tag{5}$$

Step 9. Let g = Transpose of gij

$$\mathrm{g} = \begin{pmatrix} G & & & & \\ g_{i3} & \cdots & \cdots & \cdots & g_{n3} \end{pmatrix} \tag{6}$$

Step 10. Let b = Transpose of bij

$$\mathrm{b} = \begin{pmatrix} B & & & & \\ b_{i2} & \cdots & \cdots & \cdots & b_{n2} \end{pmatrix} \tag{7}$$

Step 11. Reshaping of r into (r, c, p)

$$\mathrm{r} = \text{ reshape } (\mathrm{r},\ \mathrm{c},\ \mathrm{p}) = \begin{pmatrix} R \\ r_{i1} \\ \vdots \\ \vdots \\ \vdots \\ r_{in} \end{pmatrix} \tag{8}$$

Step 12. Reshaping of g into (g, c, p)

$$\mathrm{g} = \text{ reshape } (\mathrm{g},\ \mathrm{c},\ \mathrm{p}) = \begin{pmatrix} G \\ g_{i2} \\ \vdots \\ \vdots \\ \vdots \\ g_{n2} \end{pmatrix} \tag{9}$$

Step 13. Reshaping of b into (b, c, p)

$$\mathrm{b} = \text{ reshape } (\mathrm{b},\ \mathrm{c},\ \mathrm{p}) = \begin{pmatrix} B \\ b_{i3} \\ \vdots \\ \vdots \\ \vdots \\ b_{n3} \end{pmatrix} \tag{10}$$

Step 14. Concatenation of the arrays r, g, b into the same dimension of ‘r’ or ‘g’ or ‘b’ of the original image

$$= \begin{pmatrix} R & G & B \\ r_{i1} & g_{i2} & b_{i3} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ r_{n1} & g_{n2} & b_{n3} \end{pmatrix} \tag{11}$$

Step 15. Finally the data will be converted into an image format to get the encrypted image.

3.2 The 2-D Discrete Cosine Transform Process

The discrete cosine transform in general is related to the discrete Fourier transform. DCT is a separable linear transformation; which means that, the two-dimensional transform is equivalent to a one-dimensional DCT performed along a single dimension followed by a one-dimensional DCT in the other dimension. Below is the representation of the 2-D Discrete Cosine Transform engaged:

For a given 2-D spatial data sequence $\{X_{ij}; i, j = 0, 1, \ldots, N-1\}$, the 2-D DCT data sequence $\{Y_{mn}; m, n = 0, 1, \ldots, N-1\}$ is defined by

$$\tilde{Y}_{mn} = \frac{2}{N} E_m E_n \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{ij} \cos \frac{(2i+1)m\pi}{2N} \times \cos \frac{(2j+1)n\pi}{2N} \tag{12}$$

Where,

$$E_k = \begin{cases} 1/\sqrt{2} & \text{for } k = 0 \\ 1 & \text{otherwise.} \end{cases} \tag{13}$$

Without loss of generality, the scale factor 2E, En/N will be neglected for convenience. Thus, the 2-D DCT computation becomes

$$Y_{mn} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{ij} \cos \frac{(2i+1)m\pi}{2N} \cos \frac{(2j+1)n\pi}{2N}, \quad m, n = 0, 1, 2, \ldots, N-1. \tag{14}$$

Using the row-column decomposition and denoting cos ((2k + l)rr/2N) by C &, we have

$$Y_{mn} = \sum_{j=0}^{N-1} Z_{mj} c_{nj}, \quad m, n = 0, 1, 2, \ldots, N-1 \tag{15}$$

With

$$Z_{mj} = \sum_{i=0}^{N-1} X_{ij} c_{mi}, \quad m, j = 0, 1, 2, \ldots, N-1 \tag{16}$$

Equations (15) and (16) are generally called "column transform" and "row transform," respectively. Performing N row transforms and N column transforms to compute an N × N-point DCT N, each row transform of Eq. (16) can be written as

$$Z_{mj} = \sum_{i=0}^{N/2-1} [X_{ij} + (-1)^m X_{(N-1-i)j}] c_{mi} \tag{17}$$

Defining

$$U_{ij}^{m'} = X_{ij} + (-1)^m X_{(N-1-i)j} \tag{18}$$

where m′ = 0 when m is even and m′ = 1 when m is odd, we have

$$\begin{aligned} Z_{mj} &= \sum_{i=0}^{N/2-1} U_{ij}^{m'} c_{mi} \\ &= U_{0j}^{m'} c_{m0} + U_{1j}^{m'} c_{m1} + \cdots + U_{(N/2-1)j}^{m'} c_{m(N/2-1)} \end{aligned} \tag{19}$$

Similarly, each column transform of (15) can be expressed by

$$\begin{aligned} Y_{mn} &= \sum_{j=0}^{N/2-1} V_{ml}^{n'} c_{nj} \\ &= V_{m0}^{n'} c_{n0} + V_{m1}^{n'} c_{n1} + \cdots + V_{m(N/2-1)}^{n'} c_{n(N/2-1)} \end{aligned} \tag{20}$$

Where

$$V_{mj}^{n'} = Z_{mj} + (-1)^n Z_{m(N-1-j)} \tag{21}$$

4 Results and Analysis

The plain image in Fig. 3 below with dimension of 640 × 480 pixels is having vertical and horizon resolution of 72 dpi and bit depth of 24 (Table 1). The plain image was operated on by the proposed process and the following results were obtained from it (Figs. 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 and 14).

Fig. 3. The plain image

Table 1. Extracted features from the process.

Heading level	Entropy	Arithmetic mean
PI	7.8258	104.3025
EI	7.8258	104.3025
CEI	7.8366	1.5371
LI	2.3006	1.5310
DI	7.8366	104.3224
CPI	7.8366	104.3224

LI is the loss image

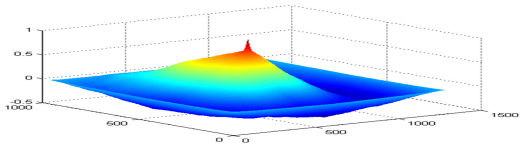

Fig. 4. The graph of the normalized cross-correlation of the matrices of the plain image

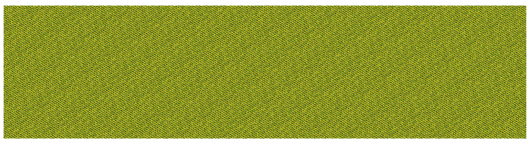

Fig. 5. The ciphered image

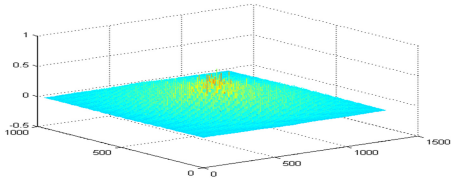

Fig. 6. The graph of the normalized cross-correlation of the matrices of the ciphered image

Fig. 7. The compressed ciphered image

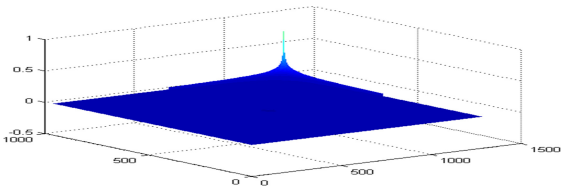

Fig. 8. The graph of the normalized cross-correlation of the matrices of the compressed ciphered image

Fig. 9. The loss image

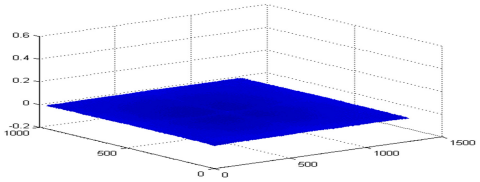

Fig. 10. The graph of the normalized cross-correlation of the matrices of the loss image

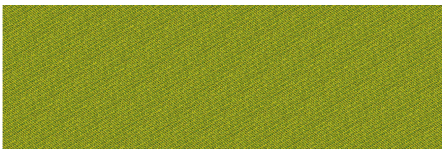

Fig. 11. The graph of decompressed image

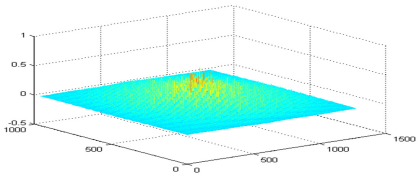

Fig. 12. The graph of the normalized cross-correlation of the matrices of the decompressed image

Fig. 13. The recovered image

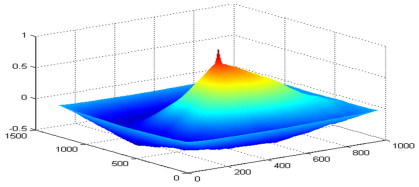

Fig. 14. The graph of the normalized cross-correlation of the matrices of the recovered image

5 Conclusion

The implementation showed to be very successful, the complexity time of the image cryptographic technique is O (N) with space complexity of O (1). The arithmetic mean and entropy of the values of the image were computed and analyzed in the table shown. From the table, it can clearly be observed that the arithmetic mean of the ciphered and plain image is the same and that of the decompressed as well as that of the recovered image are the same. The process was successful and fast and the decompressed image was successfully recovered from the ciphered image and compressed image.

References

1. Liu, J., Rao, S.G., Li, B., Zhang, H.: Opportunities and challenges of peer-to-peer internet video broadcast. Proc. IEEE **96**(1), 11–24 (2008). doi:10.1109/JPROC.2007.909921
2. Qureshi, A., Megías, D., Rifà-Pous, H.: Framework for preserving security and privacy in peer-to-peer content distribution systems. Expert Syst. Appl. **42**(3), 1391–1408 (2015)

3. Qian, Z., Zhang, X., Ren, Y.: JPEG encryption for image rescaling in the encrypted domain. J. Vis. Commun. Image Represent. **26**, 9–13 (2015)
4. Li, T., et al.: A new scrambling method based on semi-frequency domain and chaotic system. In: International Conference on Neural Networks and Brain, ICNN&B 2005, pp. 607–610 (2005)
5. Lian, S., Sun, J., Wang, Z.: A novel image encryption scheme based-on JPEG encoding. In: Proceedings of Eighth International Conference on Information Visualization, 2004, IV 2004, pp. 217–220 (2004)
6. Chen, T.-S., Chang, C.-C., Hwang, M.-S.: A virtual image cryptosystem based upon vector quantization. IEEE Trans. Image Process. **7**, 1485–1488 (1998)
7. Kester, Q., Koumadi, K.M.: Cryptographie technique for image encryption based on the RGB pixel displacement. In: 2012 IEEE 4th International Conference on Adaptive Science & Technology (ICAST), pp. 74–77, 25–27 October 2012
8. Zhou, Y., Panetta, K., Agaian, S.: Image encryption using Discrete Parametric Cosine Transform. In: 2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers, pp. 395–399, 1–4 November 2009
9. Sudharsanan, S.: Shared key encryption of JPEG color images. IEEE Trans. Consum. Electron. **51**, 1204–1211 (2005)
10. Zhou, J., et al.: Security analysis of multimedia encryption schemes based on multiple Huffman table. IEEE Signal Process. Lett. **14**, 201–204 (2007)
11. Wu, C.-P., Kuo, C.C.J.: Design of integrated multimedia compression and encryption systems. IEEE Trans. Multimedia **7**, 828–839 (2005)