ISSN 2277-3061

Secret Image Sharing Based on Discrete Cosine Transform

Ashwaq T. Hashim, Dr. Loay E. George
Dept of Computer Science, Babylon University, Collage of Science, Babil, Iraq
ashwaqtalib@yahoo.com
Dept. of Computer Science, Baghdad University, Collage of Science, Baghdad, Iraq
loayedwar57@yahoo.com

ABSTRACT

The main challenges facing secure image sharing tasks are the increase of sharing volume and sharing-control flexibility. A new scheme based on DCT is developed to perform packet secret image sharing, the DCT is used to reduce the secret image size firstly, then it employs the (k, n) threshold sharing scheme to generate the shadow images. The main signal decomposition attribute of the transform is utilized to divide the secret image into uncorrelated shares. Expermintal results are given to illustrate the characteristic of this methods.

Indexing terms/Keywords

Secret sharing, Small shadows, Secret image sharing (*k*, *n*) Threshold, DCT Transform.

Academic Discipline And Sub-Disciplines

Computer Science

SUBJECT CLASSIFICATION

Image Processing; Information Security

TYPE (METHOD/APPROACH)

Theory; Experimental Analysis

ISSN 2277-3061

INTRODUCTION

Effective and secure protection of sensitive information is the primary concern in communication systems or network storage systems. Nevertheless, it is also important for any information process to ensure data is not being tampered [1]. With the rapid advancement of network technology, multimedia information is, conveniently, transmitted over the Internet. While using secret images, security issues should be taken into consideration because hackers may utilize weak links over communication network to steal information that they want. To deal with the security problems of secret images, various image secret sharing schemes have been developed with taking into consideration the limited bandwidth and storage requirements. This contradictive requirement was firstly solved by a secret sharing scheme, which was proposed by Shamir in 1979 [2]. The ordinary secret sharing scheme separates secret information into a set of portions for participants and achieves the objective of protecting secret information. In 2002, Thien and Lin [3] extended a secret image sharing method based on Shamir's (*k, n)* threshold scheme ($k \leq n$) that shared a secret image among *n* participants, and any *k p*articipants could cooperate to reconstruct the secret image, while *k-1* or fewer participants could get nothing. Thien and lin method permutes, first, the elements of a secret image to de-correlate the pixels values and then incorporates the (*k, n)* threshold scheme to process the image pixel wise or pattern wise in the spatial domain sequentially. Each participant held his own shadow image, which contained partial information of the secret image, and the size of each shadow image was *1/k* of that of the secret image. Notably, the shadow images looked like random noise rather than ordinary images.

It is necessary to note that, most of the available image secret sharing schemes are based on spatial domain pixels operations. In 2003, Lin and Tsai [4], transformed a secret image into the frequency domain by applying the discrete cosine transform (DCT). Then all the DCT coefficients except the first 10 lower frequency ones are discarded. The values of the 2^{nd} through the 10^{th} coefficients are disarranged in such a way that they cannot be recovered without the first coefficient, this will cause the inverse DCT of them cannot reveal the details of the original image. Finally, the first coefficient is encoded into a number of shares for a group of secret sharing participants and the remaining nine manipulated coefficients are allowed to be accessible to the public. In this secret image scheme, the size of the original secret image is reduced with degrading the image quality. If higher image quality is desired, more than 10 coefficients are kept. The security of the retrieval of the original secret image by brute force attack depends only on the first coefficient C_1 which has been encrypted with the Shamir secret sharing scheme into *n* shares. In 2004, Wu et al. [5] proposed a method for sharing and hiding secret images without size expansions. The given secret image is shared and ***n*** shadow images are generated. Each shadow image is hidden in an ordinary image so as not to attract an attacker's attention. Any ***k*** of the ***n*** hidden shadows can be used to recover the secret image. The size of each stego image (in which a shadow image is hidden) is about 1/***k*** of that of the secret image, avoiding the need for much storage space and transmission time. Many secret sharing schemes have been developed for reducing the size of shares because their transmission is still a significant problem, especially in the transmission channels with limited bandwidth..In 2007, Chin and Ching [6] presented a method of image sharing based on the reversible integer-to-integer (ITI) wavelet transform. This method, works in the wavelet domain, processes the transform coefficients in each subband, divides each of the resulting combination coefficients into *n* shadows. It allows recovery of the complete secret image by using any *k* or more shadows ($k \leq n$). This method without coding has larger shadow images than those belong to the methods based on applying coding prior to inputting to the sharing phase. Also, in this method, the data is encoded either by Huffman coding or by arithmetic coding before the data is sent to the sharing phase. In 2009, Zhenfei et al. [7] proposed a method that incorporates secret image sharing with a progressive transmission that based on the integer discrete cosine transform (IntDCT). In this progressive shares transmission system, a coarse version of the secret image can be encrypted with the first part of the received shares and this coarse image can be refined through successive stages. If the decrypted image quality is good enough, shares transmission can be interrupted. In 2010, Liu et al. [8], introduced *(k, n)*-threshold scheme for image sharing based on the discrete fractional random transform (DFRNT); it is based on transform domain coefficients operations. This scheme is effective and perfectly secure due to the exploited DFRNT. However, in this scheme all shadows are of the same size as that of the secret image and ,thus, much storage space is required and more transmission time is spent. In 2011, Yang et al [9] proposed a fast secret image sharing scheme based on Haar wavelet transform and Shamir's method. They employ discrete Haar wavelet transform to reduce the secret image to its quarter size firstly (i.e., 1-level LL subband). Then, the modified Shamir's algorithm is applied only on this LL subband to generate the shadow images. In 2012, Sagar et al. [10] proposed a new way of performing color visual cryptography using wavelet technique. Wavelet technique has been used to convert the color image to gray image which is the intensity image (Y) formed from the YCbCr color transform. Then, Error-Diffusion Filter has been applied on the obtained grey image. After that the visual cryptography system (VCS) model has been applied on the generated halftone image. In 2013, Ashwaq and Loay [11], suggest a technique for sharing color image based on wavelet transform. A linear system and a random generation function have been used in construction of secret image sharing scheme. The proposed system exploits work suggested by [11] with DCT transform to built a system for sharing gray image.

PREVIOUS WORKS

In this work the (*k*, *n*) threshold secret image sharing scheme has been adapted from [11]. The image is divided into *n* number of shares and a random number generator is used, it is based on variable length key. The compressed secret information is distributed randomly into *n* shares. By using a pseudo random generator dependent on a secret key and a timestamp, the *q* segments of secret information were permuted randomly.

The following steps are taken to apply shuffling algorithm:

Algorithm 1: Randomize Function

Input: *S* // array of bytes (Secret data)

n // length of S

k // number of shares

Output: *Shares* // array of bytes (*n* shares).

Step1: Initialize *b* as a sequence of length *n*; such as

b[*i*]= *i* mod *q*, *for i=1,..,n*

Step2: Let *j*= R_1

Step3: *for i=n-1...1*

j= $(R_2 \times j + R_3) \bmod i$

swap *b*(*i*), *b*(*j*)

Where R_1, R_2 and R_3 are large prime numbers, they were generated using the key generation function suggested in [11].

After that, the secret image sharing method is used for data division. The sharing system suggested in [1] uses a set of linear equations; such that each i^{th} share has a secret set of *q* integer numbers, a_{ij} (where, $i \in [1,n]$ and $j \in [1,q]$, *n* is the total number of shares, and *q* is the minimum number of required shared to retrieve the image data). The share generation equation is:

$$S_i = \sum_{j=1}^{q} \left(a_{ij} V_j \right) mod\ 256 \qquad \text{.....(1)}$$

Where, *i=1...n* and *j=1..q*. According to equation (1), the range of shares values S_i is [0..255].

The reveal phase is the inverse coding process of the sharing phase. In this phase any *q* different shares, taken from the total n shares, are collected for decryption. These *q* shares are used to construct *q* simultaneous linear equations set (i.e. one for each share), and thereby the secret bytes $\{V_j | j=1,2,...q\}$ can be obtained by solving these linear equations set. If less than *q* of simultaneous linear equations are collected, the linear equations cannot be solved to retrieve the secret bytes {*V()*}.

. The following algorithm steps have been adopted for the recovery process :

Algorithm 2: (*k*, *n*)-Threshold Reveal phase

Input: *k* shares // k array of bytes

k // *number of allied shares*

Output: *V* // array of bytes retrieved secret bytes

Step1: Take the shares whose corresponding indexes are $\{n_1, n_2, ..., n_k\}$; such that only one secret byte value is taken from any chosen share (i.e., $\{s_m | m = 1, 2, ... k\}$).

Step2: Construct the coefficients matrix, *A'()*, of the corresponding linear equations, that is

$$a'_{ml} = a_{n_m l} \qquad \text{..... (2)}$$

Where, $a'_{ml} \in A'$, $a_{n_m l} \in A$, *m=1, 2,.., k* and *l=1, 2, ..., k*.

Step3: Determine the determinant value of *A* (i.e., *D=det(A)*), and the corresponding complementary matrix *C*; such that for all values of *j* (i.e., $j \in [1,k]$) the following condition is satisfied:

$$\sum_{i=1}^{k} a_{ij} C_{ij} = \sum_{j=1}^{k} a_{ij} C_{ij} = D \qquad \text{..... (3)}$$

Here, the matrix element C_{ij} is equal to the determinant of the reduced matrix C (whose i^{th} row and j^{th} column are removed) multiplied by the factor $(-1)^{i+j}$.

Step4: The values of the retrieved secret bytes $\{V'_j | j = 1...k]\}$ could be determined using:

$$V'_j = \frac{1}{D}\left\{\left(\sum_{i=1}^{k} C_{ij} S_{n_i}\right) + w_j\right\} \quad \text{.....(4)}$$

Where, w_j is an integer number its value is multiples of 256, such that

:

$$w_j = 256 \sum_{i=1}^{k} C_{ij}\, p_i \quad \text{..... (5)}$$

PROPOSED APPROACH

This method presents a new highly resistant algorithm based on Discrete Cosine Transform (DCT) for secret image sharing. In proposed method, the DCT is used to reduce the secret image size firstly, then it employs the compact linear sharing scheme to generate the shadow images. For most natural images, a significant number of the high-frequency coefficients are small in magnitude and can be discarded. Therefore, in this method low-frequency DCT coefficients have been only reserved that keep most visual information in images. This drastically decreases the size of the data that must be shared, and guarantees the quality of the recovered image in the mean time. That is, the critical item for sharing is restricted to be just the low-frequency DCT coefficients value, and so the amount of information to be shared and that of the created share data both decrease noticeably. Therefore, the proposed scheme has the capabilities reducing the share data size. This capability extends the proposed scheme more practical for certain applications, where the memory size and network bandwidth are restricted. For example, the scheme is suitable for applications to mobile or handheld devices, where only a small amount of network traffic for shared transmission as well as a small amount of space for data storage are allowed.

In frequency domain, the operations of secret sharing scheme based on transform domain coefficients operations is done for breaking correlation. Note that our scheme, the DCT transform has been exploited to transform image to get a minimum share size . The presented secret image sharing scheme consists of two modules: (i) sharing module and (2) revealing module. As shown in Figure (1) the proposed sharing system consists of the following algorithm steps:

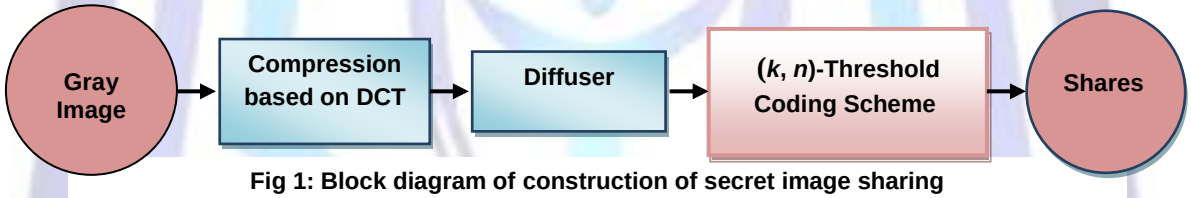


Fig 1: Block diagram of construction of secret image sharing

Algorithm 3. Encoding Process of Secret Image Sharing Based on DCT

Input: *O* // array of original gray image,

N // number of shares

α, Q_0 // Quantization Step

Output: Shares // *N* arrays of bytes (shares)

Step1: Read original gray image pixels *O*.

Step2: Read each of *N*, α, Q_0 values.

Step3: Pass the *N*, α, Q_0 and the image pixels *O* to the proposed compression method based on DCT. The compressed output is secret stream.

Step4: The compressed stream will be then diffused by applying proposed Diffuser to prune the bits significance in order to avoid localization problem.

Step5: After that, The produced compressed and ciphered stream bytes of the secret image is then shared using (*k*, *n*)-Threshold Scheme proposed by [11]**.** The output are *N* shares

The following algorithm shows the decoding phase of secret image sharing based on DCT:

Algorithm 4. Decoding Process of Secret Image Sharing Based on DCT

Input: *k* // Number of allied shares

shares // *k* array of bytes

α, Q_0 //Quantization Step.

Output: *O* // array of original gray image

Step1: Read each of *N*, α, Q_0 values.

Step2: Input the allied *k* shares pixels.

Step3: The *k* allied shares have been fed to the (*k*, *n*)-Threshold revealing scheme.

Step4: Apply the inverse of proposed Diffuser to the output of step3.

Step5: Perform the decompression based on DCT recovery where original secret image has been recovered.

THE MAIN PRINCIPLE OF THE PROPOSED COMPRESSION ALGORITHM:

The compression algorithm based on DCT has been proposed to preserve the main objectives of reducing the size of generated share size, easy maintenance and providing security. Data loss cannot affect the image clarity. It lowers bandwidth requirements for transmission, reducing cost. As shown in Figure (2), the proposed compression algorithm consists of the following algorithm steps:

Algorithm 5: Proposed Compression Based on DCT

Input: *Y, U', V'* // each of which is two dimension array

α, Q_0 //Quantization Step

Width // the width of component

Height // the height of component

Output: *S* // array of bytes (compressed data)

Step1: Input gray image is divided into blocks of 8 × 8.

Step2: Pass blocks of the to the DCT.

Step3: Each block is then compressed through quantization by using the following equations:

$$Q_{stp} = Q_0 (1 + \alpha (u+v)) \quad \text{..... (6)}$$

$$Qdct(u, v) = Round(dct(u, v) / Q_{stp}) \quad \text{..... (7)}$$

Step4: The quantized coefficients have been mapped to positive to avoid coding complexity due to existence of positive/negative values. These quantized coefficients values are mapped to be always positive, by applying equation (3). According to this equation all negative values are mapped to be odd while the positive values will be even.

$$X_i = \begin{cases} 2X_i & if\ X_i \geq 0 \\ -2X_i - 1 & if\ X_i < 0 \end{cases} \quad \text{.....(8)}$$

where X_i is the $_i$th element

Step5: A zig-zag scan is used to turn the 8×8 matrix (*F*(*u, v*)) into a 64-vector.

Step6: The DC coefficients are coded separately from the AC ones method such as following:

- *Differential Pulse Code Modulation (DPCM)* is the coding method such as following:

  $d_i = DC_{i+1} - DC_i$, and $d_0 = DC_0$ .....(9)

- Mapping *d* to positive using (3) equation.
- Determine the optimal values of the short (n_S) and long (n_L) codewords of *d* using algorithm (4)
- Perform the shift encoding and save the output codewords in the binary compression stream of bits.

Step7: AC coefficients finally undergo an shift coding step to gain a possible further compression, such as following:

- Mapping *AC* to positive using (3) equation.
- Apply RLC which aims to turn the *F*(*u, v*) values into sets {#-zeros-to skip next non-zero value}.

- Determine the optimal values of the short (n_S) and long (n_L) codewords of *AC* using algorithm (4)
- Perform the shift encoding and save the output codewords in the binary compression stream of bits.

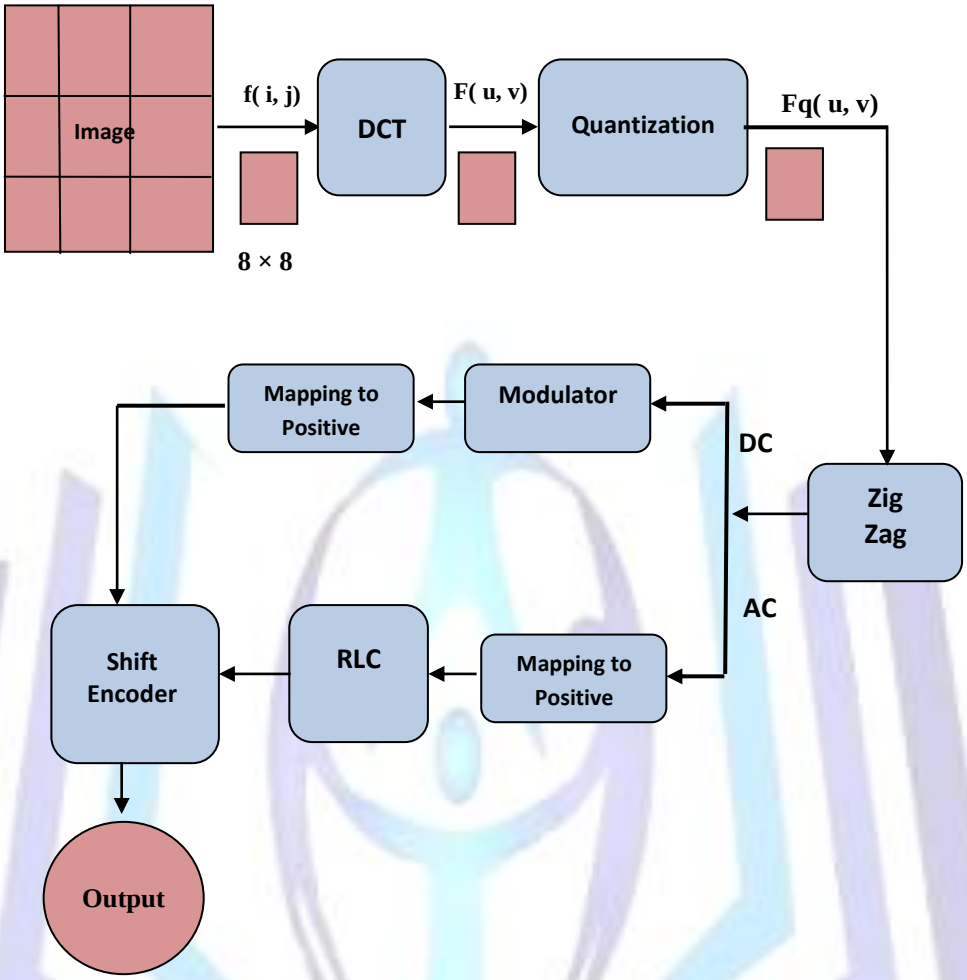


Fig 3: Proposed compression algorithm based on DCT

Algorithm 6: Determine Optimal Values of short and long Codewords

Input: *Input* // array of bytes

h // length

Output: n_S // short codeword

n_L // long codeword

Step1: Let $n_L \leftarrow \lceil \log_2(L) \rceil$,

where $\lceil x \rceil$ means the smallest integer value higher than *x*, *L* is the highest value found in the stream of collected coefficients (after making map-to-position task).

Step2: Find the n_S value that leads to lowest possible values for n_T, where

$$n_T = n_S \sum_{i=0}^{L} His(i) + n_L \sum_{i=n_L}^{L} His(i) \qquad \text{..... (10)}$$

The algorithm of decompression process is presented in algorithm (5).

Algorithm 7: Decompression Based on DCT

Input: *S* // array of bytes (compressed data)

α, Q_0 // Quantization Q_{stp}

Width // the width of component

Height // the height of component

Step1: for each chrominance isolate compressed *DC* and *AC* as DC_1 *and* AC_1.

Step2: For DC_1, the decoding such as following:

- Perform the shift decoding and save the output in DC_2.
- Mapping DC_2 to negative using the following equation

$$X_i = \begin{cases} X_i/2 & if\ X_i\ is\ even \\ -(X_i+1)/2 & if\ X_i\ is\ odd \end{cases} \quad \text{..... (11)}$$

- Then decoding of *DPCM* is applied on DC_2 using the following equation:

$$S_{i=}S_i + S_{i-1} \qquad for\ i=1,..,\ n\text{-}1 \qquad \text{..... (12)}$$

Step3:The decoding of AC_1 such as following :

- Perform the shift decoding and save the output in AC_2.
- Divide the output of step 3 into blocks of 8 × 8.
- Apply RLC decoding .
- Mapping to negative using (6).
- Combine the DC coefficient with AC coefficients .
- Zig-Zag Descan ..

Step4: Each block is then decompressed through dequantization by using the following equations:

$Q_{stp} = Q_0 (1 + \alpha (u+v))$

$Qdct(u, v) = Round(dct(u, v) \times Q_{stp})$ ..... (13)

Step5: Apply the inverse of DCT.

PROPOSED DIFFUSER

The structure of the proposed diffuser shown in Figure (3), which it implies set of operation on sets consists of four 64-bit words. This reversible diffuser is applied to compressed coefficients. It is required to provide the necessary diffusion, It uses a mixture of operations consists of different algebraic group: XOR, addition, rotation, multiplication and key dependent rotation.

The algorithm of proposed diffuser is described in the following steps

Algorithm 8: Diffuser

Input: *S* // array of bytes (compressed data)

Output: *DiffS* // array of bytes (diffused data)

Step1: Find the number of bytes that has been padded with zero to make the length *h* dividable by 32 such as following: *Diff* ← 32 - (*h* mod 32)

Step2: Padding with zero

For I= h →h + Diff

S[I] ← 0

end loop I

h←h+Diff

Step3: *For I=0⇾h step 32*

For I=1⇾8

- Take 32 bytes *X* which is 256 bits then it divided into four 64 bits *A, B, C,* and *D*.
- The 64 bits *A* is input to *F* function (as shawn in figure (4) which is a 64-bit block Feistel network controlled by four keys.
- The output from *F* function (i.e. *A'*) has been expanded into the three 32 bits *L*, *M*, and *R* by proposed *EF* function.
- Then performed the following:

  $B' = B \oplus R$

  $C' = C + M$

  $D' = D \oplus R$

- Combine *A'*, *B'*, *C'*, and *D'* to produce the output block *X'*
- Performed Byte permutation on *X'*.
- *Put X' into DiffS*

end loop J

end loop I

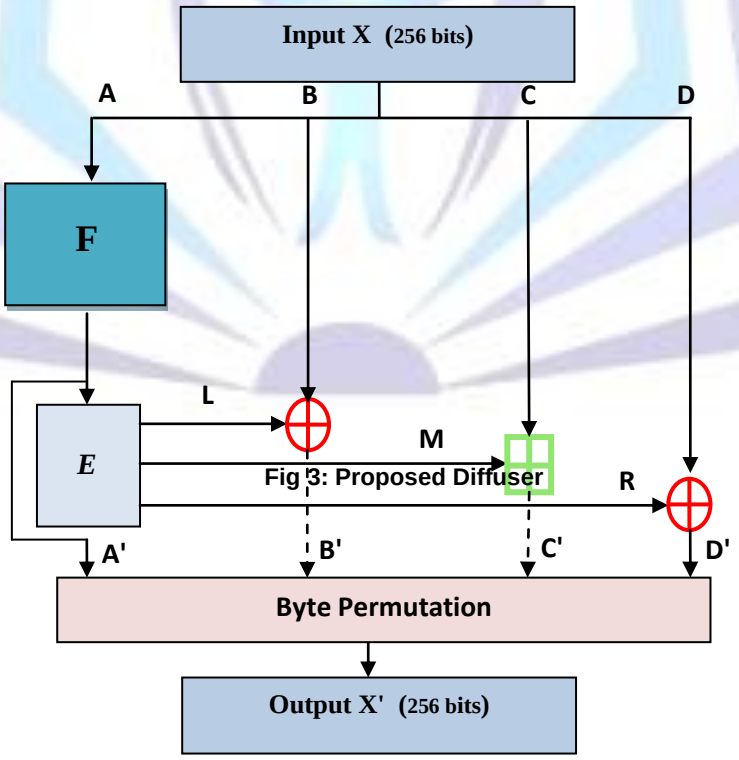


Fig 3: Proposed Diffuser

The algorithm of function *F* as shown in Figure (4) such as following:

ISSN 2277-3061

Algorithm 9: *F* function

Input: *A* // 64 bits

Output A' // 64 bits

Step1: The input *A* is divided into two 32 bits x_l and x_r, then x_l is subdivided into two 16 bits x_{l1} and x_{l2} then,

Step2: $x_{l1} \leftarrow x_{l1} \lll 4$

$x_{l2} \leftarrow x_{l2} \ggg 5$

$x_{l1} \leftarrow x_{l1} + k_1$

$x_{l2} \leftarrow x_{l2} + k_2$

$x_l \leftarrow x_{l1} \oplus x_{l2}$

$x_r \leftarrow x_r \oplus x_l$

Swap (x_l, x_r)

Step3: Repeat step 2-2 with k_3 and k_4

Step4: Combine x_l, x_r in *A'*

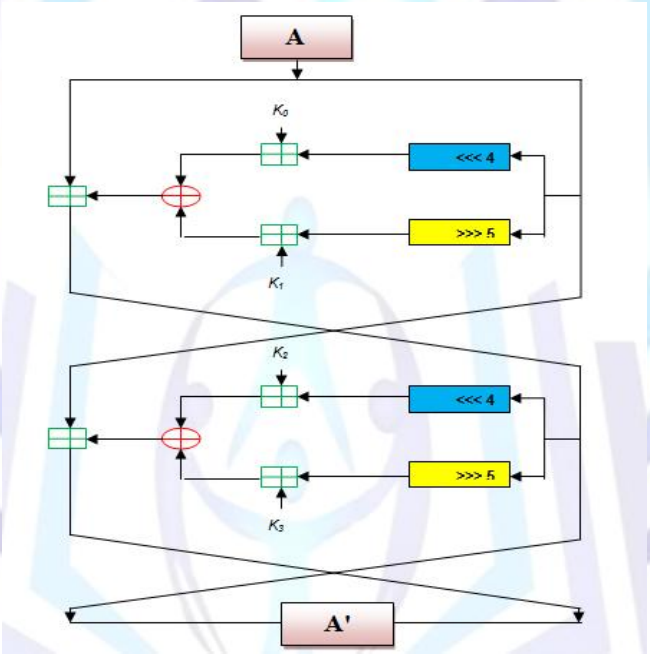


Figure (5) shows the effect of applying proposed Diffuser on of "Lena" image of size 512×512.

Fig 4: The *F* function of Proposed Diffuser

ISSN 2277-3061

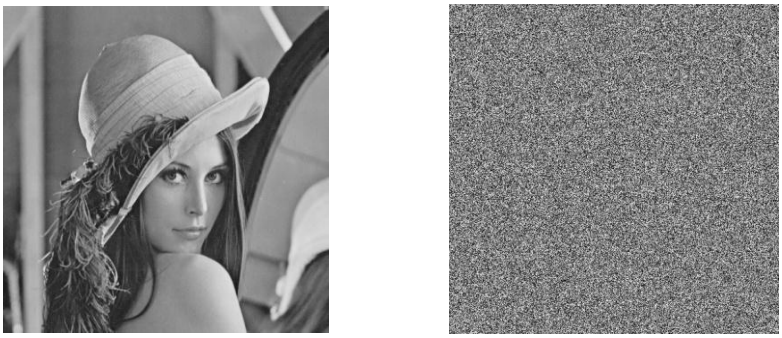


Fig 5: "Lena" image after applying proposed Diffuser

EXPERMINTAL RESULTS AND DISCUSSION OF PROPOSED APPROACH

Various experiments have been carried out to assess the performance of the proposed algorithm. Five bitmap images "Lena" ,"Splash" , "jet-plane" of size 512×512 and "House", "jelly_beams" of size 256×256 were used as secret imag ge: ure

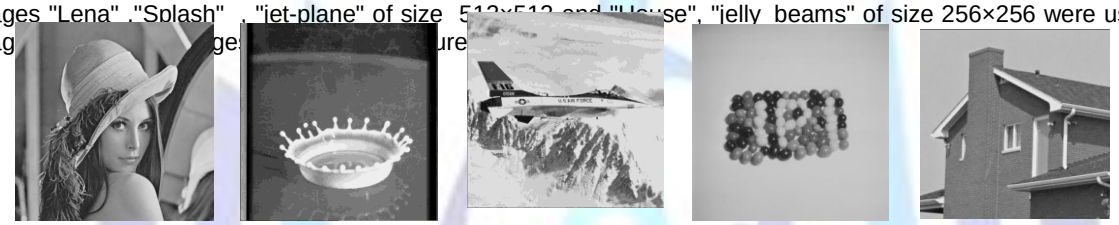


Fig 6:Test images

Table (1) lists the number of bytes spend to encode each image for different cases of quantization step of Q_0 and α by pplying method *II* based on DCT.

Table (1) The Number of bytes spend to encode each test image

Q_{stp}		Number of Bytes				
Q_0	α	Lena	Splash	Jet-plane	House	jelly_beams
1	1	74179	56830	77217	18954	10064
	2	47493	35515	51846	12389	7523
	3	37606	28155	41680	9627	5996
	4	30613	22753	34337	7664	5244
	5	27004	20182	30414	6669	4728
2	1	43484	32146	48118	11394	6696
	2	28998	21406	32771	7262	5034
	3	23448	17445	26692	5827	3919
	4	19008	14120	21291	4703	3473
	5	16819	12773	18879	4192	3165
3	1	31672	23117	35842	8140	5414
	2	22554	15516	25782	5223	3794
	3	17306	13033	19506	4306	3240
	4	14874	11637	16754	3752	2879
	5	13265	10664	14944	3409	2674

\Table (1) lists the size of compressed secret image which it has been compressed after applying proposed compression method. The size of "Lena" is became 74179 bytes after compression (with Q_0=1 and α=1), and then, the size of each share become 37,090 bytes when *k*=2 while it become 24,727 when *k*=3. The size of secret image has been reduced to approximately 1:3.5 of the size of the original secret image. Also, when the proposed (*k*, *n*)-Threshold scheme has been applied, the size of secret image has been reduced to 1:7 when k=2 and to 1:10.5 when k=3 (i.e. the reduction in size is approximately 1:3.5*k*).

Table (2) showed the compression ratio (CR) of thefive test images when proposed method *has* been applied

Table (2) A Comparison between the CR for different cases quantization steps

Q_{stp}		CR				
Q_0	α	Lena	Splash	Jet-plane	House	jelly_beams
1	1	3.53	4.61	3.39	3.46	6.51
	2	5.52	7.38	5.06	5.29	8.71
	3	6.97	9.31	6.29	6.81	10.93
	4	8.56	11.52	7.63	8.55	12.50
	5	9.71	12.99	8.62	9.83	13.86
2	1	6.03	8.15	5.45	5.75	9.79
	2	9.04	12.25	8.00	9.02	13.02
	3	11.18	15.03	9.82	11.25	16.72
	4	13.79	18.57	12.31	13.93	18.87
	5	15.59	20.52	13.89	15.63	20.71
3	1	8.28	11.34	7.31	8.05	12.10
	2	11.62	16.90	10.17	12.55	17.27
	3	15.15	20.11	13.44	15.22	20.23
	4	17.62	22.53	15.65	17.47	22.76
	5	19.76	24.58	17.54	19.22	24.51

As a useful capability should be supported by any image compression scheme is a control on the image quality. Such capability will make the user capable to specify the quality of compressed image, and also offer a control on the compression size. In image compression, it is hard to identify an accurate measure of quality. The metric Peak Signal to Noise Ratio (PSNR) is used to quantitatively evaluate the visual fidelity of the constructed image compared with its riginal ersion. For a 8 bit gray image, the PSNR is defined as [12]:

$$PSNR = 10\log_{10} 255^2 / MSE(dB) \quad \text{,..... (14)}$$

where MSE is the mean squared error between the original image and the retrieved image, it is given by:

$$MSE = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} (X_{ij} - X'_{ij})^2 \quad \text{,..... (15)}$$

Table (3) showed the PSNR results of the five test images when proposed method has been applied.

ISSN 2277-3061

Table (3) A Comparison between the PSNR for different cases of quantization Steps

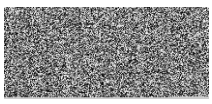
Q_{stp}		PSNR				
Q_0	α	Lena	Splash	Jet-plane	House	jelly_beams
1	1	38.83	40.96	39.36	38.82	46.68
	2	36.81	39.45	36.79	36.43	45.66
	3	35.75	38.51	35.27	35.01	44.56
	4	34.73	37.82	34.21	34.07	43.68
	5	33.79	37.16	33.24	33.40	43.26
2	1	36.46	39.04	36.32	36.05	44.29
	2	34.49	37.60	33.99	33.87	43.32
	3	33.25	36.37	32.44	32.74	42.48
	4	32.43	35.64	31.62	32.03	41.55
	5	31.71	34.89	30.74	31.36	41.11
3	1	35.06	37.89	34.68	34.49	43.66
	2	33.11	36.19	32.27	32.66	42.16
	3	31.94	35.07	31.01	31.46	41.11
	4	31.09	33.99	30.01	30.83	40.30
	5	30.64	33.44	29.27	30.35	39.81

Figure (7) shows an example of the proposed system output, the sharing scheme was set (n=4, k=2). The size of Lena image is became 74179 bytes (i.e. approximately (272×273)) bytes (with Q_0=1 and α=1), and then, the size of each share is became 37,090 bytes (i.e. approximately (272 × 137)).

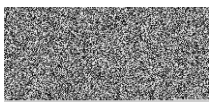
Original Image

512 × 512

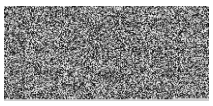
Share1
(272 × 137**)**



Share2
(272 × 137**)**



Share3
(272 × 137**)**



Share4
(272 × 137**)**

ISSN 2277-3061

Share1 & Share2 Share2 & Share3 Share1 & Share3

Fig 7: Example of (2,4) sharing conditions to retrieve Lena image.

Histograms of four diffused shares and their corresponding original shares after applying proposed Diffused have beenshown in Figure (8). The results have widely different contents.

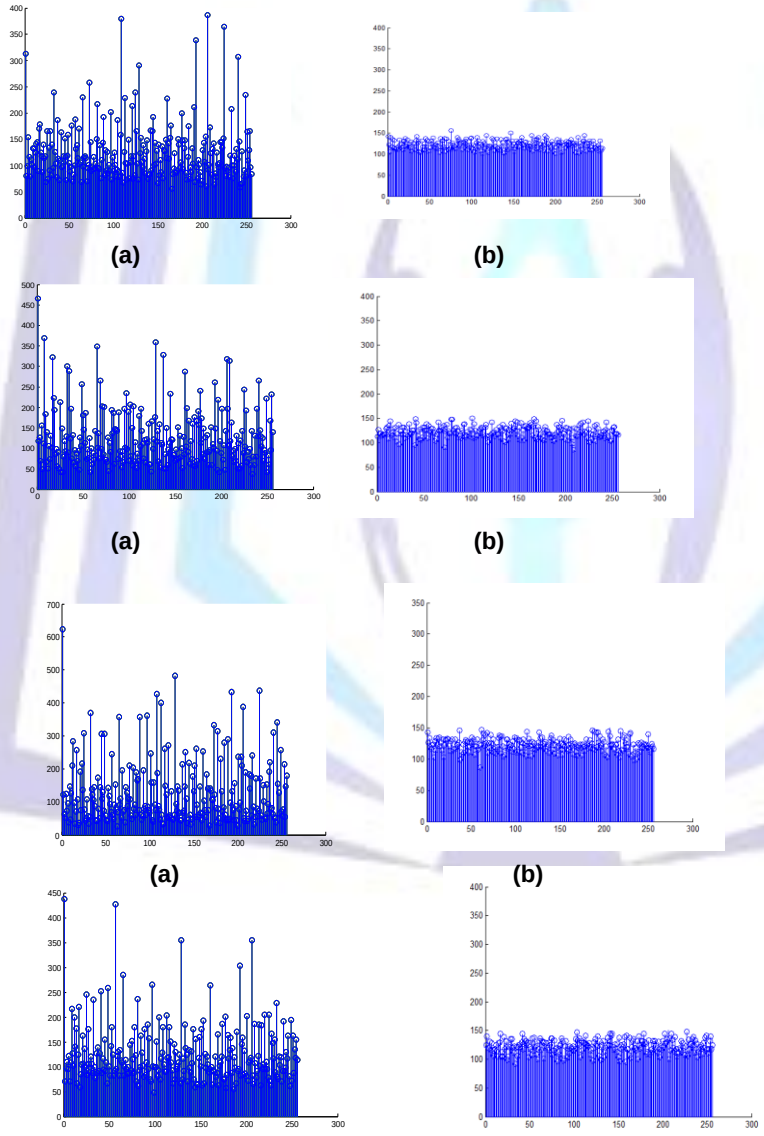


Fig 8: The diffusion results, (a) Histogram original share, (b) Histogram of diffused shares after applying proposed Diffuser

CONCLUSION AND FUTURE SCOPE

In this paper, a technique for gray image *(k*, *n*) secret sharing is introduced; it is based on the DCT. The advantages of DCT transform coefficient magnitude decay and excellent energy compaction were taken as benefit to gain data packing beside to sharing. It is noteworthy to address some features of the proposed scheme:

- Efficiency of bandwidth and storage space: Using DCT to support image shrinking property, the bandwidth for delivering shares and storage space for saving shares both decrease. Experimental results imply restored secret image identical to original image to visual perception. The schemes are thus suitable for certain application environments, such as the uses of mobile or handheld devices, where only a small amount of network traffic for shared transmission and a small amount of space for data storage are allowed.
- Security enhancement: The security of proposed system is guaranteed by several issues. The first one is each share depends on its own secret coefficients, a(), which made the recovery secret image is too complicated for attacker. second one is the use of diffuser to prune the existing bits significance in coefficients makes shares unbiased toward local significance; this will avoid the occurrence of localization problem.Finally Spatial correlation property of the secret image is eliminated.
- The proposed diffuser is designed to be used in upgraded computer environments. It uses the full menu of “strong operations” supported in modern computers to achieve better security properties. This approach enables us to get better security per-instruction ratio for our implemented software than is possible for the existing ciphers. The design takes full advantage of the ability of today’s computers to perform fast multiplications and data-dependent rotation*s.*
- Table 3 shows the PSNR values of the reconstructed secret images range from 46.68 to 27.29 dB.
- It is clear that the histogram of the diffused shares is nearly uniformly distributed, and significantly different from the respective histograms of the original shares. So, the diffused shares does not provide any clue to employ any statistical attack on the proposed diffuser, which makes statistical attacks difficult.

.

REFERENCES

[1] R. Yadagiri Rao. "Secure visual cryptography", International Journal of Engineering and Computer Science. 1(2. 2013), 265-303.

[2] A. Shamir. "How to share a secret", Communications of the ACM. 11(22, 1979), 612-613.

[3] C. Thien, J. Lin. "Secret image sharing", Computers & Graphics. (26, 2002), 765-770.

[4] C. Lin and W. Tsai. "Secret image sharing with capability of share data reduction", Optical Engineering.. 8(42, 2003), 2340-2345.

[5] Y. Wu, C. Thien, and J.C. Lin. "sharing and hiding secret iimages with size constraint", Pattern Recognition. 7(37, 2004), 1377-1385.

[6] C. Huang and C. Li. "A secret image sharing method using integer wavelet transform", Eurasip Journal on Advances in Signal Processing. (2, 2007),1-13.

[7] Z. Zhao, H. Luo, Z. Lu. 2009. "Joint secret image sharing and progressive transmission based on integer discrete cosine transform", Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, , Kyoto, Japan. 282-285.

[8] Z. Liu, S. Liu and M. Ahmed. "Image sharing scheme based on discrete fractional random transform", Optik, (121, 2010), 495-499.

[9] C. Yang, Y. Huang and J. Syue. 2011. "Reversible secret image sharing based on shamir’s scheme with discrete haar wavelet transform", Electrical and Control Engineering (ICECE), International Conference, Yichang,1250 - 1253.

[10] S. Nerella, K. Gadi, R. Chaganti, "Securing images using colour visual cryptography and wavelets", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, No. 3, Pp. 164-168, 2012.

[11] Ashwaq T. Hashim and Loay E. George. 2013. "Secret Image Sharing Based on Wavelet Transform", International Conference on Information Technology in Signal and Image Processing**,**, Mumbai, India, 324-332.

[12] Ahmet M. Eskicioglu, Paul S. Fisher, “Image Quality Measures and Their Performance” IEEE Transactions on Communication, 12(43, 1995), 2959-2965.

ISSN 2277-3061

Author' biography with Photo

Dr. Loay Edwar George received his Ph.D degree from Baghdad University , Iraq in 1997.

Thesis title is "New Coding Methods For Compressing Remotely Sensed Images". He is a member of Arab Union of Physics and Mathematics, and the Iraqi Association for Computers. Now, he is the Head of Computer Science Department, Baghdad University.

Ashwaq T. Hashim is working as Assistant Professor, in System and Control Engenerring Department, University of Technology, Iraq. She obtained M.Sc. from University of Basrah in 2003. She published more than 12 papers in cryptography, steganography and VHDL.