Middle-East Journal of Scientific Research 12 (12): 1873-1880, 2012
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2012.12.12.1236

Visual Secret Sharing Scheme for JPEG Compressed Images

T.V.U. Kiran Kumar, B. Karthik and E. Bharath Kumaran

Department of ECE,
Bharath University, Chennai-73, India

Abstract: Existing schemes modify either the pixel values or change the Color Index Table (CIT) values. Almost all the existing techniques are applicable to primarily non-compressed images in either monochrome or color domains. However, most imaging applications including digital photography, archiving and internet communications nowadays use images in the JPEG compressed format. Application of the existing Visual Secret Sharing schemes for these images requires conversion back into spatial domain. In this paper we propose a shared key algorithm that works directly in the JPEG domain, thus enabling shared key image encryption for a variety of applications. The scheme directly works on the quantized DCT coefficients and the resulting noise-like shares are also stored in the JPEG format. The decryption process is lossless preserving the original JPEG data.

Key words: Pixel values · Color Index Table · Compressed format · Shared key · Visual Secret sharing · Image encryption.

INTRODUCTION

A secret sharing encryption scheme creates several (n) shares of the original information and distributes to n participants. The decryption is carried out by using a prescribed number (k, $k = n$) of subset shares. Fewer than k shares are insufficient to reconstruct the original data. The 'k' keys are generated in such a way that the share images are "random" looking with no semblance to the original image. It is a highly secure mechanism since the decryption is being performed by the human visual system when both shares are brought together. Here first we will see 2-out-of-2 or {2, 2} secret sharing method. Next we will see an extension to k out of n {n,k} secret sharing where k stacked images are needed to reconstruct the clear image.

All the existing schemes, the encryption-decryption process is lossy. The original image is transformed into a haltoned image before deriving the share images. Halftoning is a lossy process. Furthermore, in many of the proposed schemes the reconstruction from the k shared images creates a lossy version of the half-toned image itself. This led to several researchers' efforts on improving contrast quality of the decrypted images. Though they lose the ability of decryption by visual means, such encryption schemes have a large number of potential applications in the Internet world. The computer-based decryption also opens an avenue to a wide variety of shared image encryption systems. This has also lead to development of methods that apply to color images. In addition, the methods described, this also address computational complexity issues by designing schemes that require simpler bit-level arithmetic operations during decryption. This method similar in spirit to these schemes but addressing a larger domain of images stored in compressed digital formats. Large number of imaging applications including digital photography, archiving and internet communications primarily use images stored in the JPEG format. Most of the digital cameras in the market use JPEG. Application of the existing shared key cryptographic schemes for these images requires conversion back into spatial domain. When transmission and storage of the shares are concerned, one may not necessarily be able to apply lossy compression techniques since the loss may result in an inability to decrypt. Hence, encryption techniques applicable in the compressed domain are needed.

Visual cryptography based on secret sharing has received considerable attention since the publication of [1] by Naor and Shamir. A secret sharing encryption scheme creates several (n) shares of the original information and distributes to n participants.

Corresponding Author: T.V.U. Kiran Kumar, Department of ECE, Bharath University, Chennai-73.

The decryption is carried out by using a prescribed number (k, $k<=n$) of subset shares. Fewer than k shares are insufficient to reconstruct the original data. The scheme proposed in [1] generated two shared key images from a given binary image as a printed page and a transparency of the same size. When the transparency is stacked on top of the printed page, the original image is formed. The two keys are generated in such a way that the share images are "random" looking with no semblance to the original image. It is a highly secure mechanism since the decryption is being performed by the human visual system when both shares are brought together. This is a 2-out-of-2 or {2,2} secret sharing method. Several publications that followed this development extended the basic visual cryptography using concepts from digital halftoning [2, 3] to address gray scale images and color pictures [4]. In many of these schemes, the encryption-decryption process is lossy. The original image is transformed into a haltoned image before deriving the share images. Halftoning is a lossy process. Furthermore, in many of the proposed schemes the reconstruction from the k shared images creates a lossy version of the half-toned image itself. This led to several researchers' efforts on improving contrast quality of the decrypted images.

However, in some of the recent publications [6, 7] the original intent of performing the decryption using the human visual system with a simple mechanical system of stacking transparencies is lost by requiring computer processing to reconstruct the image. Though they lose the ability of decryption by visual means, such encryption schemes have a large number of potential applications in the Internet world [6]. This has also lead to development of methods that apply to color images. In addition, the methods described in [6] and [7] also address computational complexity issues by designing schemes that require simpler bit-level arithmetic operations during decryption. In this paper, we describe a method similar in spirit to these schemes but addressing a larger domain of images stored in compressed digital formats.

In [5] the author proposed a scheme for shred key encryption, which is used in this paper for creating shares from a secret image. The author a proposed scheme for {2,2} shared key encryption for JPEG images and suggested the implementation for {n,k} shared key encryption. Based on this paper the {n,k} shared key encryption has been developed and shown the results in this paper.

Visual Secret Sharing Scheme

(A) {2, 2} Shared Key Encryption: This shared key algorithm [5] that works directly in the JPEG domain, thus enabling shared key image encryption for a variety of applications. The scheme manipulates the quantized DCT coefficients and the resulting noise-like shares are also stored in the JPEG format. The decryption process that combines the shares is lossless and hence the original JPEG file's fidelity is preserved. Our experiments indicate that each share image is approximately the same size as the original JPEG retaining the storage advantage provided by JPEG.

Both encryption and decryption require only small modifications to standard JPEG computational procedures. This in turn implies an easy adaptation for a hardware implementation which is particularly useful for digital camera applications.

Monochrome Images: The lossy version of JPEG image compression uses DCT.A monochrome image is first split into 8×8 non-overlapping blocks of pixels. An 8×8 DCT is applied to each block and the resulting coefficients are scalar quantized using a quantization matrix. The quantized coefficients are then converted from a two-dimensional representation to a one-dimensional vector by a process known as zig-zag scanning and sent to an entropy coder that uses either Huffman or arithmetic coding. The process is shown in Fig 2.

The zig-zag transformation transforms 8x8 two 2D values into 64 one dimensional quantized coefficients. For an 8×8 block Bn with an index X^n, let the quantized coefficients be denoted as, where X^n corresponds to the DC coefficient and the rest to the AC coefficients. The index n can be obtained as the standard scanning order of blocks of the image as defined in the JPEG standard. We will also use the notation of

$$X_n=[X_n^1, X_n^{1,} X_n^1,\ldots, X_n^1] \quad (1)$$

According to baseline JPEG specification, the input image components are represented by 8-bit pixel values and the quantized samples are limited to at most 15 bits of magnitude and a sign bit. For gathering the statistics for entropy coding purposes, $XN\ i$ values are placed in bins according to the number of bits needed to represent its magnitude in binary. This is given by

$$N=\log_2 X_1^n \quad (2)$$

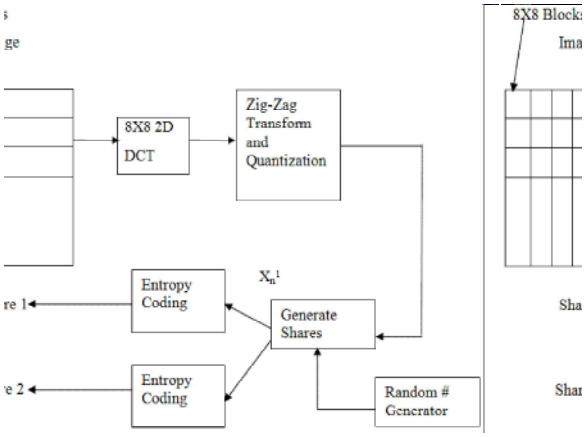

Fig. 1: {2,2} Shared Key Encryption Mechanism for JPEG images

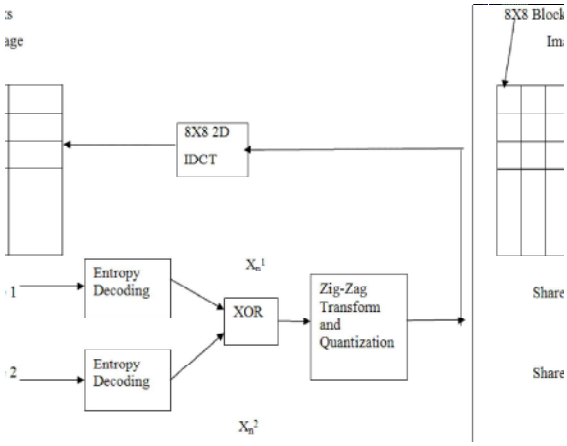

Fig. 2: Shared Key Decryption Process.

Now, X_i^n can be represented in two's complement form as

$$X_i^{nb} = b_i^n(N)\, b_i^n(N-1) \ldots b_i^n(0) \quad (3)$$

Where each $b_i^n(l)$, l=0,1,2,…N

N is a binary value, either zero or one. Hence, the value of *X ni* can be obtained as N

$$X_i^n = -b_i^n(N) \times 2^N + \Sigma b_i^n(N-L) \times 2^{N-L} \quad L=1 \quad (4)$$

Note that *N* is dependent of both *i* and *n* and we have not explicitly shown the dependency to reduce clutter. Now, let us consider an encryption system where the image to be encrypted is given in the JPEG format. Let the image be decoded partially to obtain the zig-zagged, quantized coefficients X_i^n=0,1,2,3,…63.corresponding to block *Bn*. At this stage, the two's complement representation X_i^{nb} of X_i^n as indicated in the above equation is available. Now, two versions, X_i^{n1} and X_i^{n2}, are generated using a {2, 2} shared secret key encryption method based on random assignment. We use an assignment scheme based on the one proposed by [7]. Each bit *b n i* (*l*) is split into two shares using the following scheme:

$$b_i^{n1}(L)\, b_i^{n2}(L) \in \begin{cases} \{[0,1],[1,0]\} \text{ if } b_i^{n1}(L) = 1 \\ \{[0,1],[1,0]\} \text{ if } b_i^{n1}(L) = 1 \end{cases} \quad (5)$$

The above procedure is applied to all *N*+1 bits of X_i^{nb} to obtain X_i^{n1b} *i* and $X_i^{n2b}i$ whose values X_i^{n1} and X_i^{n2} respectively are calculated as

$$X_i^{n1}(L) X_i^{n1}(L) X2^N + \sum_{L=1}^{N} b_i^{n1}(N\text{-}L) X2^{N\text{-}L} \quad (6)$$

and

$$X_i^{n2}(L) X_i^{n2}(L) X2^N + \sum_{L=1}^{N} b_i^{n1}(N\text{-}L) X2^{N\text{-}L} \quad (7)$$

Using the above procedure, we obtain two sequences, *Xn*1 and *Xn*2, that represent zig-zagged, quantized coefficients for 8×8 blocks for the entire image based on the original sequence *Xn*.These two sequences can further be entropy coded using either Huffman or arithmetic coding techniques to produce the two share images in the JPEG format. The sequence *Xni* and the binary representation *Xnbi* can be reconstructed simply by a bit-wise XOR operation (+) of the *N*+1-bit shares. (8)

Color Images: Just as with the JPEG compression, the proposed method is color blind. JPEG uses a component model for applications to enable coding of color images. Typically, most applications use three components in the YCbCr color space. The chrominance data is sub-sampled for better compression efficiency. The proposed encryption scheme uses the same JPEG approach to handle color images. Since the resulting image shares are JPEG images, any color space that can be handled by JPEG is also suitable for our application. JPEG supports up to 255 components in one image and hence support for a large variety of image formats. The description of the encryption technique in the previous subsection relies on a standard baseline mode of JPEG. JPEG also has a number of other modes such as progressive DCT,

hierarchical and lossless. The progressive DCT mode by multiple scans of the quantized coefficients allows incremental transmission of the image so that the picture could be reconstructed sequentially with increasing fidelity. There are two modes for performing progressive DCT: spectral selection and successive approximation. In spectral selection, the AC coefficients are sent after sending the DC coefficients. The successive approximation uses a lower precision at the start and improves the precision in the subsequent scans. A mixed mode by combining the two schemes is also possible. For the progressive-DCT JPEG encoding, the proposed encryption scheme applies directly. Once the shares are created, the progressive-DCT based scans could be applied to generate two JPEG shares. During the decode, the shares have to be completely decoded to produce *Xn*1 and *Xn*2. XOR operation on these values produce the *Xn* sequence which can further be processed to obtain the clear picture. JPEG also has a lossless mode. The lossless coding uses differential pulse code modulation (DPCM) and Huffman or arithmetic coding. The proposed encryption method does not directly apply to lossless mode but a scheme described in could be used for this purpose. The hierarchical mode in JPEG uses multi-scale approach to compression. While the proposed scheme can potentially work in the hierarchical mode with DCT-based coding, it will not work in the lossless hierarchical mode.

The scheme [5] for share generation is listed below.

- Read the image in JPEG format and perform variable-length decode to obtain the quantized

DCT coefficients, *X n,* for each block *Bn.*

- Using random numbers from a pseudo-random number generator, produce the Sequences *Xn*1 and *Xn*2.
- Perform entropy coding for the two sequences to produce two JPEG share images with the same dimensions and color space.

The decryption procedure is simply the reverse of it, where the decoder has also been modified to read two JPEG images and perform entropy decoding to get two share sequences of the quantized DCT coefficients. The sequences are XORed to produce the decrypted sequences and the rest of the decode operations, dequantization and inverse transform, are Continued to produce the decrypted image.

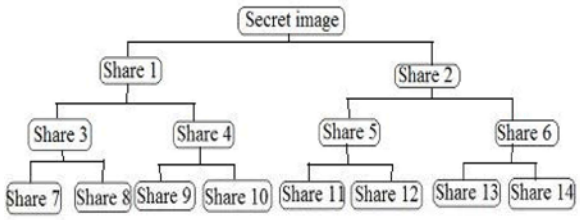

Fig. 3: Binary tree structure of {n,k} encryption

Table 1: Shared images and their positions in linear array

Name	Secret Image	Share 1	Share 2	Share 3	Share 4	Share 5	Share 6
ArrayPosition	1	2	3	4	5	6	7
Share 7	Share 8	Share 9	Share 10	Share 11	Share 12	Share 13	Share 14
8	9	10	11	12	13	14	15

{N,k} Shared Key Encryption Algorithm: We can extend the {2,2} encryption to {n,k} encryption by extending the scheme for the generated shares. (i.e). Again generating shares for the Shares 1 & 2. From Share 1 we will get Share 3 and Share 4.Similarly from Share 2 we will get Share 5 and Share 6.This scheme looks like a downward growing binary tree. For the better programming of binary tree, we put it into single dimensional array as following.

The child node of a parent node 'p' (p is position) can expressed as, Left child=2 x p Right child= (2 x p) +1

RESULTS

{2, 2} Shared Key Encryption: We performed the experiments on several types of images and compared its performance. The shares were generated using the proposed method. The decrypted image is exactly same as the original, From Figs.; it is clear that the generated shares have no visual resemblance to the originals and are "random" looking. The randomness is slightly modulated by the underlying image shapes. Hence, the storage and transmission gains obtained by JPEG are directly applicable to the proposed method.

{N,2} Shared Key Encryption: Here 14 Share images have been generated from the secret image. The decryption can be carried out from the following combination of share images.

Fig. 4: Secret Image

Share 1

Share2

Decrypted from Share 1 and Share 2 images Figure 2 {2,2} Shared key encryption

***2 Shares*:**
{Share 1, Share 2}

3 Shares:
{Share 2, Share 3, Share 4}
{Share 1, Share 5, Share 6}

4 Shares:
S=Share
{S3,S4, S5, S6} {S1, S5, S13, S14} {S1, S6, S11, S12}
{S2, S3, S9, S10} {S2, S4, S7, S8}

5 Shares:
{S4,S5,S6,S7,S8}{S3,S4,S5,S13,S14}
{S3,S5,S6,S9,S10}{S3,S4,S6,S11,S12}
{S1,S11,S12,S13,S14}{S2,S7,S6,S8,S9,S10}
{S1,S2,S3,S7,S8}{S1,S2,S4,S9,S10}
{S1,S2,S5,S11,S12}{S1,S2,S6,S13,S14}

6 Shares:
{S5, S6, S7, S8, S9, S10} {S3, S4, S11, S12, S13, S14}
{S3,S5, S9, S10, S13, S14} {S4, S6, S7, S8, S11, S12}
{S3,S6,S9,S10,S11,S12} {S4,S5,S7,S8,S13,14}
{S1,S3,S5,S6,S7,S8} {S1,S4,S5,S6,S9,S10}
{S2,S3,S4,S5,S11,S12} {S2,S3,S4,S6,S13,S14}

7 Shares:
{S6,S7,S8,S9,S10,S11,S12}
{S3,S9,S10,S11,S12,S13,S14}
{S4,S7,S8,S11,S12,S13,S14}
{S5,S7,S8,S9,S10,S13, S14}
{S1,S3,S5,S7,S8,S13,S14}{S1,S3,S6,S7,S8,S11,S12}
{S1,S4,S5,S9,S10,S13,S14}{S1,S4,S6,S9,S10,S11,S12}
{S2,S3,S5,S9,S10,S11,S12}{S2,S3,S6,S9,S10,S13,S14}
{S2,S4,S5,S7,S8,S11,S12}{S2,S4,S6,S7,S8,S13,S14}
8 Shares
{S7, S8, S9, S10, S11, S12, S13, S14}
{S1,S2,S3,S4,S7,S8,S9,S10}
{S1,S2,S5,S6,S11,S12,S13,S14}
{S2,S6,S7,S8,S9,S10,S13,S14}
{S2,S5,S7,S8,S9,S10,S11,S12}
{S1,S4,S9,S10,S11,S12,S13,S14}
{S1,S3,S7,S8,S11,S12,S13,S14}
{S1,S2,S4,S6,S9,S10,S13,S14}
{S1,S2,S4,S5,S9,S10,S11,S12}
{S1,S2,S3,S6,S7,S8,S13,S14}
{S1,S2,S3,S5,S7,S8,S11,S12}

9 Shares:
{S1,S3,S4,S5,S6,S7,S8,S9}
{S2,S3,S4,S5,S6,S11,S12,S13,S14}

10 Shares:
{S1,S3,S4,,S6,S7,S8,S9,S10,S11,S12}
{S1,S3,S4,S5,S7,S8,S9,S10,S13,S14}
{S2,S3,S5,S6,S9,S10,S11,S12,S13,S14}
{S2,S4,S5,S6,S7,S8,S11,S12,S13,S14}

11 Shares:
{S1,S3,S4,S7,S8,S9,S10,S11,S12,S13,S14}
{S2,S5,S6,S7,S8,S9,S10,S11,S12,S13,S14}
{S1,S2,S4,S5,S6,S9,S10,S11,S12,S13,S14}
{S1,S2,S3,S5,S6,S7,S8,S11,S12,S13,S14}
{S1,S2,S3,S4,S6,S7,S8,S9,S10,S13,S14}
{S1,S2,S3,S4,S5,S7,S8,S9,S10,S11,S12}
14 Shares
{S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13,S14}

The numbers which are underlined and italic are containing the redundant shares.

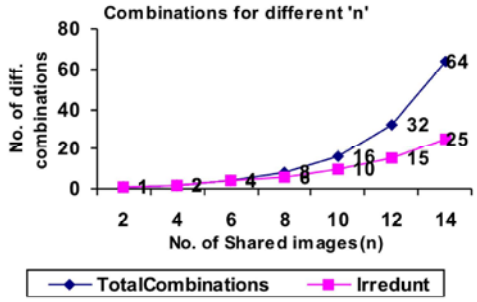

Fig. 5:

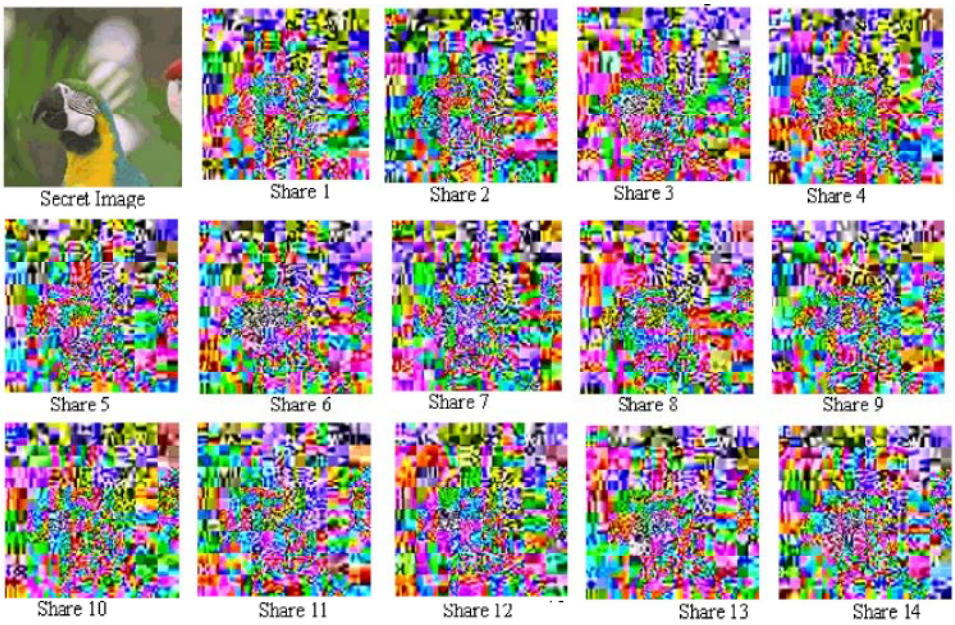

Fig. 6:

Table 2:

Shares required for decryption(k)	Shared images (n) 2	4	6	8	10	12	14
2	1	1	1	1	1	1	1
3	-	1	2	2	2	2	2
4	-	-	1	2	3	4	5
5	-	-	-	1+1	3+2	4+3	6+4
6	-	-	-	1	1+2	3+3	6+4
7	-	-	-	-	-	1+4	4+8
8	-	-	-	-	1	4	1+10
9	-	-	-	-	1	1	2
10	-	-	-	-	-	1	4
11	-	-	-	-	-	1	6
12	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-
14	-	-	-	-	-	-	1
Irredundant combinations	1	2	4	6	10	15	25
Total combinations	1	2	4	8	16	32	64

The numbers underlined and bold in table 2 are redundant combinations.

From the results we can generalize the formula as

Total Combinations=$2^{(n/2)-1}$

Where 'n' is the no of shared images.

From the results the following are identified.

- For any image and any 'n' the shared images will have almost equal size. And never equal.
- For a specified image and specified 'n' the shared images created at different period will never same.The probability for being same is 1/(ImageLength*ImageWidth*3*8).

The decryption can be carried out at different combination for different 'k' values.

- In the decryption combinations many set of combinations are redundant.

- The graph plotted for different 'n' values vs. the combinations shows, it is binary exponential.
- For the values 'k' greater than 4, contains the redundant shares.

Extension and Further Experiments

Generalizing the Formula for Decryption: The scheme can be generalized to a formula. Look out the first combination in all possible combination.
(1,2) (2,3,4) (3,4,5,6) (4,5,6,7,8) (5,6,7,8,9,10) (6,7,8,9,10,11,12) (7,8,9,10,11,12,13,14).If you start at position 'p' then up to 2p shares we need add to decrypt successfully (i.e.) p, p+1, p+2,…, 2p.Similarly this can be extended to get a generalized algorithm for decryption.

Share Creation in JPEG 2000 Image: The proposed scheme can be implemented in other transforms like wavelet (JPEG 2000) and any other transforms which is used for compression. The JPEG 2000 has higher compression and less visibility of artifacts. Thus it will be useful for higher compressed images.

Implementing in Reconfigurable Hardware: System generator tool [14] would be useful when implementing in FPGA (Field Programmable Gate Arrays).The tool can be used for converting matlab functions into VHDL/Verilog code for the specified target like Spartan 2E,Virtex-4,etc.Thus the encryption can be tested in hardware and the optimization can be achieved.More detals about the system generator are available in [14].

Summary and Future Work: The proposed method can take in JPEG and create 'n' shares in the JPEG format. The shares are generated using the quantized DCT coefficients in the JPEG representation. The shares are used to reconstruct the original quantized DCT coefficients during decryption. The proposed scheme is applicable to color and monochrome images and offers all the compression advantage of JPEG to all the share images.

The computational complexity of the encryption process is comparable to that of a standard JPEG encoder, an additional bit stream entropy encoder and a random number generator. Similarly, the decryption complexity is also augmented by an additional entropy decoder when compared to JPEG decoding. The hardware architecture of the encryption and the decryption processes can easily be obtained by modifying the standard JPEG pipelines. (A software implementation based on the Matlab code is available from the author.) It was also shown that the sizes of the resulting shares are comparable to that of their straight JPEG representations. The proposed method can be extended to any other transform or wavelet domain techniques for image coding. An extension to implement in FPGA is under development [14]. Currently, we are looking into application of this scheme for secure smartcards and digital rights management.

REFERENCES

1. Naor, M. and A. Shamir, 1995. in: A. de Santis (Ed.), Visual Cryptography, Advances in Cryptology: Eurocrypt '94, Lecture Notes in Computer Science, 950: 1-12.
2. Ateniese, G., C. Blundo, A. De Santis and D. Stinson, 1996. "Visual cryptography for general access structures," Information and Computation, 129(2): 86-106.
3. Lin C.C. and W.H. Tsai, 2003. "Visual cryptography for gray-level images by dithering techniques," Pattern Recognition Letters, 24: 349- 358.
4. Hou, Y.C., 2003. "Visual cryptography for color images," Pattern Recognition, 36: 1619-1621, 2003. Rahman Mohseni-Astani, Poorya Haghparast and Sahab Bidgoli-Kashani, "Assessing and Predicting the Soil Layers Thickness and Type Using Artificial Neural Networks - Case Study in Sari City of Iran", Middle-East Journal of Scientific Research, ISSN:1990-9233, 6(1): 62-68, 2010.
5. Ihsan Nuri Demirel, 2010. "Conceptual Description, Hastiness in Obtaining Result, Voluntary Participation, Assumption of Inefficiency", Middle-East Journal of Scientific Research, ISSN:1990-9233, 6(1): 15-21.
6. Subramania Sudharshan, 2005. "Shared key encryption of JPEG color images", IEEE Transactions on Consumer Electronics, 51: 4.
7. Hou, Y.C., C.Y. Chang and F. Lin, 1999. "Visual cryptography for color images based on color decomposition," Proceedings of the Fifth Conference on Information Management, pp: 584-191.
8. Chang, C.C. and T.X. Yu, 2002. "Sharing a secret gray image in multiple images," Proceedings of the First IEEE International Symposium on Cyber Worlds, pp: 230-237.
9. Lukac, R. and K.N. Plataniotis, 2004. "Cost-effective encryption of natural images," Proceedings of 22nd Queen's Biennial Symposium on Communications, pp: 89-91, (Also from the URL http://www.ece. queensu. ca/symposium/papers/2C_1.pdf)

10. Independent JPEG Group, http://www.ijg.org.
11. Mathematics Department, University of Salzburg, pLab: Theory and Practice of Random Number Generation,http://random.mat.sbg.ac.at/.
12. http://www.cryptography.com/resources/whitepapers/VIA_rng.pdf
13. Michael W. Marcellin, Michael J. Gormish, Ali Bilgin and Martin P. Boliek, 2000. An Overview of JPEG-2000.Proceedings of the Data Compression Conference, pp: 523-544.
14. http://www.mathworks.com
15. http://www.xilinx.com