http://dx.doi.org/10.14236/ewic/bcsme2014.8

A New Watermarking Algorithm for Scanned Grey PDF Files Using Robust Logo and Hash Function

Walid Alakk
Electrical and Computer Engineering Department
Khalifa University of Science, technology and Research
Sharjah, UAE
100032288@kustar.ac.ae

Hussain Al-Ahmad
Electrical and Computer Engineering Department
Khalifa University of Science, technology and Research
Sharjah, UAE
alahmad@kustar.ac.ae

This paper deals with the development and assessment of a watermarking technique which is suitable for scanned PDF documents. The watermark will serve two purposes. The first one is a logo to protect the copyright ownership. This watermark should be invisible and secure and can be extracted even if the document has gone through slight image manipulations. The second watermark will be used to authenticate the document. A slight editing in the document will change the second watermark and indicate forgery. The algorithm was tested successfully on a variety of scanned documents and the performances of the algorithm were assessed.

Key words: Watermarking, DCT, Hash function, Hash-key, LSB, PSNR, PDF file

1. INTRODUCTION

Watermarking algorithms are used to insert digital data or digital signatures in the original media file to prove the owner's identity of that file and prevent copyright violation. Several commercial companies around the world offer copyright protection services to their customers. The inserted watermark can be visible (Wang-2009) where it can be seen by anyone who is viewing the file, or it can be imperceptible and invisible where it can be only detected by the one who created the watermark using some decoding algorithms. For imperceptible watermark, there is a need for it to be robust so that it cannot be destroyed or lost by modifying the digital media file. There is another requirement for watermarking for copyright protection which is that the algorithm should be blind. That means that the original media file is not needed to extract the watermarking information. However, in non-blind techniques, the original file is needed to extract the embedded watermark (Al-Mansouri-2012). Moreover, Watermarking techniques can be applied either in the frequency domain or in the spatial domain. Frequency domain techniques proved to be more immune and survive different attacks. In contrast, spatial domain watermarks are more sensitive and fragile and can be used to authenticate the copyright of the watermarked file (Al-Gindy-2007).

The distortion in the watermarked file after the watermarking process is analysed and assessed objectively using the peak signal to noise ratio (PSNR). In addition, the watermarking effect is assessed subjectively by viewing the watermarked file (Wang-2004).

This paper introduces a way to watermark PDF files containing grey images in both frequency and spatial domain by converting the PDF file into an image file. The watermark will be inserted in the spatial domain using hash function and in the frequency domain using the Discrete Cosine Transform (DCT). Five sections are included in this paper. Section 2 discusses the algorithm for embedding the watermark signals. Section 3 illustrates the extraction process. Section 4 demonstrates the results of watermarking and the effects of watermarking on the original PDF file and the extracted watermark. Finally, section 5 concludes the work.

2. EMBEDDING THE WATERMARK SIGNALS

The watermarked PDF file will have two watermarks. One of them is robust and used for copyright protection. The first watermark is inserted in the frequency domain of the converted PDF file using DCT coefficients. The second watermark is fragile and used for authentication and discovering changes in the watermarked file. This watermark is inserted in the spatial domain of the file using the least significant bit (LSB) method. The fragile

© The Authors. Published by BCS Learning and Development Ltd.

watermark is generated by using SHA-256 hash function (Cannons-2004). Fig. 1 represents a block diagram for the operation of the algorithm.

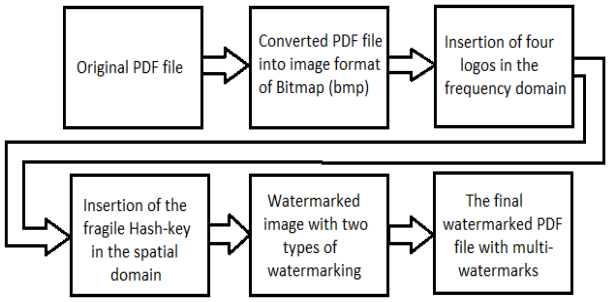

Fig 1: *A block diagram of the algorithm*

2.1 DCT algorithm

The PDF file is first converted into an image file. Then the image is divided into 8x8 blocks and then converted into the frequency domain using 2D DCT. The image is then screened to determine the best low frequency coefficients to insert the bits of the logo. If the logo is smaller than the number of available blocks then it can be repeated several times. The chosen coefficients in each block are the first four coefficients other than the DC component as illustrated in the zig-zag method. The insertion process in the coefficients is done by using Even/Odd technique. Fig. 2 shows a diagram that summarizes the DCT embedding process. The equation for the Even/Odd process is shown as follows:

$$F_k(u,v) = \mathrm{DCT}\{f_k(i,j)\}$$
$$1 \le u,v \le 8, 1 \le k \le N_{HB} \qquad (1)$$

if $w(i,j) = 1$ then

$$F_k(u,v) = \begin{cases} \Delta Q_o\left(\dfrac{F_k(u,k)}{\Delta}\right) & u,v \in H_k \quad 1 \le k \le N_{HB} \\ F_k(u,v) & u,v \notin H_k \quad 1 \le k \le N_{HB} \end{cases}$$

if $w(i,j) = 0$ then

$$F_k(u,v) = \begin{cases} \Delta Q_e\left(\dfrac{F_k(u,k)}{\Delta}\right) & u,v \in H_k \quad 1 \le k \le N_{HB} \\ F_k(u,v) & u,v \notin H_k \quad 1 \le k \le N_{HB} \end{cases}$$

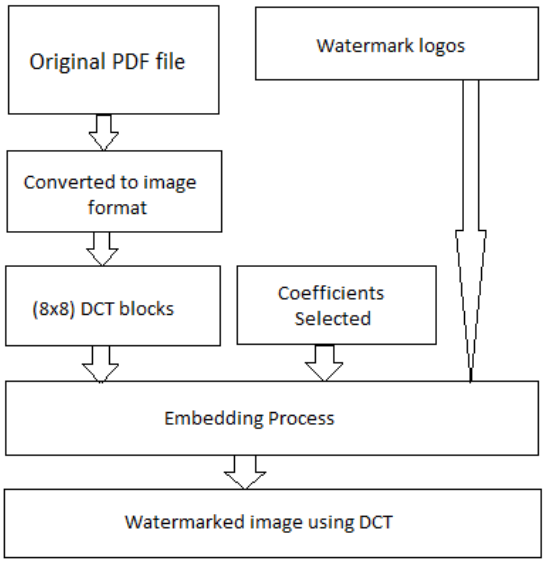

Fig 2: *The DCT embedding*

In the previous equations, $f_k(i,j)$ indicates an (8x8) blocks of the original image and $F_k(u,v)$ is its Discrete Cosine Transform (DCT). $w(i,j)$ is the watermarked image. Moreover, Q_e represents an even quantization while Q_o represents an odd quantization to the nearest integer number. The H_k points to the chosen coefficient locations and Δ is the quantization scaling factor.

2.2 Hash function algorithm

The process of embedding the hash-key is shown in Fig. 3.

Step 1: the watermarked image, with the robust watermarks using DCT, is divided into two parts. One of them is the whole image excluding the first row. The other part is the selected row.

Step 2: The hash-key using SHA256 is extracted from the first divided part of the image that is represented in Fig. 4 by region 1. Region 2 represents the first row of the DCT watermarked image. Then, the extracted Hash-key from region 1 is converted to binary number to have 256 binary bits.

Step 3: The extracted 256 bits are inserted in the first row of the DCT watermarked image. In fact they are inserted in the first 256 pixels of the first row. The method that is used is inserting the hash-key using LSB method in the spatial domain.

Step 4: The image is reconstructed by combining the row with the rest of the image. This results in an image that is watermarked with robust logos and fragile hash-key. Finally, the image is converted back into PDF file.

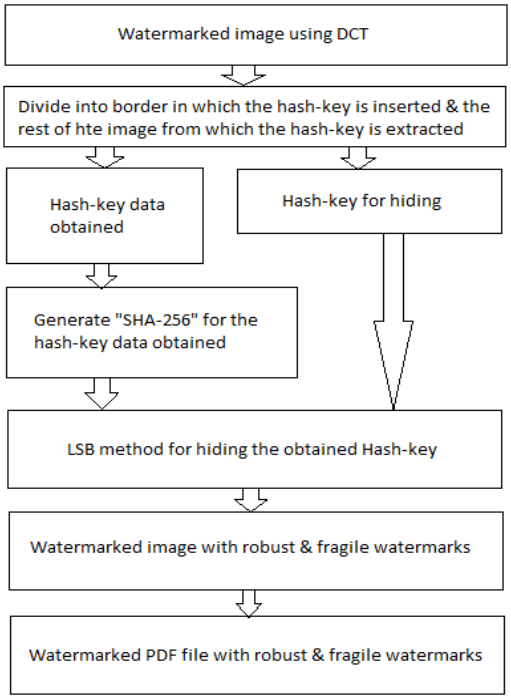

Fig 3: *The embedding of the hash-key*

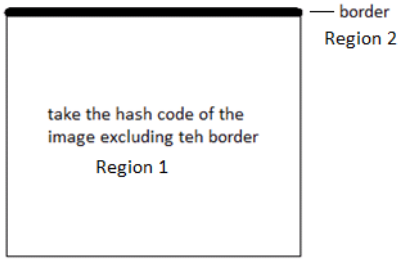

Fig 4: *Regions of extracting & embedding the hash-key*

3. EXTRACTION OF WATERMARKS

3.1 Hash function extraction

The process of extracting the hash key begins with converting the watermarked PDF file into an image file. Then, the first row of the image is cropped and taken to extract the embedded hash-key using the LSB process in the spatial domain. The hash-key is compared to the embedded one that was extracted from region 1 in Fig. 4. If they are equal, the file is authenticated.

3.2 DCT extraction algorithm

After extracting the hash-key, the robust logo watermark is extracted from the multi watermarked PDF file. It is first converted into an image file. Then it is divided into 8x8 blocks and converted into frequency domain using DCT. The watermark logos are extracted using the following equation:

$$Q\left(\frac{F_k(u,v)}{\Delta}\right) \qquad (2)$$

$$\rightarrow \text{if odd, then } w(i,j) = 1$$
$$\rightarrow \text{if even, then } w(i,j) = 0$$

In the previous equation, Q points to the nearest quantization value and Δ shows the scaling factor. Then the logos are extracted, but one logo is needed. So, they are summed to give one logo after deciding a threshold as equation 3 shows:

$$W(i,j) = w1(i,j) + w2(i,j) + w3(i,j) + w4(i,j) \qquad (3)$$
$$\text{if } W(i,j) \geq 3 \rightarrow W(i,j) = 1$$
$$\text{if } W(i,j) < 3 \rightarrow W(i,j) = 0$$

Fig. 6 shows the diagram for the DCT extraction process of the robust watermark.

4. RESULTS

The new algorithm was implemented and tested on a PDF file containing a grey image of Lena and a PDF file containing a scanned Ottoman Painting. After converting the PDF files into an image files, the image file of Lena was found to have the dimensions of 512x512 and the image of the Painting had the dimensions of 1244x972. The inserted logos were numbered as four logos, each one of them has the dimensions of 64x64. Moreover, the hash-key was embedded in the border and extracted from the hash part of the image (region 1) using SHA-256 hash function. Fig. 7 and Fig. 8 show the original PDF file and the converted into image of Lena and the Painting respectively.

The Peak Signal to Noise Ratio (PSNR) is used to show the difference between the original file and the watermarked one. Moreover, the following scaling factors were tested: 4 and 8. The used scaling factor in this paper is 4. Tables 1 and 2 show some analysis.

Table 1**:** *Performance of multi-watermarked file*

DCT robust watermark, scaling factor =4		**Multi watermarked image**
	Grey image	***Hash 'SHA-256'***
IMAGE	***PSNR***	***PSNR***
LENA	*50.4974*	*49.9044*
PAINTING	*57.5862*	*56.9474*

Table 2: *Performance of multi-watermarked file with different scaling factors*

DCT scaling factor	**DCT robust watermark**	**Multi watermarked image**
	PSNR	***PSNR***
4 (LENA)	*50.4974*	*49.9044*
8 (LENA)	*44.5636*	*44.4053*
4 (PAINTING)	*57.5862*	*56.9474*
8 (PAINTING)	*51.7518*	*51.9328*

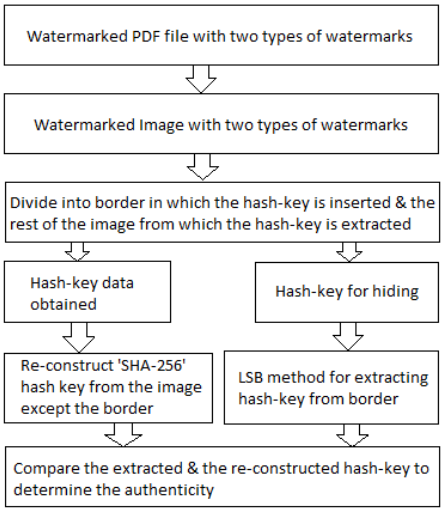

Fig 5: *The extraction process of the hash-key*

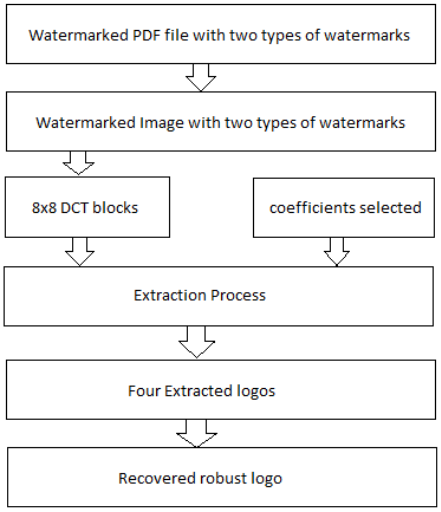

Fig 6: *The extraction process of the logo*

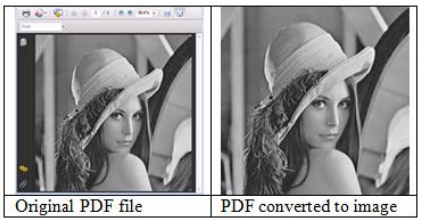

Fig 7: *PDF file and the image file*

Fig 8*: PDF file and the converted image file of the scanned painting*

The obtained Hash-key using SHA-256 Hash function represents a number with a size of 64 hexadecimal. Table 3 shows the Hash-key extracted from region 1 in Fig. 4. Any small or big change or attack on the watermarked file will change the Hash-key of the region 1, making it different from the hash-key inserted in the border. Table IV shows the difference in the hash-key.

Table 3*: Hash-key using SHA-256*

Hash function	Hash size	Image	Hash-key
SHA-256	*64 HEX*	*LENA*	*C919DA648876B0D2F569F42FEE7A5FED4B2F917BA939FB588FCCB39A6FF94A92*
		PAINTING	*DD621A82148AC24AEB97F70040D4AEF11D0CC0414A78AD56871351DF8266181AF*

Table 4 shows that any change or attack on the watermarked image will cause a change in the hash-key that is regenerated from the image.

Table 4: *change in the Hash-key*

Image	regenerated	extracted
LENA	*C919DA648876B0D2F569F42FEE7A5FED4B2F917BA939FB588FCCB39A6FF94A92*	*7F59D0947ED6E377BC9D95AF67B0EAC437C8540C2CACB7DB0E84052339F9A957*
PAINTING	*DD621A82148AC24AEB97F70040D4AEF11D0CC0414A78AD5687135DF8266181AF*	*D785F3773305057A9C160404B0B3BF7C562E40446F217E59F922E340CEC24858*

The robustness of the logo watermarks has been tested under different types of attacks such as JPEG compression and cropping. That means if the PDF file was compressed or cropped, the watermark will still survive the attack. Fig. 9 and Fig. 10 show the extracted watermark that survives different degrees of the JPEG compression for the Lena and Paining files respectively. Moreover, Fig. 12 shows the extracted watermark after cropping the PDF files. That means the files have been cropped four times, each time the files were cropped from different quarter as shown if Fig. 11.

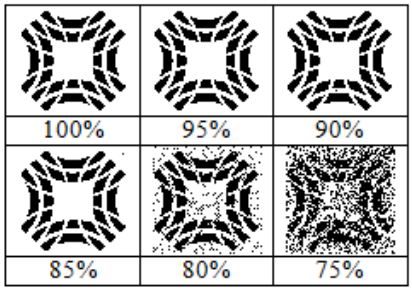

Fig 9: *Extracted Watermark logo from different compression degrees from Lena*

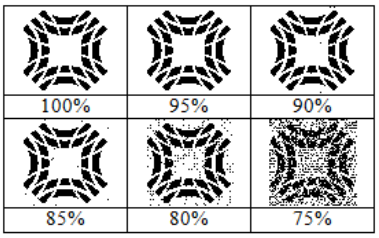

Fig 10: *Extracted Watermark logo from different compression degrees from Painting*

Fig 11: *Regions of Cropping for both files*

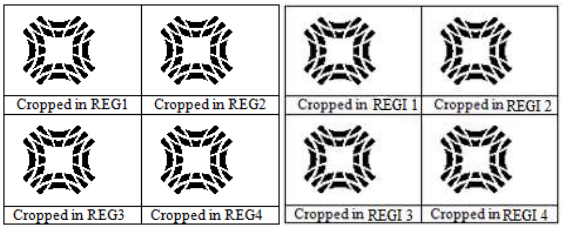

Fig 12: *Extracted watermark from different cropping Lena and Painting files respectively*

4. CONCLUSION

The new algorithm introduces a method of digital watermarking PDF files that uses two different types of watermark signals to embed them in the document. One is robust and is used to prove the ownership of the PDF file. The second watermark is fragile and changed by any type of attacks imposed on the PDF file. This type of watermarking is used to authenticate the file and to detect attacks on the file. The algorithm was tested successfully on several PDF files.

5. REFERENCES

Al-Gindy, A., Al-Ahmad, H., Qahwaji, R. and Tawfik, A. (2007) A New Blind Image Watermarking of Handwritten Signatures Using Low-Frequency Band DCT Coefficients. In proceeding of ICSPC, the IEEE International Conference on Signal Processing and Communications. Dubai, UAE, 1367-1370.

Al-Mansoori, S., and Kunhu, A. (2012). Robust Watermarking Technique based on DCT to Protect the Ownership of DubaiSat-1 Images against Attacks, IJCSNS International Journal of Computer Science and Network Security, Vol.12 No.6, pp. 1-9.

Cannons, J., and Moulin, P. (2004). Design and Statistical Analysis of a Hash-Aided Image Watermarking System, IEEE transaction on image processing, Vol.13, No.10, pp. 1393-1408.

Wang, Z., Bovic, A., Sheikh, H. and Simoncelli, E. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity, IEEE Transactions on Image Processing, pp. 600-612.

Wang, F., Pan, J. and Jain, C. (2009). Innovations in Digital Watermarking Techniques, Springer.