On the steganographic image based approach to PDF files protection

V.N. Gorbachev, L.A. Denisov, E.M. Kaynarova *, I.K. Metelev.

High School of Printing and Mediatechnoilogies
St-Petersburg State University of Industrial Technology and Design

July 7, 2017

Abstract

Digital images can be copied without authorization and have to be protected. Two schemes for watermarking images in PDF document were considered. Both schemes include a converter to extract images from PDF pages and return the protected images back. Frequency and spatial domain embedding were used for hiding a message presented by a binary pattern. We considered visible and invisible watermarking and found that spatial domain LSB technique can be more preferable than frequency embedding using DWT.

Key words: image protection, watermarking, PDF, wavelet, least significant bit.

1 Introduction

The wide spread PDF has the powerful cryptographic tools to protect information in PDF documents. However there is no unique perfect method of protection and various solutions are proposed in this field. The rich structure of PDF allows us to use steganographic techniques for embedding visible and invisible marks that protect digital data. Numerous approaches consider varying lines or words, spacing, font characteristics as well as varying certain invisible characters as a cover to hide a secrete message. The developed White Space Coding technique is based on the fact that there are a lot of white-space characters separating syntactic constructions one from another [1]. Visually they are undistinguished or invisible and may be used as covers. As a result a message bit can be encoded by two items: as the normal or as a non-breaking space. White Space Coding may include a version of RSA encryption on quadratic residual [2]. This modification is suitable for secrete communication via PDF files when a message is encoded by a between-word character location. Additionally such operators as the justified text operator TJ can be considered for data hiding [3]. Indeed, we often make the text justified so that the right margin is not ragged. This edit format results in random position of each character. A stream of such positions generated by a TJ operator is a good cover for data hiding using LSB (Least Significant Bit) embedding [4].
An example is a (t,n) secrete sharing scheme. It may be used to share secrete images [5] and to protect

*helenkainarova@gmail.com

PDF files [6]. To protect a very important PDF file it is decomposed into n parts. Each of them is embedded into its PDF cover. All stego covers look equally. So, the PDF file is shared among n parties and any t parties can recover it if they will cooperate. Indeed, a standard option such Attach File, available at Adobe Acrobat Pro can be suitable for embedding [6].
A digital watermark can be encoded by a self inverting permutation, i.e. a permutation which cycles have length less or equal 2. Using a particular representation, named 1d and 2d, the permutation is embedded into a 1d and 2d array. This technique was developed for audio files and images and generalized for PDF files [7]. The key idea is that the self-inverting permutation allows to locate the watermarked area to recover the information instead of to extract the message directly. In the case of frequency domain watermarking this method is robust to JPEG lossy compression [8].
Numerous types of the PDF files raise a large number of solutions. [9]. Indeed, now the Adobe Acrobat has options to embed a visible watermark. It protects important documents sometimes printed on a background containing large gray digits or such inscription as "watermarked". This makes the content perception worse and the interested user can remove the watermark using a key for an additional fee. However Adobe Acrobat uses a simple technique and numerous suggestions on how to delete the embedded data can be found in the Internet.

In our paper we focus on protection of images placed in a PDF files and considering two steganographic schemes. The key idea is to extract the desired image and return it back in the file after embedding watermark. We introduce a convertor that transforms PDF to SVG format and back [1]. SVG stores the image in PNG format, so a png cover is prepared. The aim of our paper is to investigate a spatial and frequency domaine watermarking for images in PDF file. We consider LSB and DWT techniques for invisible and visible and removal watermarks. It was found that the embedding in a spatial domain is more preferable. It introduces less degradation in case of visible and invisible watermarking. For spatial watermarking we found a nice quality of the image retrieved after removing visible watermark, its degradation is less than for similar DWT technique developed in [10].
The paper is organized as follows. First we consider our scheme including convertor, then introduce the frequency and spatial domain embedding schemes.

2 Scheme

Setup. The scheme has a PDF-SVG convertor and watermarking algorithms, that allows us to embed invisible watermarks into image of PDF document and also visible watermarks that can be removed. It works as follows.

- The convertor extracts an image from the PDF document
- a watermark is embedded (detected/ removed) into the extracted image
- the convertor returns the image back into the PDF document.

Some details are presented in Figure (1) (a). An image A is extracted from PDF and stored in PNG or JPEG format using the PDF-SWG convertor. So we get a cover image C that is watermarked. We consider a binary image M as a watermark The watermarked image S is returned back into PDF by the convertor. The protected image B appears in the PDF document. To get the embedded information the convertor extracts B, stores it in graphical format and either detect watermark M' or retrieves cover

[1] Available at http://pdf.welovehtml.ru

work as CR. The obtained M' and CR may differ from its originals because of errors caused by numerous transformations. So, the problem is to achieve the indistinguishability

$$M \approx M',$$
$$CR \approx C.$$

Distortion measures. To analyze the scheme we introduce the distortion measures between the original and the extracted watermarks $d(M, M')$ and between the original and the retrieved cover images $d(C, CR)$. We will use standard measures as RMSE (Root Square Error), PSNR (Peak Signal Noise Ratio) and relative entropy known also as Kullback - Leubler entropy. For particular case of two binary images M and M' we introduce Hamming distance $ham(M, M')$, There are two reasons for it. The first one says that PSNR and RMSE are not well agreed with visual perception for binary images. The second one says that, Hamming distance has a clear meaning, that is number of errors or number of different bits in two messages.
So we introduce the Hamming distance

$$ham(M, M') = (1/\Omega) \sum_{mn} h_{mn}, \tag{1}$$

where the matrix of errors h_{mn} is defined by

$$h_{mn} = 1, \quad M_{mn} \neq M'_{mn}.$$
$$h_{mn} = 0, \quad M_{mn} = M'_{mn}.$$

where Ω is a total number of pixels. It follows from (2) that $ham(M, M')$ is the relative number of errors in the extracted watermark M'. In case of binary date Hamming distance can be expressed by PSNR or Euclidian metric

$$RMSE(M, M') = \sqrt{(1/\Omega) \sum_{mn} (M_{mn} - M'_{mn})^2}$$

If M and M' are binary matrix, then

$$RMSE^2(M, M') = ham(M, M').$$

Convertor. The proposed convertor transforms PDF document into a set of SVG, PNG, JPEG files and back. This solution allows us to introduce the steganographic techniques for image watermarking. We need a reversible convertor not to loose information. Let us consider the transformation presented in Figure (1) (b)

$$\mathsf{PDF} \to \mathsf{a.png} \to \mathsf{PDF} \to \mathsf{b.png} \to \mathsf{PDF} \to \mathsf{c.png},$$

where image *a.png* is extracted from the PDF document and converted step-by-step. Operations are reversible if

$$\mathsf{a.png} = \mathsf{b.png} = \mathsf{c.png}. \tag{2}$$

Indeed, in the Internet there are a large number of services for converting PDF to PNG and PNG to PDF. Two random convertors were chosen [2] and a PNG image was processed. It was a standard 8-bit grayscale image in integer coding $u_8 = 0, 1, \ldots, 255$. The difference $a - b$ was found to be up to 129. Such

[2] http://pdf-png-jpg.eu/, http://online2pdf.com/convertor-png-to-pdf

difference in the pixel brightness can be visible in the eye.
We checked (2) for our convertors using 20 pdf documents and found all png images a, png, b.png and c.png to be equal. It tells us that our convertor is reversible at least for date in integer encoding. This is important for our scheme because errors should be introduced by embedding algorithms and any transformations except PDF-SVG conversion.

3 Frequency DWT embedding

For frequency domain watermarking a one level DWT with orthogonal wavelets was used.

Algorithm. The frequency embedding scheme is presented in Figure (2). The algorithm has the next steps.

- Transform cover image C into four blocks of DCT coefficients cA, cH, cV and cD known as approximal, horizontal, vertical and diagonal details or LL, LH, HL and HH frequency bands.
- Choose a block Z from cA, cH, cV and cD and replace a part of its coefficients with M.
- Create the watermarked PNG image S using inverse DWT.

To detection M we need to inverse steps. In Figure (2) (a) any transformation of S is denoted as $T : S \to S'$. This operator considers storing image in PNG file, sending it and converting. Generally $S \neq S'$ because of the introduced errors.
The embedding algorithm, Figure (2) (b), can replace a part, u, of the block Z with binary image M

$$uZ \to aM$$

where a is a scaling parameter, describing brightness of the watermark. Let us assume, that $u = 0.9$, $Z = cD$, and $a = 2$, so 90% coefficients of the block cD will be replaced with M whose brightness is increased twice. The brightness of watermark plays an important role. The introduced changes depend on a and may result in a visible or an invisible watermark. To get an invisible watermark we need a low value a that can be established experimentally.
Experiment. A scheme is focused on invisible watermarks presented by a binary image. The main parameters are a cover image and a watermark, the type of wavelet and scaling parameters, $\{C, M, w, a, u\}$. We used the orthogonal wavelets of Daubeshies family *db* and *sym* that have good features as for smoothness. For embedding the $Z = cD$ block was chosen.

- **Brightness of watermark.** The Figure (3) presents a binary watermark (a) and two fragments of PDF document that include a grayscale and a color images (b) and (d). [3] The watermarked images are shown in Figure (4). They were obtained by the wavelet db1 and $u = 0.7$. The images were extracted from pdf-documents after embedding with $a = 50$ and $a = 150$. In the case of $a = 150$ the watermark is visible. We can choose a lower value a to get invisible watermark.

- **Type of wavelet.** Figure (5) illustrates watermarking by various Daubeshies wavelets. A fragment of the PDF document presented in Figure(5) (a) [4] is protected by an invisible watermark shown

[3] These fragments are from paper by Y-C Lai, W-H Tsai, Covert communication via PDF files by new data hiding techniques, NSC project No, 97-2631-H-009-001and from [8]

[4] A page from paper by A. A. Ali, Al-H S Saad, New text steganography technique by using ma[ed-case-font, International Journal of Computer Application, v, 63, no, 3, 2013.

in Figure (5) (b). For this case $a = 20$, $u = 0.5$.The extracted watermarks that were embedded with various wavelets and Hamming distance $ham(M, M')$ are presented in Figures (c) - (e). We considered $db1$, $db2$, $db6$, $db28$, $db41$ and found the Hamming distance to be $ham = 0.0223$, 0.1414, 0.1513, 0.1102, 0.1016. It tells us that there were errors and up to 15% of pixels were restored incorrectly. Nevertheless extracted watermarks look good. The same result is true for wavelets from sym family.

- **Detection errors.** Degradation of watermarks is caused by nonreversible transformations and may depend on all the parameters $\{C, M, w, a, u\}$. We calculated a set of distortion measures to study degradation from parameter a that is the brightness of embedded watermarks. Hamming distance, ham, and relative entropy, $relent$, are presented in Figure (6) for wavelets $db1$ and $db6$. The measures tell that the larger a is the greater degradation. For wavelet $db1$ errors given by ham equal about 0.003% and cause for $db6$ about 3% .

  It was found that the error is 0 if watermarked images weren't stored in PNG format before detection.

4 Spatial embedding

For spatial embedding the cover image bit planes can be used to get visible and invisible watermarking.

Algorithm. The main idea of our algorithm is to use a cover grayscale image that has two identical bit planes [11]. This solution allows us to apply blind detection of invisible watermarks and allows us to retrieve the original image after removing the visible watermark.
Any 8-bit grayscale image C can be represented by its bit planes B_V, $V = 1, \ldots, 8$. Each plane has its weight 2^{V-1} and has its semantic information. Let the least significant bit plane B_U, say $U = 1, 2$, be replaced with B_V, where $V > U$. Then the obtained image C_2 has two identical planes B_V

$$C_2 : \texttt{bitget}(C_2, V) = \texttt{bitget}(C_2, U) = B_V,$$

where the function `bitget` calculates a given bit plane of a grayscale image.

The embedding algorithm has two steps:

- create a cover image C_2 with two planes B_V,
- add a binary watermark M to bit plane B_V by modulo 2

$$C_2 \to S = C_2 - B_V 2^{V-1} + (B_V \oplus M)2^{V-1}.$$

Invisible watermark can be achieved, if the embedded plane B_V is not a significant plane, e.g. $V = 2, 3$. The detection is blind, it needs the stego image only

$$S \to M = \texttt{bitget}(S, V) \oplus \texttt{bitget}(\texttt{S}, \texttt{U}).$$

As for a visible watermark we need to choose a significant bit plane B_V, e.g. $V = 7, 8$. To remove the watermark the embedded plane is replaced with its copy

$$S \to CR = S - \texttt{bitget}(S, V)2^{V-1} - \texttt{bitget}(S, U)2^{U-1} + \texttt{bitget}(S, U)2^{V-1}. \quad (3)$$

Clear, that the difference between the cover image C and CR is the least significant plane B_U. This is the main source of errors. It may be extremely small and visually we may have $C \approx CR$.

Experiment.

- **Invisible watermark**.The considered spatial algorithms keep the image in the 8 bit integer encoding. There is no loss of information when digital image is stored in the PNG format. In case of invisible watermarks we can get no degradation of the extracted data by keeping all the transformations reversible.

- **Visible and removal watermark.** Figure (7) shows the watermarked fragment of a PDF document. The bit plane $V = 7$ was used. This plane was storied earlier as the least significant plane $U = 1$. Figure (7) (b) and (c) present the retrieved images CR placed in his PDF documents and its digital versions. Both retrieved images look nice and they are visually undistinguished from their originals. To find a difference between C and CR we consider PSNR. In accordance with (3) the difference is defined as B_U. Figure (7) (d) presents PSNR calculated between cover image C and C_U that is the cover image without plane $U = 1, 2, \ldots$, In our case $U = 2$ and PSNR=42.3417 db. This corresponds to PSNR between C and CR that was found from experiment. Note, the large PSNR is in agreement with visual quality. As it follows from the Figure (d), PSNR of about 50 db is achieved if the cover bit plane is copied into the least significant plane $U = 1$.

- **Watermarking of text.** Our convertor has an option to recognize a PDF page with text as an image. Then the PDF text is converted into PNG format and can be watermarked with the help of the considered technique. As a result the PDF text is protected as the Figure (8) shows.

5 Conclusions

For frequency domain embedding we analused DWT and found that the main reason of errors is averaging when the digital image is storied in a graphical format. We used PNG format requiring 8-bit integer encoding.
The introduced spatial domain embedding turned out to be free of such errors. This technique is based on bit planes of cover images and allows to embed invisible and visible watermarks. We introduce an algorithm that can double the bit planes of gray scale cover image. As a result a visible watermark can be removed and the cover image can be retrieved with high visual quality.

References

[1] Wang J. T. and Tsai W. H. Data hiding in PDF files and applications by imperceptible modifications of PDF object parameters. Proc. of 2008 Conf. on Computer Vision, Graphics & Image Proc., Yilan, Taiwan, Aug. 24-26, 2008.
I-Shi Lee and Wen-Hsiang Tsai. A new approach to covert communication via pdf files. Signal Processing, 90, p. 557–65, 2010. 1

[2] Lin, H. F., Lu, L. W., Gun, C. Y., Chen, C. Y. A copyright protection scheme based on PDF. Int J Innov Comput Inf Control, 9(1), 1-6, 2013. 1

[3] Zhong S, Cheng X., Chen T. Data hiding in a kind of pdf texts for secret communicationl. International Journal of Network Security, v. 4, p. 17–26, 2007. 1

[4] Alizadeh-Fahimeh, F., Canceill-Nicolas, N., Dabkiewicz-Sebastian, S., Vandevenne-Diederik, D. Using Steganography to hide messages inside PDF files. SSN Project Report, 2012. 1

[5] Thien C.C., Lin J.C. Secret image sharing. Computers and Graphics, vol. 26, no. 5, (2002), pp. 765-770. 1

[6] Suiang-Shyan Lee, Shuo-Fang Hsu and Ja-Chen Lin. Protection of PDF Files: a Sharing Approach. International Journal of u- and e- Service, Science and Technology Vol.7, No.2 (2014), pp.27-40. 2

[7] Chroni M., Nikolopoulos S. D. Watermarking PDF Documents using Various Representations of Self-inverting Permutations. arXiv:1501.02686 [cs.MM] (2015). 2

[8] Chroni M., Fylakis A. , Nikolopoulos S., D. Watermarking Images in the Frequency Domain by Exploiting Self-Inverting Permutations. Journal of Information Security, 2013, 4, p. 80-91. 2, 4

[9] Subhedara M. S., Mankar V. H. Current status and key issues in image steganography: A survey. Computer Science Review, V. 13-14, p. 95-113, 2014. 2

[10] Hu Y, S. K., J. Huang S.K. An algorithm for removable Visible Watermarking. IEEE. Transactions on circuits and systems video technology, v. 16, No 1, p.129-133, 2006 3, 10, 13 2

[11] Gorbachev V.N., Denisov L.A., Kaynarova E.M. Embedding of binary image into Gray planes. Russian Computer Optics, Russian Academy Science, v 37, p, 385, 2013. 5

[12] Y-C Lai, W-H Tsai, Covert communication via PDF files by new data hiding techniques, NSC project No, 97-2631-H-009-001.

[13] Meral H.M., Sankur B., Ozsoy A. S., Gungor T.,Sevinc E. Natural language watermarking via morphosyntactic alterations, Computer Speech & Language Journal, Volume 23, Issue 1, January 2009, Pages 107-125. 1, 12.

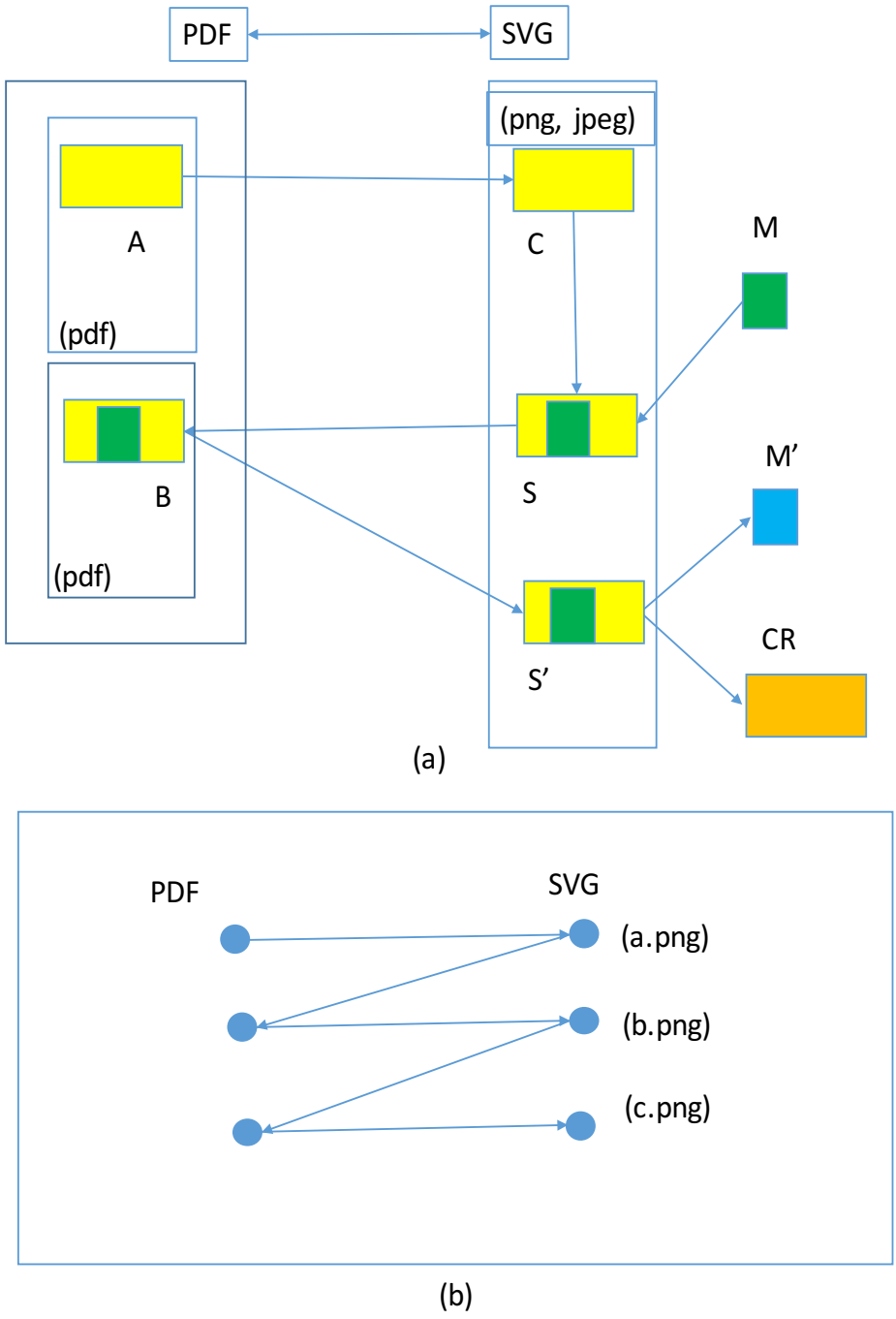
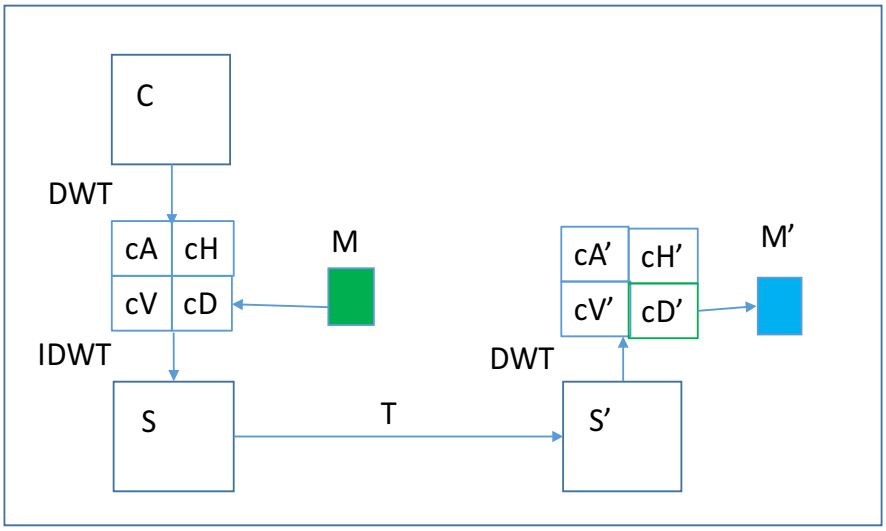
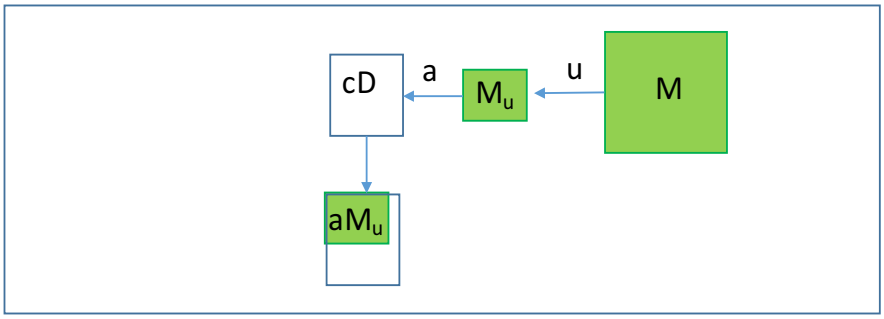


Figure 1: Image-based watermarking of PDF. a) Image A is extracted from PDF document by convertor $PDF - SVG$, the image is stored in PNG format and watermarked. Convertor returns the watermarked image back. Two pattens M and M' are embedded and extracted watermarks, CR is the retrieved original after removing watermark. b) Conversion $PDF - SVG$.

W

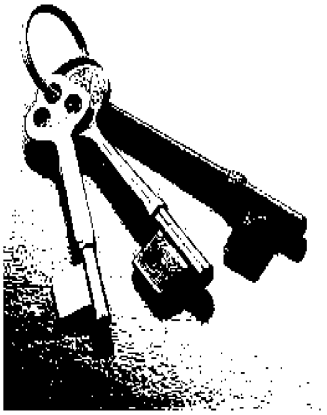


(a)



(b)

Figure 2: Frequency domain embedding. a) DWT coefficients of cD block are replaced which binary image M, a watermark, that is detected after some transformations T. b) Scaling of watermark W



(a)

1. A one-line *header* identifying the version of the PDF specification to which the file conforms.
2. A *body* consists of a sequence of indirect objects representing the contents of the PDF document.
3. A *cross-reference table* containing information about the indirect objects in the PDF file, such as the offset of each object which give the object's location in the displayed document.
4. A *trailer* giving the location of the cross-reference table and those of certain special objects within the body of the file.

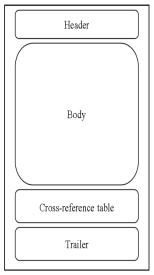


Fig. 1. Structure of a PDF file.

In a PDF file, *white-space characters* are used to separate syntactic constructs from one another.

will be described in Section 3.2.

Table 1. White-space characters in PDF files.

decimal	hexadecimal	name
0	00	null (NUL)
9	09	tab (HT)
10	0A	line feed (LF)
12	0C	form feed (FF)
13	0D	carriage return (CR)
32	20	Space (SP)

3. DATA HIDING TECHNIQUES FOR SECRET TRANSSMISSION VIA PDF FILES

3.1 Data Hiding in PDF Files by White Space Coding

According to Table 1, a white-space character may described by one of six different codes, namely, the hexadecimal number 00, 09, 0A, 0C, 0D, or 20. After some experiments, we found out in this study that in some text editors, 0C will show as a line and 0A will cause line feeding, in the displayed document of a PDF file containing these two codes. If we use these two codes to embed secret messages, people will be aware of the existence of the hidden data easily by inspecting the displayed PDF document. So we only use 00, 09, 0D and 20 to embed the secret message in this study.

Furthermore, we know that all white-space characters are equivalent except in comments, strings, and streams, so if 00, 09, 0D and 20 are not in

(b)

3.2. Main Points

First we should mention that for images with general characteristics and relatively large size this method delivers optically good results. By saying "good results" we mean that the modifications made are quite invisible. Also the method's algorithms run really fast as they simply access a finite number of pixels. Furthermore, both the embedding and extracting algorithms are easy to modify and adjust for various scenarios.

On the other hand, the method fails to deliver good results either for relatively small images or for images that depict something smooth which allows the eye to distinct the modifications on the image. Also we decided to move to a new method as there were also problems due to the fact that the positions of the crosses are centered at strictly specific positions causing difficulties in the extracting algorithm even for the smallest geometric changes such as scaling or cropping where we may lose the marked positions.

4. The Frequency Domain Approach

Having described an efficient method for encoding inte-

Output: the watermarked image I_w;

Step 1: Compute first the 2DM representation of the permutation π^*, *i.e.*, construct an array A^* of size $n^* \times n^*$ such that the entry $A^*(i, \pi_i^*)$ contains the symbol "•", $1 \le i \le n^*$.

Step 2: Next, calculate the size $N \times M$ of the input image I and cover it with an imaginary grid C with $n^* \times n^*$ grid-cells C_{ij} of size $\left\lfloor \frac{N}{n^*} \right\rfloor \times \left\lfloor \frac{M}{n^*} \right\rfloor$, $1 \le i, j \le n^*$.

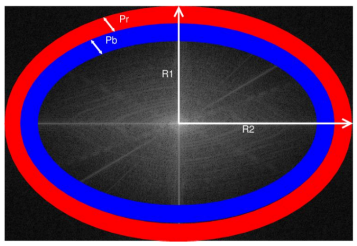
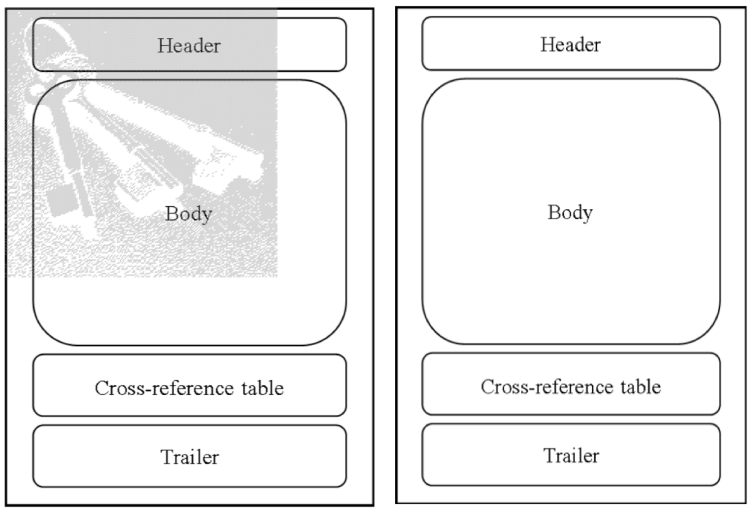


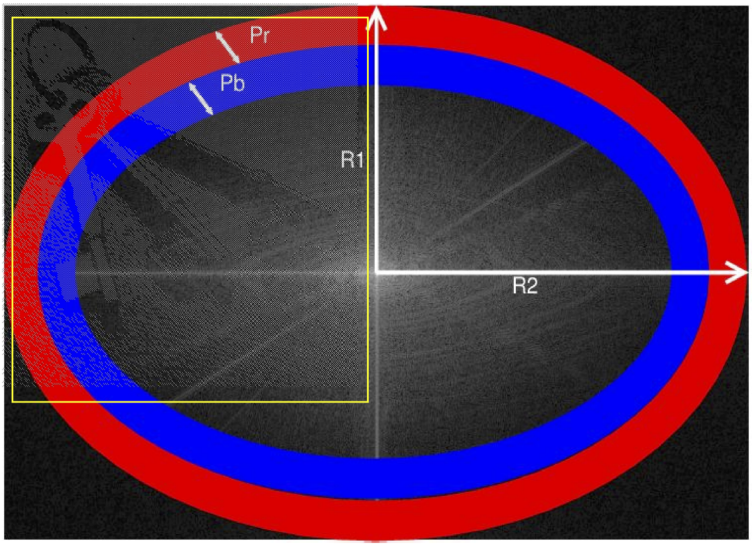
Figure 2. The "Red" and "Blue" ellipsoidal annuli.

(c)

Figure 3: Frequency watermarking of images from PDF document. a) Binary image that is watermark, b) c) two fragments of PDF documents with grayscale and color image.
W



(a) a=150, u=0.7 (b) a=50, u=0.7



(c) a=150, u=0.7

Figure 4: Watermarked Images from PDF pages in Figure (3). a), b), c) Visible and invisible watermarks. W

inverse of the embedding process, where the secret message is revealed at the end, [11].

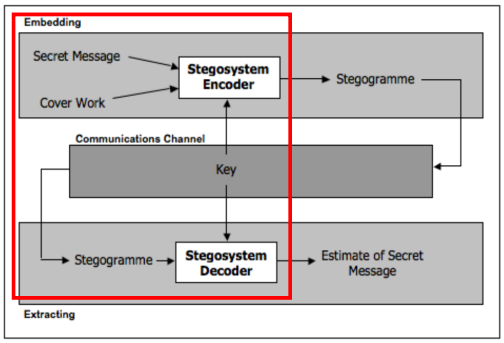


Fig. 1: Basic steganography system[11]

Figure (1) shows one example of how steganography might be used in practice. Two inputs are required for the embedding

(a)

(b)

(c) db1, ham=0.0223

(d) db2, ham=0.1414

(e) db6, ham=0.1513

Figure 5: Watermarking by Daubeshies wavelets. a) A fragment of a PDF page with image protected using invisible watermark, $a = 20$, $u = 0.5$, b) watermark, c), d), e) extracted watermarks, embedded with the wavelet $db1$, $db2$, $db6$.
W

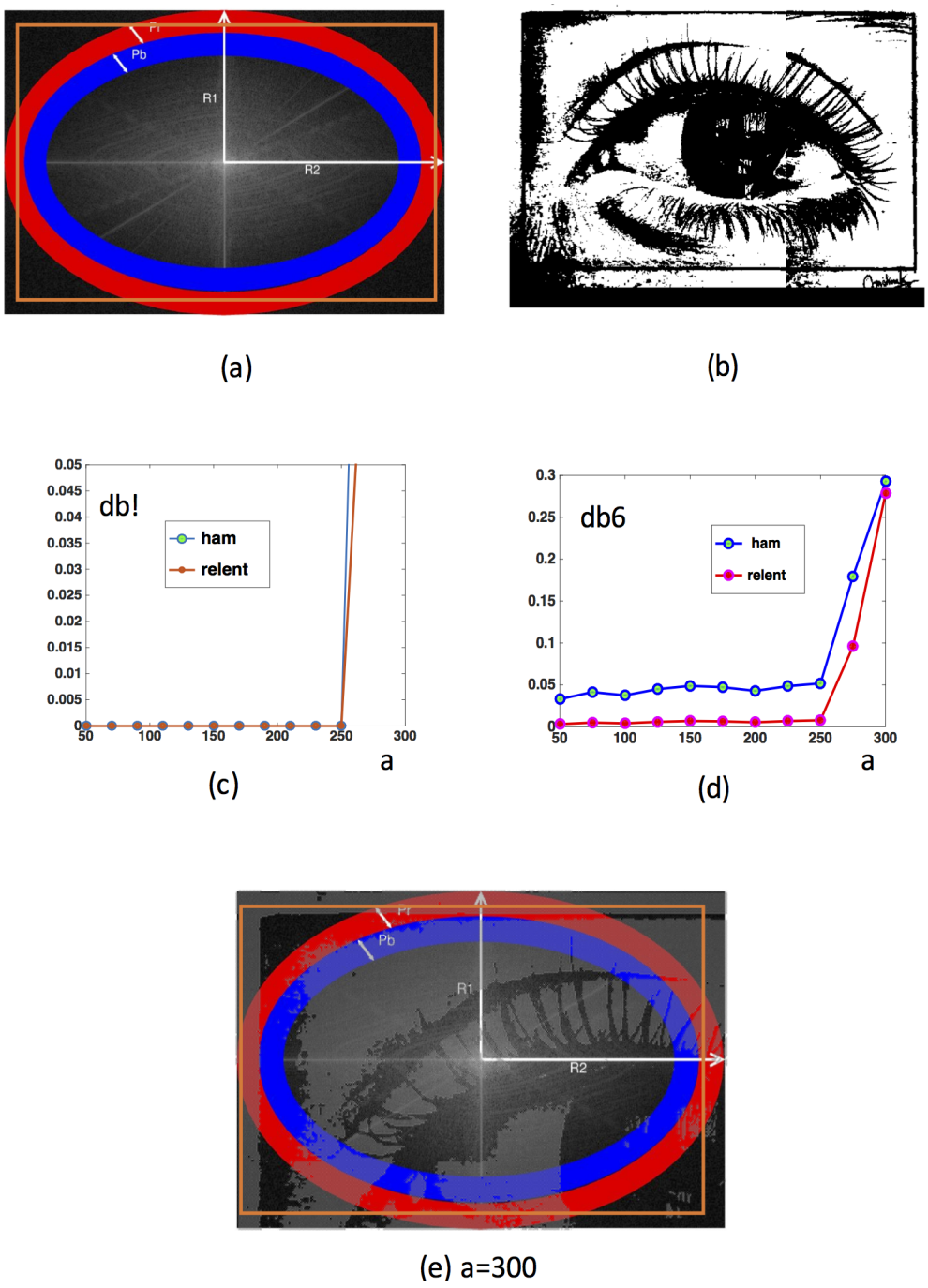


Figure 6: Distortion measures. Hamming distance, `ham`, and relative entropy, `relent`, vs brightness of watermarks. a) Cover image, b) binary watermark, c) and d) distortion measures for wavelet db1 and db6, e) watermarking with hight brightness a=300.
W

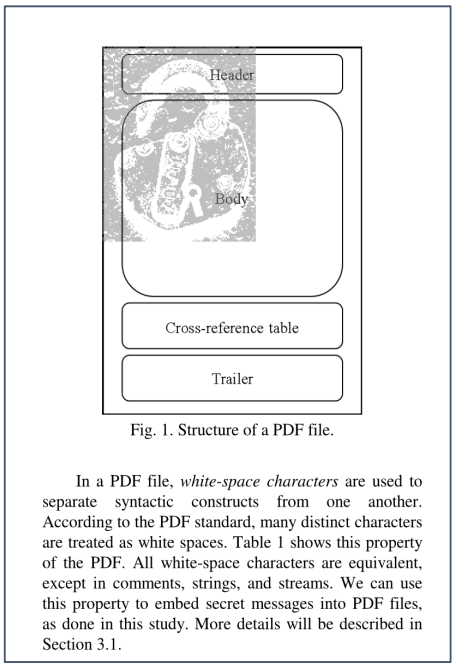
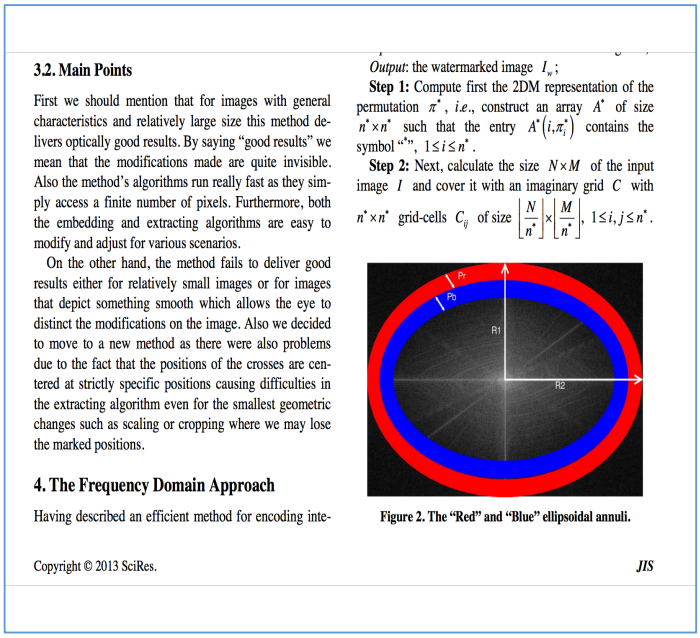


Figure 7: Visible and removal watermark. a) Watermarked image in PDF document, $V = 7$, $U = 2$, b) the image placed in PDF after removing the watermark, c) fragment of retrieved image, d) PSNR of cover image witout a bit plane $U = 1, 2, \ldots 8$,
W



3.2. Main Points

First we should mention that for images with general characteristics and relatively large size this method delivers optically good results. By saying "good results" we mean that the modifications made are quite invisible. Also the method's algorithms run really fast as they simply access a finite number of pixels. Furthermore, both the embedding and extracting algorithms are easy to modify and adjust for various scenarios.

On the other hand, the method fails to deliver good results either for relatively small images or for images that depict something smooth which allows the eye to distinct the modifications on the image. Also we decided to move to a new method as there were also problems due to the fact that the positions of the crosses are centered at strictly specific positions causing difficulties in the extracting algorithm even for the smallest geometric changes such as scaling or cropping where we may lose the marked positions.

4. The Frequency Domain Approach

Having described an efficient method for encoding inte-

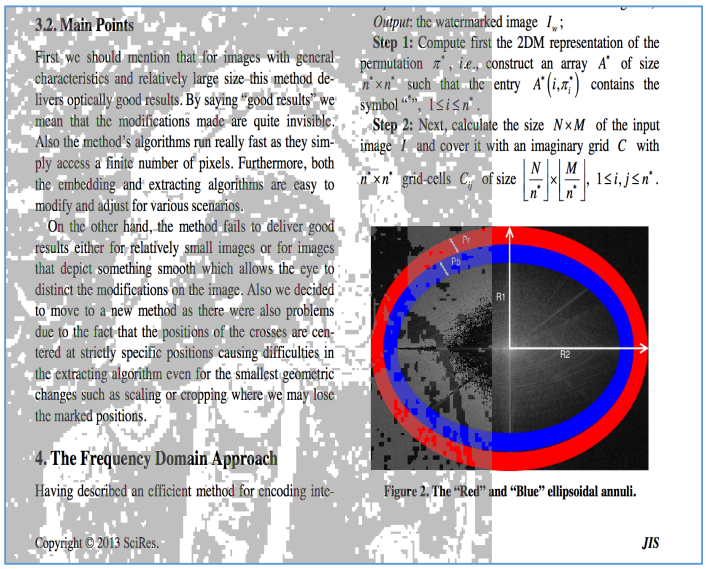
Output: the watermarked image I_w;

Step 1: Compute first the 2DM representation of the permutation π^*, *i.e.*, construct an array A^* of size $n^* \times n^*$ such that the entry $A^*\left(i,\pi_i^*\right)$ contains the symbol "*", $1 \le i \le n^*$.

Step 2: Next, calculate the size $N \times M$ of the input image I and cover it with an imaginary grid C with $n^* \times n^*$ grid-cells C_{ij} of size $\left\lfloor \frac{N}{n^*} \right\rfloor \times \left\lfloor \frac{M}{n^*} \right\rfloor$, $1 \le i,j \le n^*$.

Figure 2. The "Red" and "Blue" ellipsoidal annuli.

(a)



3.2. Main Points

First we should mention that for images with general characteristics and relatively large size this method delivers optically good results. By saying "good results" we mean that the modifications made are quite invisible. Also the method's algorithms run really fast as they simply access a finite number of pixels. Furthermore, both the embedding and extracting algorithms are easy to modify and adjust for various scenarios.

On the other hand, the method fails to deliver good results either for relatively small images or for images that depict something smooth which allows the eye to distinct the modifications on the image. Also we decided to move to a new method as there were also problems due to the fact that the positions of the crosses are centered at strictly specific positions causing difficulties in the extracting algorithm even for the smallest geometric changes such as scaling or cropping where we may lose the marked positions.

4. The Frequency Domain Approach

Having described an efficient method for encoding inte-

Output: the watermarked image I_w;

Step 1: Compute first the 2DM representation of the permutation π^*, *i.e.*, construct an array A^* of size $n^* \times n^*$ such that the entry $A^*\left(i,\pi_i^*\right)$ contains the symbol "*", $1 \le i \le n^*$.

Step 2: Next, calculate the size $N \times M$ of the input image I and cover it with an imaginary grid C with $n^* \times n^*$ grid-cells C_{ij} of size $\left\lfloor \frac{N}{n^*} \right\rfloor \times \left\lfloor \frac{M}{n^*} \right\rfloor$, $1 \le i,j \le n^*$.

Pr
Pb
R1
R2

Figure 2. The "Red" and "Blue" ellipsoidal annuli.

Copyright © 2013 SciRes. *JIS*

(b)

Figure 8: Watermarking of PDF text. a) A fragment of PDF text, b) the watermarked fragment. W