ISBN 978-952-5726-02-2 (Print), 978-952-5726-03-9 (CD-ROM)
Proceedings of the 2009 International Symposium on Information Processing (ISIP'09)
Huangshan, P. R. China, August 21-23, 2009, pp. 075-078

Steganalysis Against Equivalent Transformation Based Steganographic Algorithm for PDF Files

Shangping Zhong, Xin Fang, and Xiangwen Liao
Department of Computer Science and Technology, Fuzhou University, Fuzhou, China, 350108
Email: {spzhong, N070320066,liaoxw}@fzu.edu.cn

Abstract—**The equivalent transformation based steganographic algorithm for PDF files shows good performance, both in capacity and invisibility. In addition, data embedding by this algorithm need not change the size of a cover-file. However, a loophole exists in the steganographic scheme. Through analyzing the structure of a PDF file, we can reconstruct a PDF file by using the software tool according to the obtained "Creator" key, and reveal the presence of secret data. Furthermore, by making use of the fact that the residue sequence distributes uniformly, this paper's steganalysis scheme can also estimate the length of hidden data. To enhance security, in this paper, it is proposed to introduce a "random moduli" equivalent transformation based steganographic algorithm. It not only has more security, but also has larger capacity. Adopted practical PDF files, computing results show that this paper's steganalysis scheme and "random moduli" equivalent transformation based steganographic algorithm are all effective. Moreover, it is possible to extend this paper's methods to some other data sets, e.g., webpage structure text files.**

Index Terms—**Steganalysis, equivalent transformation based steganographic algorithm, reconstruction attack, random moduli, PDF file.**

I. Introduction

PDF is a file format used to represent a document in a manner independent of the application software, hardware, and operating system used to create it. A PDF file contains a PDF document and other supporting data. A PDF document contains one or more pages. Each page in the document may contain any combination of text, graphics, and images in a device- and resolution- independent format. A PDF document may also contain information possible only in an electronic representation, such as hypertext links, sound, and movies, etc.[1]. Because of the merits of PDF documents, they have become important interchange information among diverse products and applications.

Accordingly, the technique of data hiding has been introduced into PDF documents (e.g.[2]-[5]). Unlike an image, a PDF document has little redundancy information for secret communication. Liu et al.[2] proposed a novel PDF document steganographic algorithm, in which equivalent objects transformation is used to embed data to the cover-object. The algorithm can obtain perceptual transparency, and need not change the size of a PDF document. In [3], the steganographic method has been proposed by varying the line or word or character spacing or by varying certain character features slightly. In [4], arbitrary length data can be embedded between two adjacent objects. The secret channel of the integer numbers in the "TJ" operator string has been used to hide data [5]. This steganographic method is secure, and allows for obtaining high payload. Unlike the method [2] and [5], the method [3] and [4] need change the size of the cover-object to embed secret data and can not prevent statistical attacks.

As we know, steganalysis is the set of techniques that aim to distinguish between cover-objects and stego-objects, or go one step further and estimate some parameters of the embedded message such as its length, location, etc.. Several approaches have been proposed to solve the image steganalysis problem and we can broadly classify them into the following groups[6]: Supervised learning based steganalysis (e.g.,[7]), Blind identification based steganalysis (e.g.,[8]), Parametric statistical steganalysis (e.g.,[9]) and Hybrid techniques. Each of these methodologies has pros and cons. Therefore, it is up to the user (steganalyst) to choose an appropriate methodology [6]. To our knowledge, there is no work which focuses on the steganalysis scheme against equivalent transformation based steganographic method for PDF files[2].

In this paper, we propose a steganalysis method which attacks and successfully identifies the existence of embedding done by the equivalent transformation based steganographic method for PDF files[2]. Our steganalysis method can even estimate the length of payload size. In addition, to enhance security, a modified scheme is proposed. In section II of the paper, the equivalent transformation based steganographic method for PDF files is briefly reviewed. Our proposed steganalysis method is presented in section III. Section IV presents the modification to the equivalent transformation based steganographic method. In section V, simulation results are presented. Conclusion of the paper is found in section VI.

II. Equivalent Transformation Based Steganographic Method for PDF Files

In the equivalent transformation based steganographic method for PDF files[2], a covert channel is discovered based on the following fact: the effect of page display of PDF file is extraneous to the seriation of entries. Thereby, covert information embedded in different objects can be achieved by special array of entries, instead of by operation of adding any other data to the cover-object. Moreover, the large number of existing dictionaries and stream objects in PDF files ensures the large capacity required of covert information embedded[2].

A Embedding Algorithm

Step 1: Input the embedded data to produce the key and the initial value x_0 of Logistic Chaotic Map[11] ;

Step 2: Ransack the PDF file, to obtain E_j with every $O_j(\alpha_{n_j}), 1 \le j \le M$;

Step 3: Obtain the maximum capacity $N_{\max} = \prod_{j=1}^{M} n_j! - 1$;

Step 4: Change the binary bit string which is to be embedded to integer N , if $N_{\max} < N$., then the embedding fails; otherwise, Let $j = 1$, goto Step 5;

Step 5:Let $N' = N/(n_j!), t_i = N \bmod (n_j!)$;

Step 6: Let $N = N', j = j+1$, if $j \le M$, goto Step 5, otherwise , goto Step 7;

Step 7: Generate T , make equivalent transform to every valid object in cover-object according to T .

Step 8: Produce Logistic chaotic map sequence with x_0 as the initial value ($x_n = \mathrm{sgn}(x_k) = \begin{cases} 1, x_k \ge 0.5 \\ 0, x_k < 0.5 \end{cases}$, x_k is produced by Logistic chaotic map .The algorithm uses this random binary sequence to decide whether or not to equivalently transform entries .), make equivalent transform to every valid object again according to the embedding rule to scramble the sequence of entries.

Step 9: Output the PDF file.

B Extracting Algorithm

Step1: Input the key, and then generate the chaotic sequence with the initial value x_0 ;

Step2: Ransack the PDF file, obtain M and H, carry on the restoration of each valid object in accordance with the extraction rule, and Let $j = 1$;

Step3: According to $O_j(\alpha_{n_j}^k)$, obtain $\left|O_j(\alpha_{n_j}^k)\right| = t_j$;

Step4: Let $j = j+1$, ,if $j \le M$, , goto Step3, otherwise goto Step5;

Step5: Obtain $T' = (t_1 ... t_j ... t_M)$ ， because mapping $|\bullet|$ is bijective mapping， $T' = T$， meanwhile, the value of the H has nothing to do with the Algorithm ， and then $f(T', H) = f(T, H) = N$;

Step6: Produce 160-bits digest through SHA-1, if it is the same with the key, then output extraction information, otherwise the data is changed or destructed in the dissemination process.

More details of the equivalent transformation based steganographic method can be found in [2].

III. STEGANALYSIS AGAINST EQUIVALENT TRANSFORMATION BASED STEGANOGRAPHIC METHOD FOR PDF FILES

A. Revealing the Presence of Secret Data

As we know, PDF files may be generated either directly from applications or from files containing PostScript page descriptions. Many applications can generate PDF files directly. The PDF Writer, available on both Apple® Macintosh® computers and computers running the Microsoft® Windows® environment, acts as a printer driver. Some applications produce PostScript page descriptions directly because of limitations in the QuickDraw or GDI imaging models or because they run on DOS or UNIX® computers, where there is no system-level printer driver. For these applications, PostScript page descriptions can be converted into PDF files using the Acrobat Distiller® application. The Distiller application accepts any PostScript page description, whether created by a program or hand-coded by a human. The Distiller application produces more efficient PDF files than PDF Writer for some application programs[1]. Furthermore, there are many simple and affordable third-party software tools to produce fully featured PDF files from any application, for example:5D PDF Creator[12], Jaws PDF Creator[12], et al..

In general, a PDF document's trailer contains a reference to an Info dictionary that provides information about the document. This dictionary contains some keys, for example: Author, CreationDate, Creator, Producer, et al. Fig.1 shows an Info dictionary[1].

```
1 0 obj
<<
/Creator (Adobe Illustrator)
/CreationDate (D:19930204080603-08'00')
/Author (Werner Heisenberg)
/Producer
(Acrobat Network Distiller 1.0 for
Macintosh)
>>
endobj
```

Fig.1 An Info dictionary.

So, we can reveal the presence of secret data for a received PDF file by the following steps:

Step 1 Ransack the received PDF file, to obtain the “Creator” or “Producer” key in Info dictionary;

Step 2 Construct a PDF file by using the software tool according to the obtained “Creator” or “Producer” key;

Step 3 Compare the “valid” objects in the received PDF file with the “valid” objects in the Constructed PDF file, if the sequence of entries in some “valid” objects in the received PDF file is different from the sequence of entries in “valid” objects in the Constructed PDF file, then, the received PDF file may be a stego-file because PDF files generated from the same tool must have the same structure.

Actually, “valid” objects in an original PDF file must have the same sequence of entries. Thus, a received PDF file with disorderly sequences of entries in “valid” objects may be regarded as a stego-file.

B. Estimating the Embedding Rate and the Length of Hidden Data

Let α be the ratio between the number of blocks containing secret bits and the total number of blocks.

According to the embedding algorithm in section II, we easily know that $\alpha \approx \frac{1}{2}$, and get the following iterative formula:

$$N_i = \begin{cases} N_{i+1} \times n_j ! + t_i, if \ 1 \le i \le \sum_{l=1}^{M} n_l \\ 0, if \ i \ge \sum_{l=1}^{M} n_l \end{cases} \quad (1)$$

It is obvious that N_1 is the embedded integer which changed from the binary bit string, and

$$N_1 = A^{p_1} \times t_1 + A^{p_2} \times t_2 + ... + A^{p_B} \times t_B, \quad (2)$$

where $A = \max_{1 \le j \le M} \{ n_j ! \}; B = A - 1$.

On the other hand, because the data to be hidden can be viewed as a random bit stream since they are usually encrypted before embedding, there is no harm in assuming that:

$$p_1 \approx p_2 \approx ... \approx p_B \approx \frac{M}{A}$$

Thus, Equation (2) is equivalent to Equation (3):

$$N_1 = A^{(M/A)} \times (t_1 + t_2 + ... + t_B) \quad (3)$$

In addition, according to the Definition 2 in section II, we have:

$$N_1 = A^{(M/A)} \times (1 + 2 + ... + B) \quad (4)$$

Because M, A and B can be obtained easily by the methods of the above section A, the embedded integer can be estimated. And then, we can change the estimated integer into a binary bit string, and estimate the Length of hidden data.

IV. MODIFICATION TO THE EQUIVALENT TRANSFORMATION BASED STEGANOGRAPHIC METHOD

The same as steganographic methods for structure texts, the equivalent transformation based steganographic method for PDF files can not prevent reconstruct attacks. A PDF stego-file may be revealed the presence of secret data easily. However, by proposing a modified scheme, this paper can avoid the parameters of the embedding rate and the length of hidden data being estimated.

A. Avoiding the Embedding Rate Being Estimated

To produce the random binary sequence $\{ x_n \mid n = 1, 2, ..., x_n \in \{0,1\} \}$,which used to decide whether or not to equivalently transform entries, this paper does not take $x_n = \text{sgn}(x_k) = \begin{cases} 1, x_k \ge 0.5 \\ 0, x_k < 0.5 \end{cases}$,but takes $x_n = \text{sgn}(x_k) = \begin{cases} 1, x_k \ge \tau \\ 0, x_k < \tau \end{cases}$, where x_k is produced by Logistic Chaotic Map[11], and $\tau \in (0,1)$ is a secret threshold value known only by authorized users. Thus, the embedding ratio α is dynamic and adjustable.

Additionally, random sequences of n entries in valid objects must be produced for objects which should not be equivalently transformed in order to avoid the number of embedded units being estimated.

B. Avoiding the Length of Hidden Data Being Estimated

Obviously, the reason why the embedded integer can be estimated easily is that $A = \max_{1 \le j \le M} \{ n_j ! \}$ can be obtained easily.

To avoid the length of hidden data being estimated, this paper presents a novel “random moduli” equivalent transformation based steganographic algorithm. In this algorithm, we use moduli $R_j \times n_j !$ rather than $n_j !$, where R_j (for example: $R_j \in [1,100]$) is a random positive integer produced with x_0 as the initial value. Correspondingly, we must use the equation $\left| O_j(\alpha_{n_j}^k) \right| = \left\lfloor \frac{t_j}{R_j} \right\rfloor$.

In fact, this paper’s steganographic algorithm not only has more security, but also has larger capacity through enlarging the moduli and the embedding ratio α.

V. EXPERIMENTS AND RESULTS

Four PDF files are used as the original documents, respectively, to test the steganalysis performance against the equivalent transformation based steganographic scheme. Table I lists some features of the four PDF files. For case of $\alpha = 0.5$, the actual length, and the estimated length of hidden bits are listed in Table II. Comparative results of capacity for the two steganographic algorithms (the algorithm [2] and this paper’s modification algorithm) are listed in Table III. Table III shows that this paper’s modification algorithm has larger capacity than the algorithm [2].

TABLE I
SOME FEATURES OF THE FOUR PDF FILES

Cover PDF Files	Creator	File Length (Bytes)	The Number of Valid Object
File1	Acrobat Distiller 7.0 \(Windows\)	1352624	186
File2	Acrobat Capture 3.0	581720	48
File3	TTOD CAJ2PDF	270814	27
File4	Adobe Illustrator\(r\) 6.0	5574625	9043

TABLE II
ACTUAL LENGTH AND ESTIMATED LENGTH OF HIDDEN BITS

Cover PDF Files	Actual Length (Bytes)	Estimated Length (Bytes)
File1	39	37
File2	8	9
File3	5	4
File4	1981	1905

TABLE III
CAPACITY OF THE TWO ALGORITHMS

Cover PDF Files	Capacity of the Algorithm [2] (Bytes)	Capacity of the modification Algorithm ($\alpha = 0.9$, $R_j \in [1,10]$) (Bytes)
File1	39	55
File2	8	17
File3	5	12
File4	1981	3016

As we know, estimation results in Table II are based on the assumption: the residue sequence distributes uniformly. But, many PDF files have not enough usable blocks to meet the condition in statistical sense. Thus, although the estimation in Table II is accurate, there are some errors. Additionally, because there is no work which focuses on steganalysis schemes against the equivalent transformation based steganographic methods, this paper can not have some contrast results in Table II.

vI. Conclusion

This paper firstly focuses on the steganalysis scheme against the equivalent transformation based steganographic scheme for PDF files. Through analyzing the structure of a PDF file, this paper's steganalysis scheme can reveal the presence of secret data. In addition, by making use of the fact that the residue sequence distributes uniformly, this paper's steganalysis scheme can also estimate the length of hidden data. Furthermore, to enhance security, it is proposed to introduce a "random moduli" equivalent transformation based steganographic algorithm in this paper. It not only has more security, but also has larger capacity.

The following conclusion can be drawn from the above analysis and computing results: steganographic schemes for PDF files(or for other structure text files) are hard to prevent reconstruction attacks.

One of our future works is to extend this paper's steganalysis scheme to some other data sets, e.g., webpage structure text files[13]..

ACKNOWLEDGMENT

This work was supported by the Fujian Province Young Scientists Foundation under Grant No.2006F3076..

References

[1] Adobe Systems Incorporated. Portable Document Format Reference Manual. Version 1.7. http://www.adobe.com. May,2009

[2] Xingtong Liu, Quan Zhang, Chaojing Tang, et al, "A Steganographic Algorithm for Hiding Data in PDF Files Based on Equivalent Transformation," in Proc. International Symposiums on Information Processing, Moscow, Russia, May. 23–25, 2008, pp. 417–421.

[3] wbStego Studio. The steganography tool wbStego4. http://www.filetransit.com/view.php?id=4177. May, 2009

[4] Shangping Zhong, Tierui chen. Information Steganography Algorithm Based on PDF Documents. Computer Engineering, Vol.32, No.3, Feb. 2006, pp.161–163.

[5] Shangping Zhong, Xueqi Cheng, Tierui Chen., "Data Hiding in a kind of PDF Texts for Secret Communication," International Journal of Network Security, Vol.4, No.1, Jan. 2007,pp.17–26

[6] Chandramouli R,Subbalakshmi K P,"Current trends in steganalysis:a critical survey, "In Proceeding of Eighth International Conference Control on Automation, Robotics and Vision, KunMing: Elseviser Press,2004. pp.964–967.

[7] I. Avcibas, N. Memon, and B. Sankur, "Steganalysis using image quality metrics," IEEE Trans. on Image Processing, vol. 12, no. 2, Feb. 2003, pp. 221–229.

[8] R. Chandramouli, "A mathematical framework for active steganalysis," ACM Multimedia Systems, vol. 9, no.3 , September 2003, pp. 303–311.

[9] X. Zhang and S. Wang, "Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security", Pattern Recognition Letters 25, 2004, pp.331–339.

[10] K C Lu. Combination Mathematics Second Edition, Tsinghua University Press, Beijing, 1991, pp.16–18..

[11] B.L. Hao. Starting with Parabolas-An Introduction to Chaotic Dynamics.IEEE Journal on Selected Areas in Communications, Shanghai, China, 1993, pp.10-12.

[12] Global Graphics Software Ltd. http://www.jawspdf.com/pdf_creator/ May, 2009

[13] Sun Xingming, Huang Huajun, Wang Baowei, et al, "An Algorithm of Webpage Information Hiding Based on Equal Tag", Journal of Computer Research and Development,vol.44,no.5,2007,pp.756–760.