A-025

A Solution of Printing Encryption in PostScript®

Xian Chen †, Hideo Fujii †

1. Introduction

As the technology developing, people pay more and more attention to information security. Now, the printout documents still take a large part in our daily information transferring. In this paper, we present a printing encryption solution in PostScript®(PS), which is one of the most popular Page Description Language (PDL) in the world. One common way used in printing encryption area is by inserting Watermark image after parsing the PDL file. Therefore, there are two aspects of this problem have to be addressed. The first question involves PS file parsing. The second problem relate to implement properly way to insert Watermark image.

2. System Design

When the user wants to print a document by a PS printer, the driver generates a PS file, which contains the document page descriptions. Therefore, by parsing that PS file, getting the page delimiters and inserting Watermark image or security printing commands, we will achieve our encryption goal. Figure 1 shows how the whole system works.

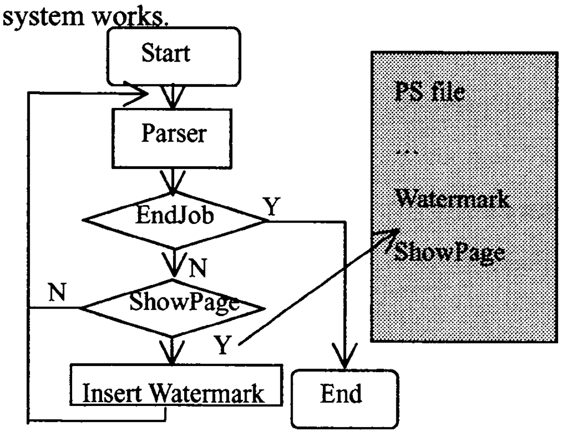

Figure 1: System Flow Chart

The design showing in Figure 1 also can work in the other kinds of PDLs, not only in PS. However, the methods used in parsing the PS file and inserting Watermark image vary through different PDLs.

3. Parser Design

PS can be used as a programming language to describe the original document. The flexibility of PS causes the parsing job very difficult. Fortunately, Adobe® recommends some standard comments in PS file to indicate the file structures. For example, Table 1 lists some typical comments used in standard PS.

Table 1: Some typical PS comments

Comment	Meaning
%!PS-Adobe-3.0	file header
%%BeginPageSetup	page attributes setup
%%PageTrailer	end page
%%Trailer	end PS file

We should also pay attention to that the comments are not essential. It means that the PS file also works if it contains no comments at all. In this special case, we put forward the third party library (i.e. GhostScript) to parse PS file.

4. Insert Watermark Image

After finding the page delimiters, we can insert any Watermark image we like. However, the image mode in PS is opaque. It means that PS doesn't support Raster Operations at all and the Watermark image will override the original document, if we draw the Watermark image secondly. Vice versa. So, we put forward two solutions to merge Watermark image with the original documents.

† Fuji Xerox Co., Ltd.

† {Chen.Xian, Hideo.Fujii}@fujixerox.co.jp

4.1Using "*imagemask*" operator

We can apply "*imagemask*" to achieve our goal. "*imagemask*" operator uses a monochrome sampled image as a stencil mask of 1-bit samples to control where to apply paint to current page in the current color.

According to the rule of using "*imagemask*" above, we should create Watermark image in 1-bit/pixel format. Figure 2 shows a 1-bit/pixel monochrome Watermark image. Figure 3 shows the result of merging Watermark image with the original document.

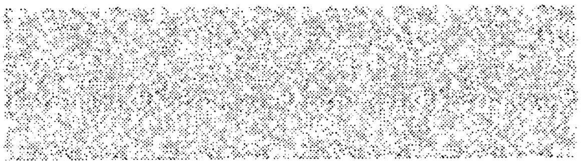

Figure 2: A sample monochrome Watermark image

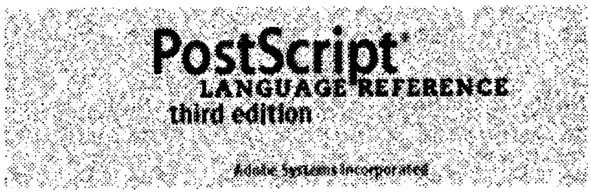

Figure 3: Merged Watermark image

Without "*imagemask*" operator's help, we cannot get a result like Figure 3 shows. The Watermark image will overwrite the characters in the document. In the other hand, if we draw Watermark image first and the original document just consists by not very big characters. The result looks similar with Figure 3. However, if there are images in the document, the Watermark image will be overwritten.

Another problem we have to deal with during our merging Watermark image is how to decrease the file size. A typical A4 image size is 4960*7015 in 600dpi. It is about 4.14MB. We use two steps to decrease the file size. First step is compressing. After applying Run Length and ASCII85 compressing methods, the size is reduced to 224KB. Second step is creating a form, which contains the Watermark image. So, we can print any documents with Watermark image only by adding 224KB(A4 size).

4.2 Converting To BMP

Another way to achieve our goal is converting PS file to BMP image by the third party library (i.e. GhostScript) and merge it with the Watermark image by using Windows Raster Operations. After that convert BMP to PS again. Since "*imagemask*" operator supporting monochrome image only, we can use this method when we want to insert color WM image to PS file.

However, there are two problems during our converting. First, when converting CMYK color from PS file to RGB color, we will miss some color information and vice versa. Second, the size of BMP image converted from PS is about 90MB. It is too big to send to the printer.

5. Conclusion

In this paper, we proposed a solution to merge Watermark image with the original document in PS. It can be used in many areas like secret information hiding and unauthorized copy prohibition. In contrast to the other ways of merging Watermark images in PDL, which used Raster Operation, our work focus on PS, which doesn't support Raster Operation.

References

[1]"PostScript® Language Reference third edition", Addison-Wesley Publishing Company.

[2]"Thinking in PostScript®", Addison-Wesley Publishing Company.