

A Print Control and Manage Method Based on CUPS

Bo Wan, Yifeng Wang, Gang Liu, Quan Wang

The School of Computer Science, Xidian University, Xian Shaanxi

Email: wanbo@xidian.edu.cn, yfwang@xidian.edu.cn, gliu@xidian.edu.cn, qwang@xidian.edu.cn

Received: Oct. 1st, 2015; accepted: Oct. 15th, 2015; published: Oct. 21st, 2015

Copyright © 2015 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

A print control and manage method based on CUPS (Common Unix Printing System) is presented in this paper. A virtual printer driver based on CUPS is designed and installed at first, and then the access control of physical printer and virtual printer is configured through CUPS, which makes the user can access the virtual printer only. When user executes print request from virtual printer, print record is generated and the management info is embedded into each page of print job. At last, the print job embedded management information is sent to the physical printer selected by user to complete the output. The presented method implements print control through configuring the access of physical printer and implements the print management through saving print record and embedding management information into each print page. As the method is based on CUPS, it is suitable for all operating systems which using CUPS such as UNIX, Linux, Domestic operating system and Mac OS X.

Keywords

CUPS, Virtual Printer, Print Control and Manage

一种基于CUPS的打印控制和管理方法

万 波, 王义峰, 刘 刚, 王 泉

西安电子科技大学计算机学院, 陕西 西安

Email: wanbo@xidian.edu.cn, yfwang@xidian.edu.cn, gliu@xidian.edu.cn, qwang@xidian.edu.cn

收稿日期: 2015年10月1日; 录用日期: 2015年10月15日; 发布日期: 2015年10月21日

文章引用: 万波, 王义峰, 刘刚, 王泉. 一种基于 CUPS 的打印控制和管理方法[J]. 计算机科学与应用, 2015, 5(9): 313-321. <http://dx.doi.org/10.12677/csa.2015.59039>

摘 要

本文给出了一种基于CUPS (Common Unix Printing System, 通用Unix打印系统)的打印控制和管理方法。该方法首先设计并安装一个基于CUPS架构的虚拟打印机驱动, 并通过CUPS配置虚拟打印机和物理打印机的访问控制, 使得用户只能访问虚拟打印机。当用户从虚拟打印机发起打印请求时, 在虚拟打印机驱动内产生打印记录并将管理信息嵌入到作业的每个页面。最后将嵌入管理信息后的打印内容发送给用户指定的物理打印机完成输出。该方法通过对物理打印机的访问控制实现打印控制, 并通过存储打印记录和在打印页面上嵌入管理信息实现打印管理。由于该方法建立在CUPS架构上, 因此适用于所有使用CUPS架构的操作系统, 包括Unix系统、Linux操作系统、国产操作系统以及Mac OS X操作系统。

关键词

CUPS, 虚拟打印机, 打印控制和管理

1. 引言

随着数字化技术的迅猛发展, 目前许多行业和个人普遍都使用电子文档处理资料和保存信息, 并且经常通过打印, 复印等途径将电子文档中的内容显示在纸张上, 以供阅读和传播。这一过程无疑对资料和信息保密是一个重要挑战。存在的问题包括: 如何控制用户使用打印机避免纸张浪费和打印机损耗; 如何对进行打印操作的用户信息进行记录, 并嵌入到打印数据中使形成的纸质文档具有明显身份标识, 便于纸质文档管理和来源追踪。

目前已经出现了针对 Windows 操作系统下的打印控制和管理方法的诸多研究成果及成熟的应用系统, 例如黎伶俐等在“打印监控审计技术研究”中公开了 Windows 系统下常见的四种打印监控技术[1]。刘慧[2]在其学位论文中对 Windows 系统打印原理与网络打印机连接方式以及网络打印协议进行了研究与分析并提出了多种方式实现对网络打印的监控管理方案。北大方正集团在“一种文档打印管控与文档溯源追踪方法和系统”专利中给出的打印管控和文档溯源追踪方法等[3]。出于对企业成本的节约以及对企业信息安全的考虑, 不少公司已经在打印机监控方面做出了不少的研究, 并做出了许多好的产品。如上海银兆的 PrintUsage、上海达安科技的 PrinterAdiut 打印监控系统、较流行的打印机监控王、山明打印机监控系统、WebJetAdmin 等。

然而在 UNIX/Linux 操作系统或其衍生系统下, 尚未出现相应的研究成果。随着 Linux 系统的广泛使用, 该系统下的打印控制和管理方法的重要性已经日渐提升。此外, 国产操作系统都是基于 Linux 的衍生系统, 随着国产化的推进和保密要求的日渐严格, 此类操作系统下的打印控制和管理方法也越来越具有重要意义。Unix/Linux 下打印总是有许多限制, 但如果安装了 CUPS (Common UNIX Printing System), 则可以得到一个完整的打印解决方案[4], UNIX 操作系统、Linux 操作系统、国产操作系统和 Mac OS 操作系统均采用 CUPS 作为打印系统架构。因此, 研究基于 CUPS 架构的打印管控方法, 是实现 Linux/UNIX 系统及国产操作系统下打印管控的关键。本文给出一种基于 CUPS 架构的方法, 以解决 Linux 操作系统和国产操作系统上的打印控制及打印管理等问题。

图 1 为本文给出的打印控制和管理方法的整体结构。该方法以 CUPS 架构为基础, 包括虚拟打印机驱动、用户配置、打印记录、多个物理打印机。虚拟打印机驱动和物理打印机安装在 CUPS 架构上, 用户配置和打印记录保存在系统存储器。CUPS 是本发明的底层架构, 用于支持虚拟打印机及物理打印机, 并对打印机访问控制提供支持。虚拟打印机驱动用于获得打印操作用户名, 打印内容及其他管理信息。

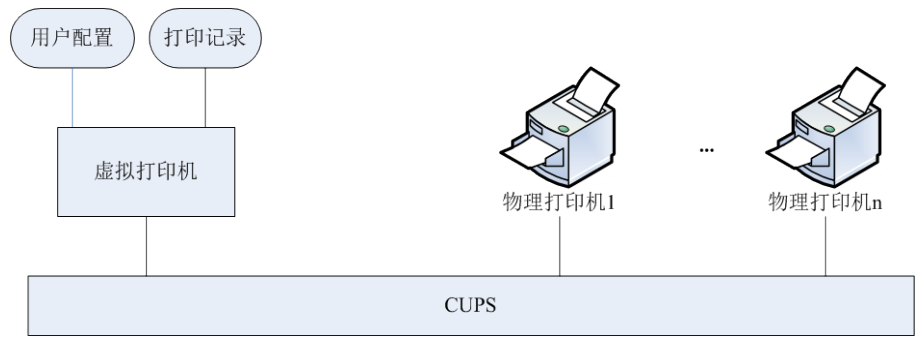


Figure 1. Architecture of the print control and manage system based on CUPS
图 1. 基于 CUPS 的打印控制和管理方法架构

用户配置保存不同用户的配置信息，包括可操作物理打印机列表等，其作用是为显示给用户的操作界面提供数据。打印记录保存打印操作的用户名、拷贝数、标题、页数以及打印内容等，为后续的查询及审计等管理功能提供数据。物理打印机通过通信线缆与计算机连接，接收嵌入用户信息的打印文档，完成实际的打印操作，得到纸质文档。

2. 虚拟打印机驱动设计

CUPS 架构下的打印机驱动包括两种类型：PostScript 打印机驱动和光栅打印机驱动[5]。图 2 和图 3 给出了这两种类型的打印机驱动过滤器链。无论是哪一种类型的打印机驱动，打印作业的数据流是相同的。在用户打印的时候，调度程序 cupsd 确定打印作业的格式以及用于将打印作业转换成打印机能识别的信息的程序(CUPS 已经包含了针对多种通用格式的过滤程序，例如将 PDF 文件转换为 CUPS 光栅数据的过滤程序)。然后将打印作业转换为 CUPS 的两种标准数据格式之一，转换后的数据发送到过滤器程序，由过滤器程序转换为打印机能识别的格式(例如 HP-PCL)。转换后的数据发送到后端程序(backend)，后端程序再将数据通过物理接口发送到打印机。在图 2 和图 3 中不同组件之间的数据传递均采用标准输入(stdin)和标准输出(stdout)，即从标准输入获得前级组件传来的数据，处理后的数据通过标准输出发送给后级组件。对比图 2 和图 3 所示的两种类型打印机驱动的数据流，其区别仅仅在于从 CUPS Filter 得到的数据是 PostScript 格式还是光栅格式。由于 PostScript 语言是一种文本格式的打印控制语言[6]，而且在 Linux 系统下 PostScript 格式数据可以方便的转换到其他格式且便于嵌入自定义信息，因此本文设计的虚拟打印机驱动以 PostScript 格式作为中间数据格式，即采用图 3 的流程设计虚拟打印机驱动。

基于 PostScript 的虚拟打印机驱动，包括一个打印机描述文件(PPD 文件)、一个过滤器程序(filter)和一个后端程序(backend)。打印机描述文件包含虚拟打印机驱动的信息和用户界面配置。过滤器程序是一个命令程序，用于获得 CUPS 传入的打印操作信息和内容。后端程序用于产生一个在 CUPS 架构中对用户可见的虚拟端口，并存储打印内容，在安装虚拟打印机驱动时需要将虚拟打印机安装到该端口。通过打印机描述文件将虚拟打印机安装在 CUPS 下之后，所有通过该虚拟打印机进行的打印操作及相关信息都会发送到该虚拟打印机的过滤器程序，而过滤器处理后的数据将发送到后端程序。

2.1. PPD 文件设计

PPD 文件由文本格式的驱动信息文件(通常以.drv 为后缀)通过 CUPS 的 PPD 编译器编译得到。驱动信息文件中定义了打印机的各种属性和选项，包括产品信息、过滤器、介质和分辨率以及自定义选项。由于虚拟打印机驱动并不直接输出打印内容而是要将打印内容通过用户选择的物理打印机进行输出，因此为了兼容绝大部分物理打印机，虚拟打印机需要提供常用的纸张、分辨率、单双面打印等打印选项，

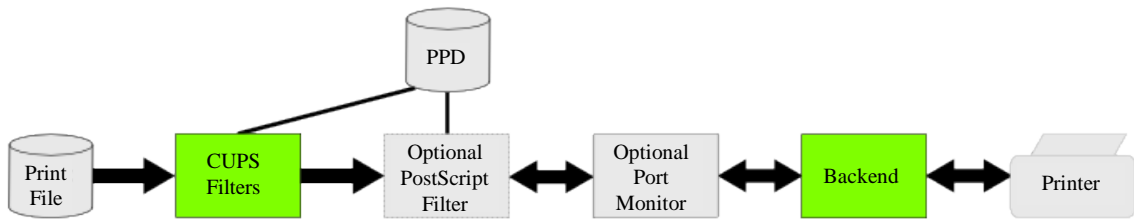


Figure 2. Postscript filter chain
图 2. PostScript 驱动过滤器链

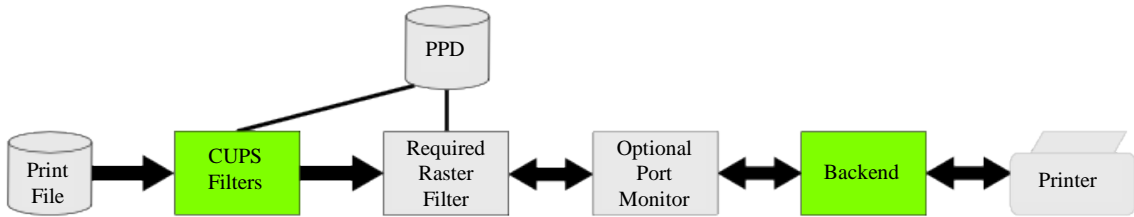


Figure 3. Raster filter chain
图 3. 光栅驱动过滤器链

这些选项只需要在驱动信息文件中参照 CUPS 的要求进行设置即可。在我们的打印机驱动信息文件中最重要的是指明该驱动将要用到的过滤器程序，使得 CUPS 在对打印操作进行调度时，可以把打印内容发送到指定的过滤器程序。而要指定驱动使用的过滤器程序只需要按照下面的格式在文件中增加 **Filter** 命令即可。

```
DriverType ps
Filter application/vnd.cups-postscript 100 vfilter
```

其中“**DriverType ps**”命令表明我们的驱动类型是 PostScript。而“**Filter application/vnd.cups-postscript 100 cpifilter**”表明我们的驱动对应的过滤器程序名为“**vfilter**”，且过滤器内得到的数据为 PostScript 格式。驱动信息文件完成后通过 PPD 编译器 PPDC 进行编译即可得到安装驱动所需的打印机描述文件。

2.2. 过滤器程序设计

过滤器程序为标准命令行程序，由 CUPS 调度。当过滤器程序被调度时，通过命令行参数传入了打印作业的相关信息，如表 1 所示。从表 1 中可知，发起打印作业的用户名可从 argv[2] 中获得。其他打印选项如纸张尺寸，送纸器等可以利用 cups 标准函数 cupsGetOption 从 argv[5] 中获得。

通过命令行参数 argv[2] 得到当前用户名，并使用该用户名在用户配置文件中查找该用户可以使用的所有物理打印机，生成物理打印机列表并通过对话框展示给用户。在用户选择物理打印机后，过滤器程序使用用户名，物理打印机名称，拷贝数量等信息生成打印记录，然后进行打印内容处理。

打印内容处理的主要目的是将包括用户名，作业 ID，作业标题以及打印时间等内容的打印管理信息嵌入打印作业的每个页面。在过滤器中获得的打印内容是 PostScript 格式的打印数据。PostScript 格式的打印数据会使用“showpage”命令来打印每一页，因此只需在打印内容数据流中找到所有“showpage”命令，并在每一条“showpage”命令之前增加一条“show”命令将打印管理信息文本添加到页面中不影响页面内容的地方(通常在页面顶部或底部)。在完成页面内容处理后，嵌入了管理信息的打印内容通过标准输出发送给后端程序。过滤器程序的处理流程如图 4 所示。

Table 1. Command line arguments of filter
表 1. 过滤器程序传入的命令行参数

命令行参数	参数内容
argv[1]	作业 ID
argv[2]	打印用户
argv[3]	作业名/标题
argv[4]	拷贝数
argv[5]	打印选项

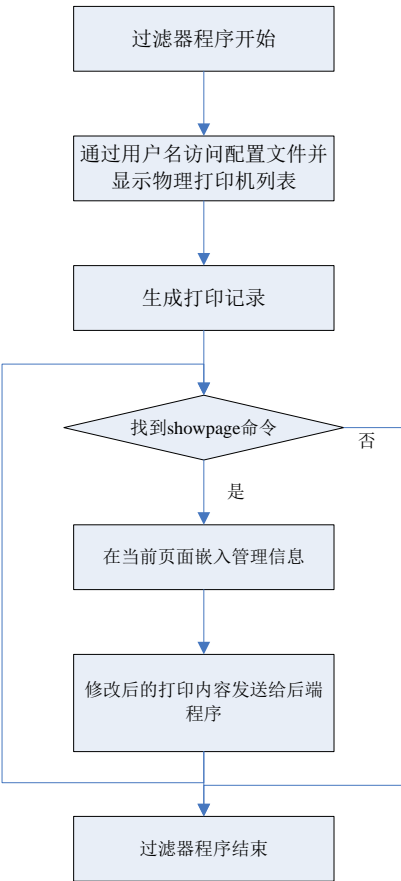


Figure 4. Filter flow chart
图 4. 过滤器程序处理流程

2.3. 后端程序设计

后端程序(backend)将过滤器程序处理后的 PCL 数据流发送给打印机，完成打印过程。设计后端程序的关键是要让 CUPS 识别所扩充的后端程序是一个有效的后端程序，这一过程在后端程序中称为“设备发现”。其核心是当后端程序以不带命令行参数形式被 CUPS 调用时将该后端程序支持的设备信息输出到标准输出(stdout)，输出的内容包括“设别类别”(例如，direct、network、file 等)，“设备 URI”，“制造商信息”和“设备信息”。为了使虚拟打印机可以安装到该端口上，虚拟端口类别为 direct，其设备发现信息如下所示。

```
direct vend:/dev/null "" "VirtualPort"
```

此外, 后端程序还需要从标准输入读取所有打印数据并存储到文件, 同时发送给物理打印机, 直到接收完所有打印数据, 后端程序流程如图 5 所示。

3. 打印控制和管理方法

在完成虚拟打印机驱动设计后, 可以通过 CUPS 和虚拟打印机驱动协作完成打印控制和管理。打印控制和管理方法的具体步骤如下:

- 1) 安装基于 CUPS 架构的虚拟打印机驱动。
- 2) 通过 CUPS 对虚拟打印机和物理打印机(包括本地打印机、共享打印机和网络打印机)进行访问控制, 使得所有物理打印机只能被一个区别于普通用户的特定用户(该特定用户的用户名可以由 root 用户指定, 假设为 `print_user`)访问, 而虚拟打印机可以被所有用户访问。
- 3) 配置各个用户可以用来输出打印内容的物理打印机, 并形成配置文件。
- 4) 用户在客户端启动打印操作, 此时只能访问虚拟打印机。当用户在打印操作界面中选择虚拟打印机启动打印操作时, 虚拟打印机获得打印请求。
- 5) 虚拟打印机驱动获得打印操作的用户名, 并根据用户名从步骤 3) 的配置文件中获得该用户可以使用的物理打印机列表, 并将列表展示给用户以供选择。
- 6) 用户在物理打印机列表中选择用于完成实际打印操作的物理打印机名称, 虚拟打印机驱动记录用户名, 用户选择的物理打印机名称、打印格式及打印作业的相关信息(包括标题, 拷贝数量, 打印时间等)。
- 7) 虚拟打印机驱动在 PostScript 格式的打印页面中嵌入步骤 6) 的打印管理信息。
- 8) 虚拟打印机驱动通过特定用户(`print_user`)将嵌入管理信息的打印内容按照用户指定的打印格式发送给用户选择的物理打印机, 完成实际输出。

4. 实验结果

为了验证本文给出的打印管理方法, 我们设计并实现了一个简单的原型系统。该原型系统在 Linux Fedora 16 操作系统下进行安装并测试。测试内容主要包括: 虚拟打印机安装, 物理打印机访问控制、打印记录生成及管理信息嵌入。

4.1. 虚拟打印机安装测试

使用 2.1 节给出的 PPD 文件, 过滤器程序和后端程序, 可以将虚拟打印机驱动安装到 CUPS。安装完成后该虚拟打印机可以在 CUPS 中查看, 如图 6 所示。

4.2. 物理打印机访问控制

通过 CUPS 的管理页面可以设置每个物理打印机的访问控制, 在原型系统中我们将所有物理打印机设置为不能被普通用户访问, 而虚拟打印机可以被所有用户访问。如图 7(a)所示, 在设置访问控制之前, 用户在进行打印操作时, 可以看到所有物理打印机和虚拟打印机。如图 7(b)所示, 设置了访问控制后, 用户在进行打印操作时, 只能看到虚拟打印机。

4.3. 打印记录生成及管理信息嵌入

用户进行打印操作时, 首先根据用户名访问用户配置文件, 弹出图 8 所示的选择物理打印机对话框, 在用户选择了某个物理打印机后, 用户名, 物理打印机名、作业 ID、作业标题、打印时间等信息被嵌入到打印作业的每个页面, 如图 9 所示。

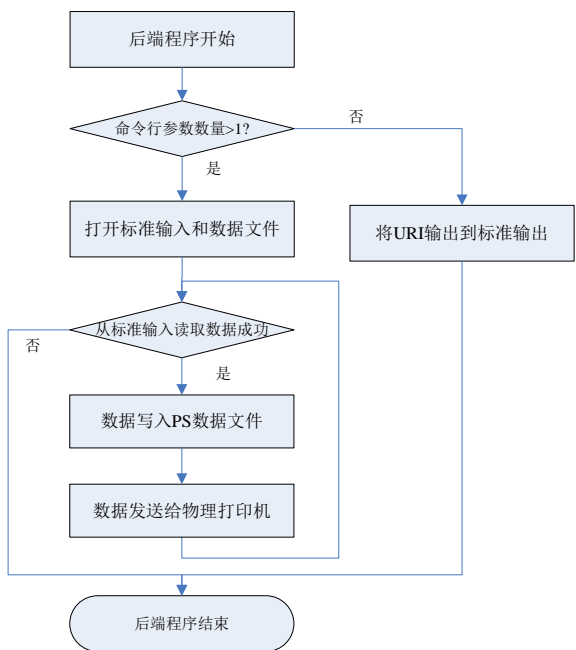


Figure 5. Backend flow chart

图 5. 后端程序流程

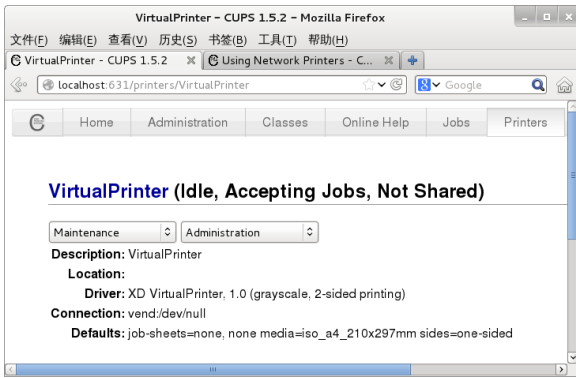
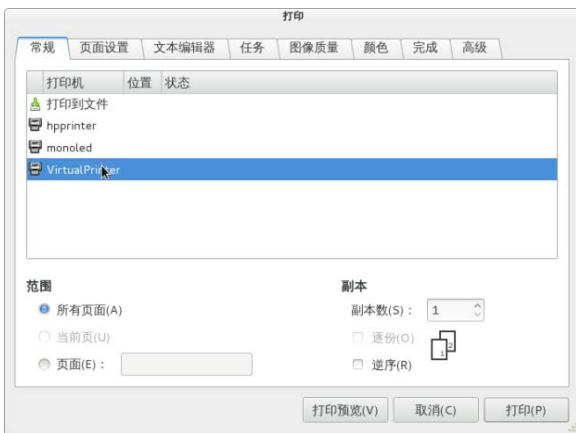


Figure 6. The virtual printer installed in CUPS

图 6. CUPS 中安装的虚拟打印机



(a)



(b)

Figure 7. Printer access control: (a) before access control; (b) after access control

图 7. 打印机访问控制 (a) 设置访问控制之前; (b)设置访问控制之后

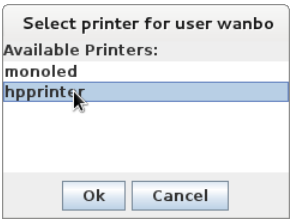


Figure 8. The dialog used to select physical printer

图 8. 选择物理打印机对话框

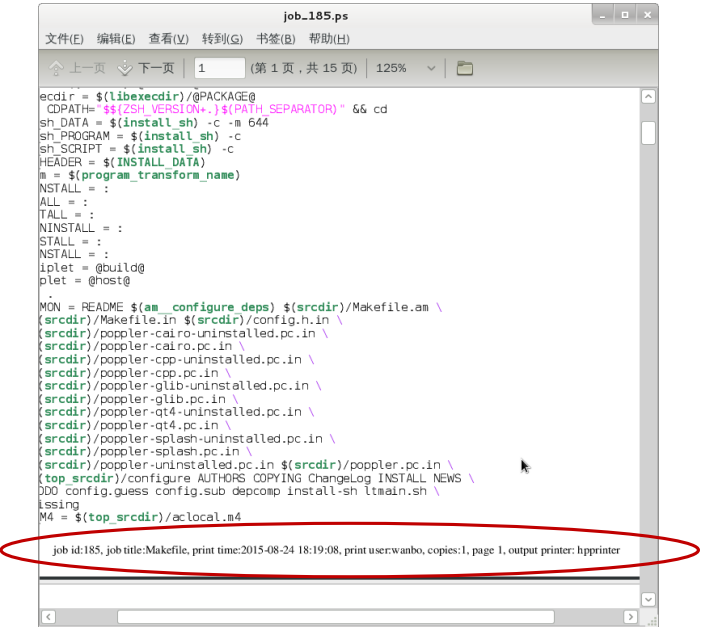


Figure 9. The page embedded manage information

图 9. 嵌入了管理信息的打印页面

5. 结论

小节 4 的测试证明了本文所给出的打印控制和管理方法及原型系统的可行性。由于目前国内还没有针对 Linux 系统及国产操作系统的打印控制和管理方法的研究成果,因此本文给出的方法及原型系统具有独创性。本文给出的方法以 CUPS 架构为基础,而 CUPS 架构在 Linux、UNIX,国产操作系统以及 Mac 系统下都被广泛使用,因此该方法也适用于这些系统,具备较强的通用性。

文中给出的原型系统还存在若干可以进一步完善的地方:1)除了明文形式的来源信息外,还可以使用打印管理信息生成二维码图像,然后通过数字水印技术在页面中嵌入二维码图像,嵌入水印之后的页面在打印后与原始文档的区别无法用肉眼分辨,但通过扫描和特定水印提取算法可以得到嵌入在页面中的二维码图像,从而得到打印管理信息,这种方式更加安全和隐蔽;2)除了嵌入打印来源信息外,还可以通过[7]给出的方法获得打印作业中包含的文本信息,然后通过文本匹配算法找到所有敏感的关键字。获得的关键字可以用于对打印作业进行审核,进一步增强打印管理系统的安全性。

基金项目

本文由西安市科技计划项目《智能管理与数字生活关键技术与研发——打印安全管理系统》资助,项目编号: CXY1440(2)。

参考文献 (References)

- [1] 黎伶俐, 郝东白, 张诚, 黄皓 (2009) 打印监控审计技术研究. *计算机工程与设计*, **2**, 484-486.
- [2] 刘慧 (2008) 内网监管系统中打印机监控模块的设计与实现. 硕士论文, 电子科技大学, 成都.
- [3] 北京大学, 北大方正 (2012) 一种文档打印管控和文档溯源追踪方法和系统. 中国专利: CN102541482A.
- [4] Sweet, M. (1999) A Bright New Future for Printing on Linux. *Linux Today*.
- [5] CUPS 2.1. Apple Inc. <http://www.cups.org>
- [6] PostScript Language Reference. 3rd Edition, Adobe Systems Incorporated, San Jose.
- [7] Nevill-Manning, C.G., Reed, T. and Witten, I.H. (1998) Extracting Text from PostScript. *Software: Practice and Experience*, **28**, 481-491. [http://dx.doi.org/10.1002/\(sici\)1097-024x\(19980425\)28:5<481::aid-spe163>3.0.co;2-p](http://dx.doi.org/10.1002/(sici)1097-024x(19980425)28:5<481::aid-spe163>3.0.co;2-p)