Подход за защита на PDF документи срещу копиране

Георги Върбанов, Петър Антонов

Approach to protect PDF documents against copying: *The aim of this study was to add additional features to protect a certain class allows electronic documents to be protected from copying and scanning, provided that such a requirement is necessary. Like WEB-based solution for would help protect copyright of the creators of this document and added a variety of tools built into many PDF. This idea can successfully find a place where copyright protection for remote e-learning.*

Key words: *protect copyright PDF, Model, remote e-learning.*

ВЪВЕДЕНИЕ

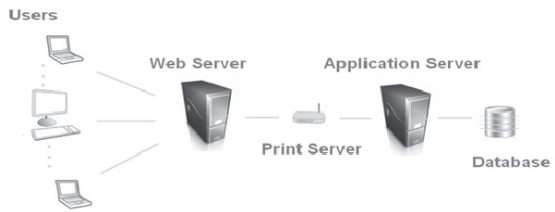
Основната идея на разработката е да се намери начин защитаващ правата на създателите на електрони документи от принтиране и OCR сканиране, когато се изисква такава възможност. Известни са много методи вградени в частност и в Pdf документи-като защита от копиране и отпечатване, добавяне на пароли за достъп.

Има разработени много методи, които маркират съответния текстов документ чрез водни знаци- използващи различни техники – изместване на отделни букви на сравнително малки разтояния преместване на редове с пиксел и много такива позволяващи да се намери вмешателство в дадения документ. Известни са и техниките даващи най-добрата защита чрез електронен подпис[2][3][4][5]. Всички те обаче не защитават автора от лесно копиране поради наличието на много програми разбиващи паролите и последващо премахване на защитите и копиране на текста чрез OCR програми. В настоящата статия се прави опит свежадащ до минимум възможноста да бъдат сканирани и принтирани документите чрез виртуални принтери или такива направени при разбиване на паролите и опит за печат и последващо сакниране с OCR програми.

Приложение на такъв тип система може да бъде например при дистанционното обучение. При него в ролята на потребители са учениците(студентите), а учебниците и публикациите, които те искат да изтеглят са частна собственост на авторите със съответно авторско право. Желанието ни е тези документи да бъдат предоставени за ползване, но да не могат да се копират и разпространяват. И ако все пак някой реши да се възползва от интелектуалния труд - то злоупотребата да може да бъде установена.

ИЗЛОЖЕНИЕ

Разработката на подобна Web базирана система защитаваща правата потребителите съгласно изложеното по-горе е представена на фиг. 1, в която са включени:



Фиг. 1

Потребители отправящи заявки за изтегляне на даден файл
Web Server автентикира съответния потребител и проверява правата му.

Print Server – играе ролята на виртуален скенер. Той приема заявката за документа и я предава на Application Server-а за изпълнение и след това праща резултата отново на Web Server-а.

Application Server – където се извършват основните изчисления и конвертиране на документите и добавяне на защитата.

Database – база от данни на която се държат документите, възможно е при Web сървъра да има допълнителна база с регистрираните потребители и правата им

Това е едно възможните приложения на подобна система, но тя може да бъде внедрена и на много други места.

Недостатъци на съществуващите стандартни решения

- най-простата атака върху документ, който няма никаква защита е просто да се копира
- ако сме добавили парола може да се използват средства за нейното разбиване.

При цифровия подпис нещата не са толкова лесни, но ние целим предпазване на съдържанието на документа, а не на самия него, така че ако някой може да се добере до съдържанието и да създаде нов документ с него подписа е неефективен.

Ако е копирана само част от документа или са сменени няколко думи с техни синоними хеш функциите няма да върнат съвпадение но смисъла на документа остава. Остава и идеята на автора.

Изисквания предявени към системата:

- Да се намери начин да се предпази текстов документ от копиране и OCR сканиране.
- Да може да се установи дали е подправен.
- Да може да се разпознае на целия или част от него при нужда.
- Да може да се чете свободно.
- Автоматизация на горепосочените.

Анализ на поставените цели:

Целите ни ще бъдат изпълнени само, ако за нарушителите бъде оставен само варианта да пренапишат всичко наново.

Възможни решения на всеки от проблемите

За предпазването на текстовия документ от копиране:

- може да му се сложи парола за да нямаме достъп до съдържанието
- забранява се четенето -понеже докато го се разглежда документа може да бъде копиран.

За да се установи дали е променян:

- цифрово подписване

За разпознаване:

- сравнение с помощта на хеш фукнкции или програми даващи разлики
- скриване на информация

За свободно четене:-**никаква защита!**

Както се вижда стандартните подходи за справяне с всеки от проблемите са в голяма степен взаимно изключващи се и трудно приложими за стандартен текстов документ, който има много ниско ниво на сигурност.

Недостатъци на съществуващите стандартни решения

- най-простата атака върху документ, който няма никаква защита е просто да се копира.
- ако сме добавили парола може да се използват средства за нейното разбиване.
- при цифровия подпис нещата не са толкова лесни, но ние целим предпазване на съдържанието на документа, а не на самия него -така че ако някой може да се добере до съдържанието и да създаде нов

документ с него подписа е неефективен.

- Ако е копирана само част от документа или са сменени няколко думи с техни синоними хеш функциите няма да върнат съвпадение.

Предложено решение

Предложеното решение е сравнително просто, комбинира няколко подхода и има много възможности за развитие.

Използвания формат на разпространение е PDF - защото той е свободен има много на брой безплатни програми, с които може да се разглежда. Притежава големи възможности и е много по-сигурен. Портируем е т.е. независимо от машината и операционната система ще може да бъде четен. Основен стандарт е при електронните документи в Интернет.

Форматът предоставя много възможности за ограничаване като:

- Забрана на принтирането
- Забрана на достъпа документа
- Забрана на копиране на съдържанието
- Забрана на вземането на страница
- Забрана на попълване на полетата
- Подписване
- Забрана на изработка на шаблонна страница
- Парола

В генерирания PDF файл се:

- Добавя парола, която трябва да бъде достатъчно дълга за не може да бъде разбита с нищо друго освен с подхода на грубата сила т.е. да се пробват всички възможни комбинации.
- Добавя се скрит слой на който имаме маска. Този слой е видим единствено при принтиране на документа и маската е така разположена, че да прави текста нечетаем.
- Маската е създадена с алгоритъм, чрез който да може да се определи информация за самия текст и по-късно тя да се използва за сравнение. Всяка област се разделя чрез делител, които зависи от степента на това дали е четен или не във функция от разпознатия текст. Всяка област се разделя и кодира чрез подходяща хеш функция и се разделят четните и нечетните области с видими- прозрачни и такива с бял или черен цвят. Предварително се разпознава височината на текста и между текстовото разтояние така, че получената маска се поставя на отделен слой, които се показва в случай на опит за принтиране или сканиране. Например ако се използва алгоритъм за кодиране на разстоянията между думите във всеки параграф дори при замяна на някоя дума с друга или директно копиране ще може да се изчисли степен на разпознаване и да се каже колко процента съвпадат, коя част е подменена и др.

Възможни допълнения за повишаване на сигурността:

- Възможност за добавяне на баркод, който може да е видим или отново върху отделен скрит слой [4].
- Възможност за цифрово подписване[4][5].
- Възможност за воден знак[5][4].

Атаки

При така създадения документ са възможни различни видове атаки.

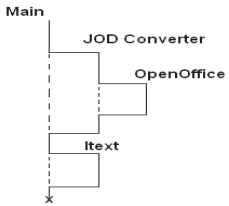
- за разбиване на паролата на файла и вземане на съдържанието
- за разпознаване на документа с OCR програма
- за принтиране и след това сканиране на документа и разпознаване
- опити за премахване на слоевете

Алгоритъм

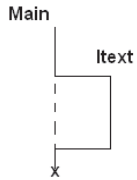
Използвани средства: за реализация на проекта са използвани език Java 1,6 JodConverter,OpenOffice[13],Itext,BauntyCastel[17],За тестване и разглеждане на получените резултати ABBYY FineReader 11, PDF Creator, Image Printer 2.0.1, Adobe Acrobat Reader 9 и Foxit Reader 4.1, password pdf remuver 3.1 Font Creator 5.6

Разработени класове: JPDFFileChooseDialog JPDFOptionsDialog JPDFEncryptDialog JPDFDecryptDialog JPDFGuardFileFilter JPDFAddWatermarkDialog, JPDFEncryptOptions, JOD Converter- се използва за да конвертиране от Microsoft Word → PDF

iText за генериране и добавяне на маската.



Фиг. 2



Фиг. 3

Фиг. 2.:JOD Converter – етапи за конвертирането в → PDF

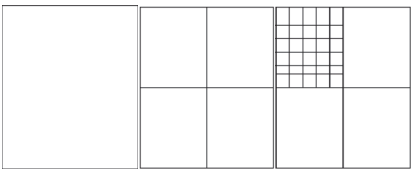
- Създава нов процес и стартира OpenOffice като услуга
- Отваря нова конекция към нея
- Конвертира файла
- Затваря конекцията
- Спира процеса

Алгоритъм за декодиране на вече кодиран PDF документ

Фиг. 3.:Main – главна програма, която декодира PDF документа

- iText – метод, който взема маската и я декодира.
- отваря документа и за всяка страница търси картинки, ако те се намират на скрития слой -значи това е маската.
- декодира маската.
- връща декодираната хеш стойност.

Етапи на изграждане на маската показан на фиг 4:



Фиг. 4

- Изображението се разделя на четири равни части
- Едната четвърт се разделя на толкова на брой равни клетки, колкото са
- числата, които се избират за запис в случая 32-
- Генерира се ред от клетката според избраното число за запис.
- Числата са от 0 – до 15, случаен генератор.
- Начина на кодиране е да се запишат толкова на брой битове в нечетни клетки колкото е числото и процента на запълване на реда

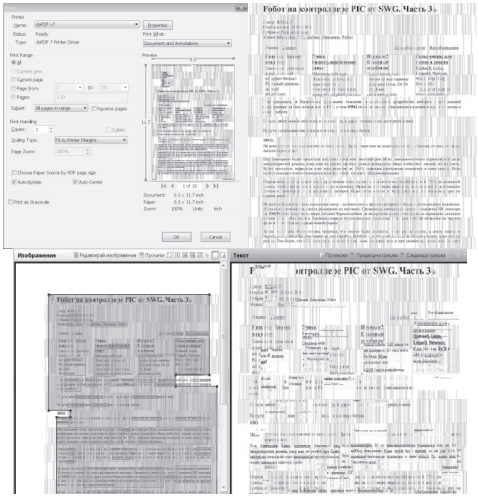
[Число]*4% < Процент на запълване < [Число + 1]*4%

Пример:

За запис на 4 → трябва да има четири бита в нечетни клетки
и процента на запълване да е между 16% и 20%.

РЕЗУЛТАТИ

Получените резултати дават възможност да се потвърди в ефикасноста на предложения метод. След тестване с виртуален принтер и последващо сканиране с OCR степента на разпознаваемост на документите не превишава повече от 20 % независимо от обекта, който е принтиран и сканиран. Опита с премахване на защитите на Pdf документа и последващо принтиране не дават желания резултат от атаката и резултата от разпознаването остава в същите минимални граници.



Фиг. 5

ЗАКЛЮЧЕНИЕ

Практическото използване на създадения програмен модел дава основание да се работа в тази посока и добавяне на допълнителни функции за разпознаване на подправен текст и прилагането му в защита на електронни документи дистанционно обучение. Би могло да се помисли за в бъдеще за вграждане на този слой с психо визуланите особености на човешкото зрение при работа с картини съдържащи текст.

ЛИТЕРАТУРА

[1] iText In Action - Creating And Manipulating PDF -Bruno Lowagie 2007

[2] Key Based Text Watermarking of E-Text Documents in an Object Based Environment Using Z-Axis for Watermark Embedding -World Academy of Science, Engineering and Technology 46 2008

[3] Efficient Text Color Modulation for Printed Side Communications and Data Hiding -Paulo Vinicius Koerich Borges, Ebroul Izquierdo Queen Mary, University of London Multimedia and Vision Research Group London, UK

[4] Tamper-proofng of Electronic and Printed Text Documents via Robust Hashing and Data-Hiding -

[5] R. Villan, S. Voloshynovskiy, O. Koval, F. Deguillaume, and T. Pun Computer Vision and Multimedia Laboratory - University of Geneva

[6] http://www.pdfforge.org/pdfcreator,

[7] http://code-industry.net/imageprinter.php
[8] http://www.adobe.com/,
[9] http://www.foxitsoftware.com/pdf/reader/,
[10] http://www.abbyy.com/,
[11] http://www.high-logic.com/fontcreator.html,
[12] http://bg.openoffice.org/,
[13] http://www.artofsolving.com/opensource/jodconverter,
[14] http://code.google.com/p/jodconverter/,
[15] http://itextpdf.com/,
[16] http://www.bouncycastle.org/

За контакти:
гл. ас. Георги Върбанов, Катедра "Компютърни системи и технологии", Технически университет-Варна тел.: 052 383-614, e-mail: gvarbanov@gmail.com
доц. д-р. Петър Цветанов, Катедра "Компютърни системи и технологии" тел.052 383 439

Докладът е рецензиран.