DOI: 10.15514/ISPRAS–2020–32(5)–7

Реализация маркирования в подсистеме печати ОС семейства Windows на основе виртуального XPS-принтера

[1] С.В. Козлов, ORCID: 0000-0003-1269-1681 <kozlov_sv@mail.ru>
[1] С.А. Копылов, ORCID: 0000-0003-2841-5243 <gremlin.kop@mail.ru>
[2] Б.В. Кондратьев, ORCID: 0000-0001-6348-117X <gae@mil.ru>
[3] Д.О. Обыденков, ORCID: 0000-0002-9296-6333 <obydenkov@ispras.ru>
[1] Академия Федеральной службы охраны Российской Федерации,
302015, Россия, г. Орёл, ул. Приборостроительная, д. 35
[2] Министерство обороны Российской Федерации,
119160, г. Москва, ул. Знаменка, д.19
[3] Институт системного программирования им. В.П. Иванникова РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25

Аннотация. В статье представлен подход к маркированию электронных документов, выводимых на печать посредством реализации виртуального XPS-принтера в операционных системах семейства Windows. Разработанный подход позволяет осуществлять маркирование электронных документов в процессе печати вне зависимости от формата представления исходного документа и требований, предъявляемых к процессу печати. В ходе разработки и реализации подхода к маркированию осуществлен сравнительный анализ технических решений в области маркирования электронных документов, определены их достоинства и недостатки. Определены требования и ограничения, накладываемые на подход к маркированию. Обоснован выбор технологии виртуальных принтеров для реализации маркирования документов в процессе вывода их на печать. В ходе реализации подхода маркирования, основанного на технологии виртуальных принтеров, приведена структура организации и взаимодействия процесса маркирования с компонентами службы печати операционных систем семейства Windows. Разработана архитектура драйвера виртуального принтера, реализующего маркирование документов. Описан процесс практической реализации внедрения маркера в электронный документ посредством разработанного виртуального принтера. В ходе практической реализации подхода маркирования представлено описание особенностей взаимодействия разработанного фильтра печати с подсистемой печати, параметров обработки метаданных и особенностей организации многопоточной реализации сервера, выполняющего маркирование. Рассмотрены особенности реализации разработанного подхода к маркированию в отдельных операционных системах семейства Windows. Определены ограничения и допущения для каждой из рассмотренных операционных систем. Сформулированы требования к процессу маркирования и направления дальнейших исследований.

Ключевые слова: защита от утечки информации; маркирование документов; виртуальный принтер печати, подсистема печати; операционные системы семейства Windows

Для цитирования: Козлов С.В., Копылов С.А., Кондратьев Б.В., Обыденков Д.О. Реализация маркирования в подсистеме печати ОС семейства Windows на основе виртуального XPS-принтера. Труды ИСП РАН, том 32, вып. 5, 2020 г., стр. 95-110. DOI: 10.15514/ISPRAS–2020–32(5)–7

Implementing Watermarking Based on a Virtual XPS Printer for Windows Operating Systems

[1] S.V. Kozlov, ORCID: 0000-0003-1269-1681 <kozlov_sv@mail.ru>
[1] S.A. Kopylov, ORCID: 0000-0003-2841-5243 <gremlin.kop@mail.ru>
[2] B.V. Kondrat'ev, ORCID: 0000-0001-6348-117X <gae@mil.ru>
[3] D.O. Obydenkov, ORCID: 0000-0002-9296-6333 <obydenkov@ispras.ru>
[1] The Academy of Federal Security Guard Service of the Russian Federation,
35, Priborostroitelnaya st., Oryol, 302015, Russia
[2] Ministry of Defence of the Russian Federation,
19, Znamenka st., Moscow, 119160, Russia
[3] Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia

Abstract. The article presents an approach to electronic documents printed marking by implementing a virtual XPS printer in Windows operating systems. The developed approach allows marking electronic documents during printing, regardless of the document presentation format and requirements for the printing process. During the marking approach development and implementation, a comparative analysis of technical solutions in the field of marking electronic documents was carried out, advantages and disadvantages were determined. Requirements and limitations imposed on the marking approach are defined. The virtual printer technology choice for the marking documents implementation in the printing process is substantiated. In the course of the marking approach implementing based on virtual printer technology, the structure of the organization and interaction of the marking process with the components of the print service of Windows family operating systems is given. The architecture of a virtual XPS printer driver has been developed. The process of practical implementation of the marker embedding into an electronic document using the developed virtual printer is described. In the process of the marking approach practical implementation, a interaction features description of the developed print filter with the printing subsystem, the parameters of metadata processing and the organization features of the marking server multithreaded implementation is presented. The implementation features of the developed marking approach in individual operating systems of the Windows family are considered. Limitations and assumptions are determined for each of the considered operating systems. Marking process requirements and further research directions are formulated.

Keywords: information leakage protection; document marking; virtual printer; printing subsystem; operating systems of the Windows family

For citation: Kozlov S.V., Kopylov S.A., Kondrat'ev B.V., Obydenkov D.O. Implementing watermarking based on a virtual XPS printer for Windows operating systems. Trudy ISP RAN/Proc. ISP RAS, vol. 32, issue 5, 2020, pp. 95-110 (in Russian). DOI: 10.15514/ISPRAS–2020–32(5)–7

1. Введение

С развитием информационно-телекоммуникационных сетей в последние десятилетия в значительной степени увеличилось количество цифровой информации, содержащей как персональные данные пользователей, так и конфиденциальную информацию. Помимо роста числа и разнородности цифровых данных отмечается возросшее количество инцидентов нарушений информационной безопасности, связанных с утечкой защищаемой информации. Так, по данным аналитического центра InfoWatch в 2019 году в мире обнародовано и зарегистрировано 1276 случаев утечки конфиденциальной информации, а также информации, содержащей различные виды тайн, что на 23% превышает количество инцидентов, зарегистрированных за аналогичный период 2018 года (1039 утечек). В результате данных утечек скомпрометировано более 8,74 миллиарда записей персональных данных. При этом, более чем в 55 процентах случаев нарушение информационной безопасности осуществлялось внутренними нарушителями [1]

В распределении утечек по типам информации, подвергшейся компрометации, на персональные данные приходится 71,2% от общего числа, на информацию, содержащую

коммерческую тайну – 22% и сведения, составляющие государственную тайну – 6,8%. Одним из наиболее распространенных типов данных, подвергшихся компрометации, является текстовая информация [2]. При этом почти половина (47,7%) от общего числа утечек реализована через напечатанные на бумаге электронные текстовые документы.

Высокий процент утечек напечатанных текстовых документов обусловлен тем, что существенная часть документооборота совершается в неэлектронной форме. Кроме того, во многих компаниях и государственных организациях далеко не всегда соблюдаются правила обращения с бумажными носителями. При этом утечка напечатанных на бумаге текстовых документов реализуется посредством сканирования или фотографирования бумажного текстового документа с последующим выносом на машинном носителе информации или отправкой за пределы контролируемого периметра сети сформированного изображения, содержащего исходный текстовый документ.

Для защиты печатных копий электронных текстовых документов от утечки широкое распространение получили методы маркирования документов, основанные на технологии цифровых отпечатков и цифровых водяных знаков (ЦВЗ), позволяющие осуществить обнаружение факта утечки информации, а также идентифицировать ее источник [3-6].

2. Обзор существующих решений в области маркирования документов

Технология цифровых отпечатков основана на извлечении из анализируемого объекта (текстового документа, изображения, данных мультимедиа и т. д.) уникальных свойств («отпечатков»), характеризующих исходный документ [7]. Извлеченные характеристики составляют базу данных «отпечатков» (сигнатур), по которой осуществляется контентный анализ передаваемой информации [8, 9].

К практическим решениям в области маркирования, основанным на технологии цифровых отпечатков, относится программный продукт Trace Doc [10]. В процессе маркирования на основе методов машинного обучения и теории распознавания образов для электронного текстового документа, выводимого на печать, создается уникальная копия. Формула создания этой копии, а также метка времени и информация, идентифицирующая пользователя, сохраняются в базу данных. В случае утечки выполняется поиск формулы, позволяющей получить из исходного документа копию, попавшую в публичный доступ (по ошибке или намеренно). Достоинством рассмотренного решения является возможность обнаружения факта утечки и идентификации источника утечки по фотографии или изображению, полученному посредством сканирования напечатанного текстового документа. К недостаткам Trace Doc относится необходимость создания базы данных исходных электронных текстовых документов.

В отличие от технологии цифровых отпечатков в подходе к маркированию, основанном на технологии ЦВЗ, в исходный текстовый документ при выводе на печать внедряется дополнительная информация (маркер) [11].

В качестве маркера может выступать идентификационная информация, характеризующая владельца данных, сами данные, а также другая метаинформация. К существующим решениям в области маркирования электронных текстовых документов, выводимых на печать, относятся такие решения, как EveryTag [12], SafeCopy [13].

Программное средство EveryTag позволяет осуществить внедрение ЦВЗ посредством сдвига строк и слов исходного электронного текстового документа. При этом ЦВЗ содержит уникальную информацию, идентифицирующую пользователя, осуществляющего создание и обработку текстового документа. К недостаткам разработанного программного продукта относится необходимость создания и хранения защищенной базы данных подписанных электронных копий. В случае отсутствия в базе подписанной копии установить факт утечки и идентифицировать нарушителя не представляется возможным.

SafeCopy так же как и EveryTag позволяет маркировать электронные текстовые документы посредством изменения форматирования текста электронного текстового документа. При этом для внедрения ЦВЗ, содержащего идентификатор пользователя и метку времени, необходимо, чтобы страница документа содержала не менее 9 строк текста (40% страницы должно быть текстом). Как и в предыдущем случае требуется создание защищенной базы данных, содержащей подписанные копии электронных текстовых документов.

Требования по наличию подписанной копии электронного текстового документа в процессе идентификации источника утечки конфиденциальных электронных текстовых документов не позволяет использовать рассмотренные программные решения для защиты от утечки, обусловленной преобразованием «печать-сканирование» или «печать-фотографирование», и накладывает дополнительные ограничения на процесс документооборота.

Для обеспечения защищенности электронных текстовых документов необходимо разрабатывать программные решения, способные осуществлять внедрение ЦВЗ в электронный текстовый документ в процессе вывода на печать, который может быть извлечен только из подписанного напечатанного текстового документа или соответствующего изображения без необходимости наличия исходного документа.

При реализации таких алгоритмов возникает необходимость внедрения подсистемы маркирования в операционные системы. В данной статье предлагается подход к реализации маркирования на основе виртуального XPS-принтера для операционных систем семейства Windows.

3. Реализация маркирования в подсистеме печати операционных систем семейства Windows на основе виртуального XPS-принтера

Для маркирования документов в процессе вывода на печать применяются следующие подходы:

- встраивание маркера непосредственно в принтере;
- встраивание маркера в драйвере принтера;
- встраивание маркера в очереди печати (print spooler);
- реализация виртуального принтера с функцией маркирования.

Встраивание маркера непосредственно в принтере реализуется, как правило, аппаратно или в микропрограмме принтера. Эта возможность закладывается производителями принтеров. При этом у пользователя отсутствует или крайне ограничен выбор параметров маркирования.

Встраивание маркера в драйвере принтера во многом аналогично первому подходу, но функция маркирования выполняется на компьютере программным обеспечением драйвера принтера. Такой способ предоставляет пользователю больше возможностей для установки параметров маркирования, однако, изменить алгоритмы маркирования чаще всего невозможно, так как они реализованы производителем в исполняемых файлах драйвера.

Если драйвер принтера разработан в соответствии с архитектурой Microsoft Universal printer driver (Unidrv или V3 Printer Driver), которая поддерживает подключаемые модули и имеет необходимые для этого COM-интерфейсы, то сторонние разработчики могут разработать для такого драйвера подключаемый модуль маркирования. Однако производители принтеров и драйверов к ним чаще всего не предоставляют такую возможность исходя из коммерческих интересов или используют ее только в своих целях.

Первые два подхода могут использоваться в комбинации. Их объединяет одно – отсутствие открытого API для взаимодействия с драйвером и влияния на рендеринг страниц, и как следствие, невозможность реализации собственных алгоритмов маркирования сторонними разработчиками.

Два других подхода базируются на открытом API системы печати Windows, поэтому являются более перспективными с точки зрения реализуемости поставленной задачи.

Встраивание маркера в очереди печати средствами Print Spooler API – это наиболее старый API системы печати, доступный в user mode, реализованный еще в Windows 2000. Он предоставляет доступ к модификации растрового изображения каждой страницы, выводимой на печать. Из недостатков такого подхода можно отметить следующие:

- низкоуровневый API, имеющий некоторые проблемы в безопасности исполняемого кода;
- сложность в конфигурировании параметров маркирования;
- сложность или невозможность в получении доступа к метаданным печати: идентификационным данным компьютера, пользователя, сессии, процесса, из которого запущена печать.

Реализация маркирования в виртуальном принтере является наиболее мощным и гибким подходом в маркировании документов в Windows. Этот подход не имеет недостатков, характерных для Print Spooler API. Он реализован в подсистеме контроля печати в линейке средств защиты от несанкционированного доступа SecretNet [14].

Для реализации виртуального принтера может быть использована одна из следующих архитектур:

- GDI;
- PostScript Printer Driver;
- Microsoft Universal Printer Driver (Unidrv или V3 Printer Driver);
- V4 Printer Driver.

Архитектура Unidrv обладает хорошо проработанным API, позволяющим гибко управлять процессом печати. Существенным является также и то, что драйверы Unidrv работают в пространстве пользователя (User-mode drivers), а не ядра.

V4 Printer Driver предоставляет более современный и более гибкий API, но он поддерживается только с Windows 8.

Для универсального представления документов в Windows целесообразно использовать формат XPS (XML Paper Specification). Это открытый графический формат разметки документа на основе XML. Среди его преимуществ можно отметить следующее [15]:

- разметка документа проще и легче, чем у PDF;
- используется векторная непоследовательная разметка;
- интегрирован в .NET Framework;
- поддерживает многопоточную работу;
- поддерживает шифрование и цифровые сертификаты.

Архитектура XPS-драйвера принтера полностью совместима с Unidrv. При реализации алгоритма маркирования такой драйвер позволяет получить документ в виде универсальной DOM-структуры. Это дает возможность независимо обрабатывать отдельные части документа, включая текст, изображения, шрифты, ресурсы, метаданные задания, документа и отдельных страниц; добавлять в DOM-структуру видимые и невидимые маркеры, а также другие метаданные.

3.1 Маркирование документа в XPS-фильтре

Поддержка XPS-драйверов доступна в Windows XP; при этом требуется установить либо Microsoft XPS Essentials Pack, либо .NET Framework 3.0. Начиная с Windows Vista, компоненты XPS встроены в операционную систему.

Код, реализующий алгоритм встраивания маркера, в данной работе был реализован как XPS-фильтр, встраиваемый в конвейер драйвера виртуального принтера. Общая архитектура драйвера XPS-принтера представлена на рис. 1.

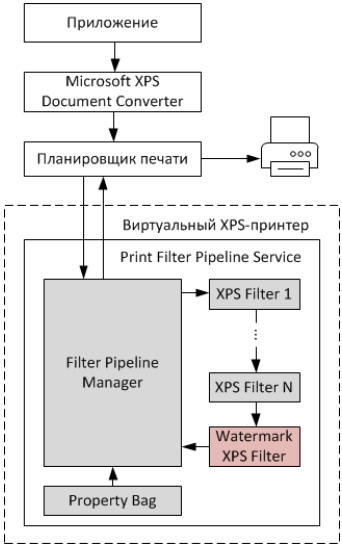

Рис. 1. Общая архитектура драйвера XPS-принтера
Fig. 1. General architecture of the XPS printer driver

Драйвер виртуального XPS-принтера представляет собой конвейер из одного или нескольких XPS-фильтров [16]. Каждый фильтр работает независимо от других. Их взаимодействие между собой осуществляется посредством Filter Pipeline Manager. XPS-фильтр представляет собой COM-объект, что позволяет обеспечить многопоточность, надежность и безопасность всего драйвера.

```
<Filters>
  <Filter dll = "Filter_1.dll"
    clsid = "{5552021F-9D7A-4514-93A6-18FA8EE43601}"
    name = "XpsRenderFilter">
    <Input guid = "{b8cf8530-5562-47c4-ab67-b1f69ecf961e}"
       comment="IID_IXpsDocumentProvider"/>
   <Output guid = "{4368d8a2-4181-4a9f-b295-3d9a38bb9ba0}"
       comment="IID_IXpsDocumentConsumer"/>
 </Filter>
 <Filter dll = "Filter_2.dll"
    clsid = "{C7B17E67-33FF-45DE-8E5B-47A4AFBF370B}"
    name = "XpsRenderFilter">
   <Input guid = "{b8cf8530-5562-47c4-ab67-b1f69ecf961e}"
       comment="IID_IXpsDocumentProvider"/>
  <Output guid = "{4368d8a2-4181-4a9f-b295-3d9a38bb9ba0}"
       comment="IID_IXpsDocumentConsumer"/>
 </Filter>
 <Filter dll = "XpsWatermarkFilter.dll"
    clsid = "{925A12DE-A638-466B-927A-15CBF786C827}"
    name = " XpsWatermarkFilter ">
   <Input guid = "{b8cf8530-5562-47c4-ab67-b1f69ecf961e}"
       comment="IID_IXpsDocumentProvider"/>
   <Output guid = "{4368d8a2-4181-4a9f-b295-3d9a38bb9ba0}"
       comment="IID_IXpsDocumentConsumer"/>
  </Filter>
</Filters>
```

Рис. 2. Пример конфигурационного файла XPS-фильтров
Fig. 2. Example of XPS filters configuration file

Каждый фильтр может обеспечивать определенную функциональность в процессе маркирования документа: нормализацию, выделение областей, маркирование текста,

маркирование изображений, обработку шрифтов и т.д. Обработанный конвейером фильтров документ возвращается в планировщик печати и отправляется в порт физического принтера. Физический принтер, на который выводится готовый документ, указывается в параметрах печати и может быть сконфигурирован в UI-плагине драйвера.

Очередность фильтров в конвейере определяется конфигурационным файлом, пример которого представлен в листинге на рис. 2.

Здесь же определяются стандартизованные COM-интерфейсы для входа и выхода каждого фильтра. Параметры печати передаются фильтрам посредством специального COM-объекта PropertyBag.

Далее будет рассмотрена реализация драйвера с одним XPS-фильтром, осуществляющим маркирование каждой страницы печатаемого документа. Структура фильтра маркирования представлена на рис. 3.

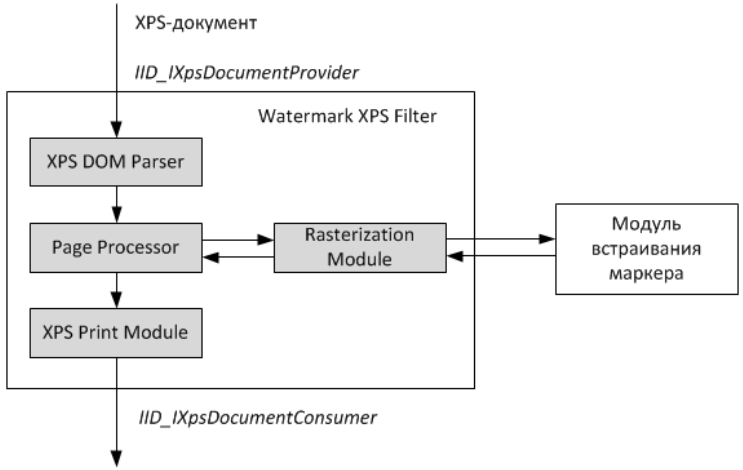

Рис. 3. Структура XPS-фильтра маркирования
Fig. 3. XPS marking filter structure

Сначала XPS DOM Parser разбирает документ или цепочку документов на страницы. Page Processor извлекает страницу, растеризует ее полностью (или часть) и передает модулю встраивания маркера. После встраивания маркера страница уже в виде растра помещается в DOM-модель документа, при этом старая страница заменяется на новую – растеризованную и маркированную.

3.2 Реализация модуля встраивания маркера

Модуль встраивания может быть реализован одним из перечисленных ниже способов.

1) Как многопоточный или однопоточный COM-объект, устанавливаемый вместе с драйвером фильтра XPS. Растровое изображение страницы для обработки передается через разделяемую память процесса виртуального принтера.
2) Как многопоточную или однопоточную динамическую библиотеку с обычным stdcall-интерфейсом. Растровое изображение страницы для обработки передается также через разделяемую память.

Особенностью двух этих способов является то, что исполняемый код модуля работает в изолированной среде процесса печати, что продиктовано соображениями безопасности [17]. Опытным путем было установлено, что в этой изолированной среде запрещен файловый ввод-вывод, а также запрещено создание дочерних процессов. Это накладывает

определенные ограничения на возможность использования внешних программных компонентов и интерпретируемых языков, таких как Python.

3) Как отдельный процесс. Этот способ предоставляет больше возможностей для реализации маркирования. В частности, он может быть реализован как набор скриптов на языке Python и работать в отдельном процессе интерпретатора.

Вместе с тем, при реализации такого способа маркирования возникают две проблемы, обусловленные ограничениями контекста процесса печати:

- невозможность из XPS-фильтра запустить внешний процесс;
- необходимость передачи растеризованной страницы модулю маркирования и приема обратно маркированного изображения.

Внешний процесс можно запустить автоматически в пространстве пользователя в фоновом режиме. Передать данные из драйвера во внешний процесс через файл невозможно. Для этих целей могут быть использованы следующие механизмы межпроцессного взаимодействия:

- именованные каналы;
- проецирование в память области из системного файла подкачки;
- сокеты.

Для синхронизации с внешним процессом драйвер может использовать именованный канал или объекты ядра для межпроцессной синхронизации.

4) Как TCP-сервер. Этот способ является разновидностью предыдущего, если для взаимодействия с внешним процессом использовать сокет. Преимуществом данного способа является то, что сервер маркирования может быть размещен на отдельном сетевом узле, что позволит легко масштабировать систему маркирования и оптимизировать вычислительные ресурсы сети организации.

Авторами был реализован четвертый способ. При этом модуль маркирования был реализован как скомпилированный бинарный Python-скрипт, запускаемый HTTP-сервером, написанным на C#. Структура модуля маркирования представлена на рис. 4.

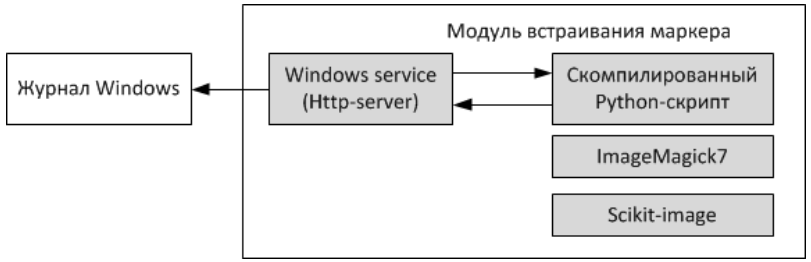

Рис. 4. Структура модуля маркирования
Fig. 4. Marking module structure

HTTP-сервер запускает отдельный процесс маркирования в синхронном режиме. Обмен данными осуществляется посредством временных файлов, создаваемых в кэше файловой системы.

3.3 Обработка задания на печать

В этом подразделе детально описан процесс обработки документа, поступающего на печать в XPS-фильтр.

Данные для печати передаются в фильтр системой печати через интерфейс IID_IXpsDocumentProvider (рис. 3). DOM-структура данных для печати включает в себя четыре вложенных уровня (рис. 5) [15]:

- задание для печати (XpsDocument);
- последовательность документов (FixedDocumentSequence);
- документ (FixedDocument);
- страница (FixedPage).

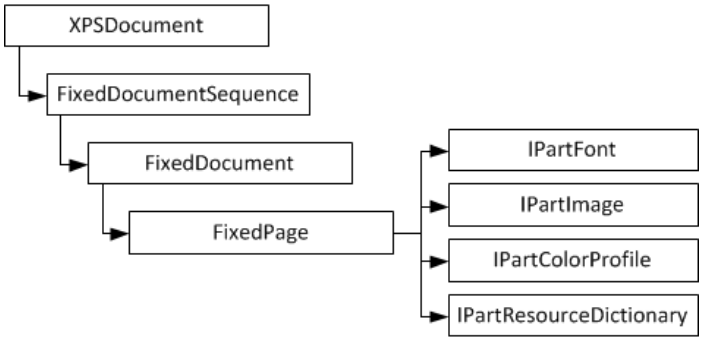

Рис. 5. DOM-структура данных для печати
Fig. 5. DOM data structure for printing

Фильтр запрашивает последовательно объект каждого уровня DOM-структуры, начиная с верхнего уровня. Таким образом, фильтр выполняет обход всей структуры данных в глубину.

Для объектов XpsDocument, FixedDocumentSequence, FixedDocument фильтр обрабатывает только метаданные и передает их на сервер маркирования в виде JSON-пакетов. Объекты FixedPage обрабатываются отдельно модулем PageProcessor (рис. 3).

Обработка FixedPage заключается в выполнении следующих действий:

- извлечение структуры страницы, включая текст, ссылки на ресурсы и данные верстки;
- извлечение внедренных объектов: шрифтов, изображений, цветовых профилей, словарей;
- упаковка извлеченных данных в пакет и передача его на HTTP-сервер;
- извлечение данных из пакета, растеризация и маркирование изображения страницы внешним модулем на HTTP-сервере;
- вывод маркированной страницы на физический принтер.

Опытным путем было установлено, что при печати в XPS-принтер из некоторых приложений (например, Microsoft Word), подсистема Microsoft XPS Document Converter внедряет в XPS-документ шрифты ODTTF в обфусцированном виде с целью защиты правообладателя. Однако, алгоритм обфускации шрифтов был раскрыт исследователями [18]. Для рассматриваемого модуля маркирования был разработан алгоритм деобфускации шрифтов и реализован в HTTP-сервере.

Сокращенный код деобфускации шрифтов ODTTF приведен на рис. 6.

```
int[] guidMapping = new int[] {15,14,13,12,11,10,9,8,6,7,4,5,0,1,2,3};
for (int i = 0; i < 16; i++) {
    buf[i] = (byte)(buf[i] ^ guid[guidMapping[i]]);
    buf[i + 16] = (byte)(buf[i + 16] ^ guid[guidMapping[i]]);
}
```

Рис. 6. Код деобфускации шрифтов ODTTF
Fig. 6. ODTTF font deobfuscation code

3.4 Многопоточная реализация сервера маркирования

В процессе маркирования документов наиболее ресурсоемкой операцией является внедрение маркера в растеризованную страницу. Поэтому печать документов, содержащих множество страниц большого формата (A3 и более), может занимать продолжительное время, зависящее от сложности алгоритма маркирования. Поскольку маркирование осуществляется в отдельном фоновом процессе Windows, это может потенциально порождать коллизии при одновременной печати нескольких документов одним или разными пользователями через один и тот же виртуальный XPS-принтер. Чем больше объем печатаемых документов, тем выше вероятность коллизии.

Для предотвращения коллизий и улучшения производительности системы маркирования предложена модифицированная схема с многопоточным сервером (рис. 7).

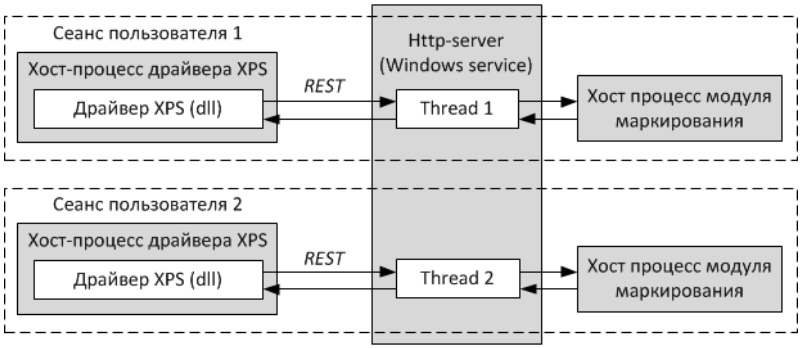

Рис. 7. Схема маркирования с многопоточным сервером
Fig. 7. Marking scheme with multithreaded server

Поскольку для каждого задания, отправляемого пользователем на печать, запускается код драйвера в отдельном хост-процессе, то для маркирования страниц HTTP-сервер создает отдельный поток для каждого подключения драйвера.

Маркирование осуществляется внешним процессом, каждый экземпляр которого запускается из потока сервера. Протокол обмена данными с модулем маркирования простой и позволяет синхронизироваться только с дескриптором процесса.

Протокол взаимодействия драйвера с сервером требует, чтобы состояние сессии сохранялось между HTTP-запросами. Это нужно, в частности, для реализации процедуры отмены печати или для отслеживания прогресса печати. Сохранение состояния сессии может быть реализовано двумя способами [19, 20]:

- с помощью менеджера сессий;
- на основе механизма REST.

Менеджер сессий усложняет архитектуру сервера, поскольку он не только должен хранить состояние каждой сессии, но и отслеживать ее время жизни, находить и удалять "мертвые" сессии, осуществлять «сборку мусора».

Механизм REST (Representation State Transfer) позволяет сохранять состояние сессии в GET или POST запросах, тем самым упрощая архитектуру сервера. Такой способ был реализован в системе, разработанной авторами данной работы.

Предлагаемая многопоточная схема обладает высокой надежностью, поскольку все сеансы печати изолированы друг от друга, и сбой при маркировании или печати в одном сеансе не повлияет на все остальные

3.5 Обработка метаданных XPS-документа

Для маркирования и последующей печати маркированных документов необходимы метаданные, которые должны передаваться драйверу XPS-принтера помимо DOM-структуры документа:

- для растеризации необходимо разрешение страниц (в DPI);
- для маркирования необходимы данные для генерации маркера, позволяющего идентифицировать пользователя, который запустил печать;
- для печати необходимы данные о физическом принтере, размере и ориентации страницы, цветовой схеме, разрешении и другие параметры.

Все необходимы данные могут быть получены драйвером через стандартные COM-интерфейсы из структуры PropertyBag (рис. 1).

Идентификационные данные для генерации маркера могут быть получены из системного токена пользователя, запустившего печать. Дескриптор токена, в свою очередь, передается драйверу через PropertyBag. Упрощенный код извлечения идентификационных данных представлен ниже (рис. 8):

```
VARIANT varUserSecurityToken;
VariantInit(&varUserSecurityToken);
m_pIPropertyBag->GetProperty(XPS_FP_USER_TOKEN,
  &varUserSecurityToken));

HANDLE user_handle = varUserSecurityToken.byref;
DWORD tuLen = 0;
GetTokenInformation(user_handle, TokenUser, NULL, 0, &tuLen);
PTOKEN_USER tu = (PTOKEN_USER)LocalAlloc(LPTR, tuLen);
GetTokenInformation(user_handle, TokenUser, tu, tuLen, &tuLen);

const size_t max_len = 256;
wchar_t nameUser[max_len] = { 0 };
wchar_t domainName[max_len] = { 0 };
DWORD nameUserLen = 256;
DWORD domainNameLen = 256;
SID_NAME_USE snu;
LookupAccountSidW(NULL, tu->User.Sid, nameUser, &nameUserLen,
  domainName, &domainNameLen, &snu);
```

Рис. 8. Код извлечения идентификационных данных для генерации маркера
Fig. 8. Identification data extraction code for mark generation

Параметры, используемые для растеризации и печати, извлекаются из структуры PrintTicket, доступ к которой может быть получен через PropertyBag, аналогично дескриптору токена пользователя.

DOM-структура PrintTicket соответствует спецификации Print Schema Specification [21]. В соответствие с ней параметры печати задаются на трех уровнях:

- задание (Job);
- документ (Document);
- страница (Page).

При обработке параметров печати драйвер выполняет слияние объектов PrintTicket с разных уровней по схеме JobPrintTicket -> DocumentPrintTicket -> PagePrintTicket.

Слияние осуществляется в соответствии с принципом наследования. Т.е. все параметры, установленные для задания, наследуются или переопределяются на уровне документа и страницы. Для слияния параметров используется системная функция PTMergeAndValidatePrintTicket.

Различные приложения, из которых осуществляется печать, могут по-разному формировать объекты PrintTicket для всех трех уровней. Так, например, Microsoft Word формирует параметры печати для всех трех уровней, а Notepad формирует параметры печати только для задания (PrintTicket для страницы не содержит данных).

Для управления параметрами печати в XPS-драйвере целесообразно использовать интерфейсы и компоненты Unidrv. В их составе имеется набор типовых параметров печати, таких как стандартные размеры бумаги, ориентация страниц, разрешение, качество печати и т.д. Кроме того, Unidrv предоставляет типовой пользовательский интерфейс для задания пользователем параметров печати.

Для настройки параметров печати, не предусмотренных Unidrv, DOM-структура PrintTicket может быть расширена, так как это предусмотрено спецификацией Print Schema Specification. Unidrv предоставляет COM-интерфейсы для расширения и модификации страниц свойств интерфейса настройки параметров печати драйвера XPS.

Для рассматриваемой системы маркирования было разработано расширение стандартной схемы параметров печати (листинг, рис. 9.) и выполнена модификация интерфейса настройки параметров Unidrv путем добавления новой страницы свойств.

```
<psf:Feature name="psk:PageWatermarking">
    <psf:Option>
        <psf:ScoredProperty name="psk:PageRedirectPrinterName">
        <psf:ParameterRef
           name="psk:PageWatermarkingPageRedirectPrinterName"/>
        </psf:ScoredProperty>
    </psf:Option>
</psf:Feature>
<psf:ParameterInit
  name="psk:PageWatermarkingPageRedirectPrinterName">
    <psf:Value xsi:type = "xsd:string">
        hp LaserJet 1010 HB
    </psf:Value>
</psf:ParameterInit>
```

Рис.9. Фрагмент расширенной схемы параметров печати
Fig. 9. Fragment of the extended print settings scheme

После того, как параметры печати попадут в драйвер, они преобразуются в бинарную упакованную структуру DEVMOD для последующей обработки конвейером печати. Это осуществляется с помощью системной функции PTConvertPrintTicketToDevMode.

4. Требования к процессу маркирования

Ключевыми требованиями к процессу маркирования являются:

- возможность извлечения параметров встраиваемого маркера и встроенной информации при отсутствии исходного документа;
- обеспечение выполнения требований по робастности относительно заданных искажений;
- обеспечение должной емкости встраивания для отдельных документов (изображений документов);
- обеспечение заданного уровня достоверности извлечения встроенной информации (с учетом возможного применения помехоустойчивого кодирования при встраивании информации);
- инвариантность изменяемых в процессе маркирования характеристик документа к виду его представления или преобразования на основе которого был получен его электронный эквивалент.

Далее будут рассмотрены основные задачи решаемые в процессе внедрения и извлечения маркера и требования к ним. Изменяемые характеристики документа при этом могут быть различными, но должны выполняться указанные выше требования.

4.1 Формирование и внедрение маркера

Одним из возможных вариантов внедрения маркера является кодирование информации за счет изменения яркости фона отдельных областей маркируемого документа при встраивании битов маркера. Формирование маркера при этом состоит из следующих этапов:

- кодирование встраиваемой информации, содержащей сведения об авторе, дате и времени

формирования документа в двоичный вид;

- преобразование полученной информации в последовательность встраивания по следующему правилу: символу "1" соответствует наличию изменяемой яркости фона определенной области документа, символу "0" – отсутствие изменений в яркости фона документа.

Полученная последовательность встраивания внедряется в электронный документ в процессе печати посредством разработанного подхода, основанного на технологии виртуального XPS-принтера.

Указанный подход к формированию маркера позволяет осуществлять маркирование не только электронных текстовых документов, но электронных таблиц и электронных документов, содержащих графические объекты, диаграммы и изображения, а также электронных документов, подготовленных для демонстрации и визуального восприятия.

В результате маркирования формируется печатный документ, содержащий информацию, однозначно идентифицирующую владельца данных. В процессе жизненного цикла распечатанный документ может быть подвергнут преобразованию формата в электронный вид посредством операции сканирования или фотографирования. Для извлечения встроенного маркера из сформированного изображения предложен подход к извлечению встроенного маркера, основанного на изменении яркости фона документа.

4.2 Извлечение встроенного маркера

В процессе преобразования формата напечатанного документа в изображение могут быть внесены следующие искажения:

- поворот изображения;
- масштабирование изображения;
- искажение перспективы изображения (горизонтальный наклон и вертикальное отклонение);
- наличие областей изображения, содержащих неравномерный фон.

Для устранения указанных искажений, обусловленных процессом фотографирования (сканирования) предлагается подход к извлечению маркера из изображений, состоящий из следующих этапов:

- коррекция перспективы (поворота) изображения;
- определение контуров (границ) документа;
- определение областей с измененными характеристиками;
- декодирование значений яркости в двоичный вид;
- идентификация автора документа и его атрибутов.

В ходе экспериментальной оценки предложенного подхода к маркированию, основанного на изменении областей фона изображения посредством реализации виртуального XPS-принтера в операционных системах семейства Windows, была осуществлена серия экспериментов, направленных на определение возможности извлечения встроенной информации из изображений электронного документа.

Полученные значения точности извлечения встроенного маркера при осуществлении преобразования "печать-сканирование" или "печать-фотографирование" (порядка 0,95 для отдельных документов и маркера размером 16 бит) позволили сделать вывод о наличии потенциальной возможности реализации системы маркирования на основе виртуального XPS-принтера для защиты авторских прав владельцев электронных документов. Однако реализация данной системы требует проведения существенно большего количества

экспериментальных исследований для обоснования выбираемых параметров и требований к процессу маркирования, что является направлением дальнейших исследований.

5. Выводы и направление дальнейших исследований

Предложенный авторами подход позволяет реализовать надежное внедрение маркера при печати документов в операционных системах семейства Windows через виртуальный XPS-принтер, установленный в качестве принтера по умолчанию для всех пользователей, а также, в случае наличия соответствующих алгоритмов маркирования, преодолеть недостатки существующих средств, требующих наличия оригинального документа для проведения расследования инцидента информационной безопасности.

Дальнейшего исследования требуют следующие вопросы:

- выбор и обоснование характеристик электронного документа, используемых для внедрения маркера с учетом предъявляемых требований;
- оптимизация протокола взаимодействия драйвера и HTTP-сервера;
- оценка ресурсозатрат по времени и производительности в зависимости от используемого типа электронно-вычислительных машин и средств вычислительной техники;
- улучшение производительности модуля встраивания маркера путем использования специализированных процессоров или ПЛИС;
- разработка подхода к внедрению маркера в подсистеме печати операционных систем семейства UNIX;
- разработка сетевой архитектуры системы маркирования, вынесение наиболее ресурсоемких операций на отдельную вычислительную платформу.

Список литературы / References

[1]. Глобальное исследование утечек конфиденциальной информации в первом полугодии 2019 года. InfoWatch. 2019, 30 стр. / Global Confidential Information Leak Survey in the first half of 2019. InfoWatch. 2019, 30 p. (in Russian).
[2]. Исследование утечек информации ограниченного доступа в госсекторе. Мир – Россия. InfoWatch. 2019, 24 стр. / Research of information leaks of limited access in the public sector. World – Russia. InfoWatch. 2019, 24 p. (in Russian).
[3]. M. Jain, S. K. Lenka. A Review on Data Leakage Prevention using Image Steganography / // International Journal of Computer Science Engineering, vol. 5, issue 2, 2016, pp. 56-59.
[4]. Lopez G., Richardson N., Carvajal J. Methodology for Data Loss Prevention Technology Evaluation for Protecting Sensitive Information. Revista Politécnica, vol. 36, issue. 3, 2015, pp. 1-69.
[5]. Alneyadi S., Sithirasenan E., Muthukkumarasamy V. A survey on data leakage prevention systems. Journal of Network and Computer Applications, vol. 62, 2016, pp. 137-152.
[6]. Jadhav P., Chawan P. M. Data Leak Prevention system: A Survey. International Research Journal of Engineering and Technology, vol. 6, issue 10, 2019, pp. 197-199.
[7]. Milano D. Content control: Digital watermarking and fingerprinting. White Paper. Rhozet, a business unit of Harmonic Inc., 2012, 11 p.
[8]. Data loss prevention in Exchange Server. Available at: https://docs.microsoft.com/en-us/Exchange/policy-and-compliance/data-loss-prevention/data-loss-prevention?redirectedfrom=MSDN&view=exchserver-2019, accessed 14.09.2020.
[9]. Graham R. How The Intercept Outed Reality Winner. Available at: https://blog.erratasec.com/2017/06/how-intercept-outedreality-winner.html, accessed 14.09.2020.
[10]. Trace Doc. Available at: https://secretgroup.ru/trace-doc, accessed 14.09.2020.
[11]. Kozachok A. V., Kopylov S. A., Shelupanov A. A., Evsutin O. O. Text marking approach for data leakage prevention. Journal of Computer Virology and Hacking Techniques. 2019, vol. 15, issue. 3, pp. 219-232. DOI: 10.1007/s11416-019-00336-9.
[12]. Unique Interface. EveryTag. Available at: https://everytag.ru/ui, accessed 14.09.2020.
[13]. Safe Copy. Available at: https://www.niisokb.ru/products/safecopy, accessed 14.09.2020.

[14]. Secret Net Studio. Available at: https://www.securitycode.ru/products/secret-net-studio/, accessed 14.09.2020.
[15]. Open XML Paper Specification (OpenXPS). Standard ECMA-388. 2009, 496 p.
[16]. XPSDrv Render Module. Available at: https://docs.microsoft.com/ru-ru/windows-hardware/drivers/print/xpsdrv-render-module, accessed 14.09.2020.
[17]. Understanding Printer Driver Isolation. Available at: https://sourcedaddy.com/windows-7/understanding-printer-driver-isolation.html, accessed 14.09.2020.
[18]. Celil U., Bekir K. Breaking Font Parsers. Available at: http://www.powerofcommunity.net/poc2015/celil.pdf, accessed 14.09.2020.
[19]. Архитектура REST / REST architecture. Available at: https://habr.com/ru/post/38730, accessed 14.09.2020 (in Russian).
[20]. Введение в REST API – RESTful веб-сервисы / Introduction to REST API – RESTful web services. Available at: https://habr.com/ru/post/483202, accessed 14.09.2020 (in Russian).
[21]. Print Schema. Available at: https://docs.microsoft.com/ru-ru/windows/win32/printdocs/printschema, accessed 14.09.2020.

Информация об авторах / Information about authors

Сергей Викторович КОЗЛОВ – кандидат технических наук. Сфера научных интересов: информационная безопасность, защита от несанкционированного доступа, особенности построения и функционирования операционных систем, средства и методы программирования.

Sergey Viktorovich KOZLOV – Candidate of Technical Sciences. Research interests: information security, protection from unauthorized access, construction and functioning features of operating systems, programming tools and methods.

Сергей Александрович КОПЫЛОВ, научные интересы включают методы машинного обучения, обработка цифровых изображений, текстовая стеганография.

Sergey Alexandrovich KOPYLOV, research interests include machine learning methods, digital image processing, text steganography.

Борис Владимирович КОНДРАТЬЕВ; сфера научных интересов: безопасность информации, защита информации от несанкционированного доступа и утечки по техническим каналам, построение информационных систем в защищённом исполнении, сертификация программного обеспечения по требованиям безопасности информации.

Boris Vladimirovich KONDRAT'EV, research interests: information security, protection of information from unauthorized access and leakage through technical channels, building information systems in a secure design, certification of software for information security requirements.

Дмитрий Олегович ОБЫДЕНКОВ – аспирант. Его научные интересы включают методы сокрытия и защищённой передачи информации, компьютерные сети, технологии анализа сетевого трафика.

Dmitry Olegovich OBYDENKOV is a graduate student. His scientific interests include methods for information hiding and secure transmission, computer networks, technologies of network traffic analysis.