Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

УДК 681.382.26

**ПОРТЯНКО СЕРГЕЙ СЕРГЕЕВИЧ**

# ЗАЩИТА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРИ ПОМОЩИ ЦИФРОВЫХ ВОДЯНЫХ ЗНАКОВ

05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Автореферат диссертации
на соискание ученой степени кандидата технических наук

Минск 2006

Работа выполнена в Учреждении образования «Белорусский государственный университет информатики и радиоэлектроники»

Научный руководитель – д.т.н., профессор Ярмолик В.Н., кафедра программного обеспечения информационных технологий, БГУИР

Официальные оппоненты: д.т.н., профессор Птичкин В.А. кафедра информационных технологий автоматизированных систем, БГУИР

к.т.н. Захаров В.В. унитарное предприятие "Творческая лаборатория"

Оппонирующая организация – Государственное научное учреждение "Объединённый институт проблем информатики НАН Беларуси"

Защита состоится 21 сентября 2006 г. в 14 часов на заседании совета по защите диссертаций Д 02.15.04 при Учреждении образования «Белорусский государственный университет информатики и радиоэлектроники» по адресу: 220013, г. Минск, ул. П. Бровки, 6, БГУИР, 1 уч. корпус, ауд. 232, тел. 293-89-89.

# ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

***Актуальность темы диссертации.*** В области разработки программного обеспечения (ПО) существуют достаточно серьёзные проблемы, связанные с нарушением прав интеллектуальной собственности разработчиков, обусловленные наличием таких угроз для программных продуктов как: несанкционированное использование, обратное проектирование и несанкционированная модификация. Как показывает мировая практика, уязвимость ПО к перечисленным угрозам является причиной существенных экономических потерь для компаний-производителей.

Данные проблемы частично решаются за счёт применения ряда традиционных юридических и технических мер по предотвращению нарушений прав интеллектуальной собственности разработчиков ПО. К ним относятся: патентование и заключение лицензионных соглашений; программно-аппаратные методы защиты; методы, основанные на криптографическом преобразовании кода программного обеспечения; методы, основанные на привязке программного обеспечения к носителю; организация выполнения ПО на удалённом сервере ("server-side execution"). Однако, перечисленные подходы не являются достаточно эффективными, что подтверждается ежегодно фиксируемыми ассоциацией Business Software Alliance многомиллиардными потерями мировой индустрии ПО. В частности, использование юридических мер затруднено несоответствием законодательства в различных государствах и ограниченной территорией действия патентов. Программно-аппаратная защита, основанная на использовании аппаратных ключей, уязвима к эмуляции работы ключа либо модификации защищаемой программы с целью полного отключения защиты. Недостатком методов, основанных на дешифровании кода программы "на лету" ("on-the-fly"), является наличие возможности считывания из оперативного запоминающего устройства вычислительной машины значений ключей дешифрования либо кода программы в открытом виде. Привязка ПО к носителю создаёт неудобства его использования и неприменима при доставке программного продукта конечному пользователю путём загрузки с web-сайта разработчика. Выполнение ПО на удалённом сервере связано с существенной потерей производительности приложений.

Недостаточная эффективность традиционных методов защиты ПО послужила причиной происходящего в последние годы интенсивного развития принципиально новых подходов к защите ПО, основанных на использовании цифровых водяных знаков (ВЗ). На сегодняшний день предложено множество алгоритмов внедрения водяных знаков в ПО, однако остаются открытыми и продолжают активно обсуждаться такие вопросы как обеспечение защиты, минимально влияющей на качество ПО и удобство его использования, а также создание технологий защиты, устойчивых к применению существующих контрмер.

В связи с этим разработка новых подходов, методов, алгоритмов внедрения водяных знаков в ПО и реализация на их основе эффективных технических средств защиты является весьма актуальной.

***Связь работы с крупными научными программами, темами.*** Исследования проводились в рамках задания ГБЦ 02-3086 "Разработать теоретические основы построения систем защиты цифровой интеллектуальной собственности в современных условиях развития информационных технологий" Государственной программы ориентированных фундаментальных исследований "Научные основы новых информационных технологий" ("Инфотех"), выполненной НИЛ 3.3 БГУИР; в рамках международного гранта BMBF BLR 02/006 "DSP-based control systems for safety-related applications"; в рамках научно-исследовательской работы ГБ 01-2004 "Разработать методы, алгоритмы и программные модули для исследования характеристик, оценки технического состояния и диагностирования сложных систем", выполненной кафедрой ПОИТ БГУИР; в рамках научно-исследовательской работы ГБЦ 04-3072 "Разработка методов, алгоритмов и программных средств для проектирования отказоустойчивых микросистем с перестраиваемой архитектурой на базе перепрограммируемой логики", выполненной НИЛ 3.3 БГУИР; в рамках научно-исследовательской работы ГБЦ 02-3178 "Разработать теорию, методы и программно-аппаратные средства для тестирования и обеспечения надежности высокопроизводительных систем цифровой обработки сигналов с перестраиваемой архитектурой", выполненной НИЛ 3.3 БГУИР.

***Цель и задачи исследования.*** Целью диссертационной работы является разработка эффективных методов защиты ПО от несанкционированного использования, обратного проектирования и модификации.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ существующих методов защиты ПО от несанкционированного использования, обратного проектирования и модификации;
- провести анализ ПО с точки зрения возможности внедрения информации, идентифицирующей автора либо фирму-разработчика;
- разработать эффективные методы защиты внедряемой информации от атак типа "искажение" и "удаление";
- разработать эффективные алгоритмы постановки водяных знаков на уровне исходного кода на языке высокого уровня;
- разработать эффективные алгоритмы постановки водяных знаков на уровне машинного кода;
- разработать метод защиты программных модулей при помощи многоуровневой системы внедрения водяных знаков и "отпечатков пальцев".

***Объект и предмет исследования.*** Объектом исследования является программное обеспечение. Предметом исследования являются методы технической защиты авторских прав разработчиков программного обеспечения.

***Методология и методы проведённого исследования.*** Решение рассматриваемых в диссертации задач основывается на методологии объектного программирования, методах стеганографии.

***Научная новизна и значимость полученных результатов.***

1. Определены характеристики программных модулей, обладающие постоянными, то есть независящими от функциональности программы свойствами, что допускает их использование для размещения водяного знака.

2. Разработан метод внедрения признака авторства в машинный код, основанный на подходе, использованном в методе Patchwork, применяемом для внедрения водяного знака в растровые изображения, что обеспечивает повышенную устойчивость внедряемой информации к обнаружению и удалению.

3. Впервые предложен подход к внедрению водяных знаков в ПО, основанный на преобразовании машинного кода, приводящем к модификации вида его автокорреляционной характеристики; предложен подход к внедрению водяных знаков в ПО, основанный на модификации свойств числовой последовательности, получаемой в результате интерпретации очерёдности следования независимых команд, что позволяет снизить объём преобразований кода, необходимых для размещения информации, идентифицирующей автора.

4. Впервые предложен способ совместного использования разработанных методов внедрения водяного знака на уровне исходного текста программы и на уровне машинного кода, при котором защита информации, внедряемой на низком уровне, обеспечивается при помощи водяного знака верхнего уровня, что повышает стойкость внедрённой информации к применению известных атак.

***Практическая значимость полученных результатов.***

1. Предложенные в диссертационной работе методы внедрения водяных знаков могут быть использованы для защиты ПО, доставляемого конечному пользователю путём загрузки с web-сайта разработчика.

2. Предложенный метод внедрения информации об авторе на уровне исходного кода обеспечивает защиту свободно распространяемых исходных кодов ПО от несанкционированного использования, а также может быть применён для защиты приложений, созданных по технологии “Active Server Pages”.

3. Разработанный метод размещения признака авторства на уровне машинного кода обеспечивает эффективную защиту готовых программных продуктов, исполняемые модули которых скомпилированы под конкретную платформу, и не требует наличия исходных кодов защищаемых приложений.

4. Результаты диссертационной работы внедрены и использованы на предприятии “ЛАПТА” в системе внедрения многоуровневых водяных знаков и “отпечатков пальцев” «EMWERS», используемой для контроля за распространением экземпляров разрабатываемого программного продукта и защиты отдельных программных модулей. Результаты диссертационной работы: методы построения систем защиты прав интеллектуальной собственности на программное обеспечение внедрены в учебный процесс на кафедре ПОИТ БГУИР

в качестве лабораторного и лекционного материала для курсов "Защита информации" и "Элементы теории информации".

***Основные положения диссертации, выносимые на защиту.***

1. Новый метод внедрения водяных знаков в программное обеспечение, основанный на подходе, использованном в методе внедрения водяного знака в растровые изображения (метод Patchwork), и характеризуемый повышенной устойчивостью внедряемой информации к попыткам обнаружения и удаления.

2. Методы внедрения признака авторства в машинный код, основанные на преобразовании машинного кода, приводящем к модификации вида его автокорреляционной характеристики, и модификации свойств числовой последовательности, получаемой в результате интерпретации очерёдности следования независимых команд, отличающиеся от существующих методов меньшим объёмом преобразований кода, необходимых для размещения информации, идентифицирующей автора.

3. Метод внедрения водяных знаков в ПО на уровне исходного текста, обеспечивающий возможность их распознавания на уровне машинного кода и обладающий повышенной устойчивостью к попыткам удаления и искажения.

4. Технология защиты программ от несанкционированного использования, основанная на многоуровневой системе водяных знаков, обеспечивающей защиту водяного знака низкого уровня при помощи механизма, используемого для размещения водяного знака высокого уровня, и выполняющей дополнительную функцию защиты от обратного проектирования и модификации. Программный комплекс защиты ПО, реализующий разработанные методы внедрения водяных знаков, обеспечивающий внедрение средств защиты программного обеспечения от несанкционированного использования, обратного проектирования и модификации как на этапе создания, так и на этапе формирования дистрибутивов из бинарных исполняемых файлов.

***Личный вклад соискателя.*** Все вошедшие в диссертацию результаты получены при непосредственном личном участии автора и обсуждались с научным руководителем и сотрудниками научно-исследовательской лаборатории №3.3 БГУИР. В публикациях с соавторами вклад соискателя определяется рамками излагаемых в диссертации результатов.

***Апробация результатов диссертации.*** Результаты диссертационной работы докладывались и обсуждались на International Conference on Computer Information Systems and Industrial Management Applications (Элк, Польша, 2003), на International Multi-Conference ACS-CISIM "Computer Information Systems and Applications" (Элк, Польша, 2004), на Белорусско-российской научно-технической конференции "Технические средства защиты информации" (Минск-Нарочь, 2003), на VIII международной научно-технической конференции "Современные средства связи" (Нарочь, 2003), на II Белорусско-российской научно-технической конференции "Технические средства защиты информации" (Минск-Нарочь, 2004), на X Всероссийской научно-технической конференции студентов, молодых ученых и специалистов "Новые информационные техноло-

гии в научных исследованиях и в образовании" (Рязань, Россия, 2005), на двух конференциях студентов, магистрантов и аспирантов, проводимых в Учреждении образования Белорусский государственный университет информатики и радиоэлектроники (Минск, 2003; Минск, 2005).

***Опубликованность результатов.*** По теме диссертации опубликовано 15 печатных работ, из них 4 статьи в научных журналах, 11 работ в сборниках трудов и материалов конференций. Суммарный объем публикаций составляет 63 страницы. Результаты работы включены в 4 отчета по НИР.

***Структура и объем диссертации.*** Диссертация изложена на 147 страницах машинописного текста, в том числе основная часть – на 107 страницах, содержит 63 рисунка, 14 таблиц, состоит из введения, четырёх глав, заключения, списка литературы, включающего 101 название, и четырёх приложений.

## ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обозначена проблема компьютерного пиратства как одна из причин, негативно сказывающихся на развитии индустрии разработки ПО. Кратко охарактеризованы современные подходы к решению проблемы. Показана актуальность исследования в данной области.

В **первой главе** проведён критический анализ существующих подходов к защите программного обеспечения от таких угроз как: несанкционированное использование, обратное проектирование и модификация. Выделены следующие классы приложений, наиболее остро нуждающихся в обеспечении защиты:

- ПО, легко подвергаемое декомпиляции (получению исходного кода из исполняемого);

- ПО, поставляемое пользователю в виде исходных кодов на языке высокого уровня;

- ПО, из которого легко извлекаются интересующие атакующего функционально законченные модули (компоненты).

Рассмотрены наиболее значимые достижения в области защиты ПО при помощи водяных знаков и отмечены нерешенные проблемы такие как: разработка методов повышения устойчивости внедряемой информации против атак типа "искажение" и "удаление"; разработка методов, ориентированных на защиту готовых программ, представленных в виде машинно-ориентированного кода; разработка эффективных методов внедрения водяных знаков в программное обеспечение, представленное в виде исходных текстов. С учётом обозначенных проблем и классов приложений, требующих обеспечения защиты, сформулированы требования к новым методам внедрения водяных знаков:

- возможность обеспечения защиты как на уровне машинного кода, так и на уровне исходного кода на языке высокого уровня;

- применимость для внедрения информации, идентифицирующей разработчика, в готовые бинарные исполняемые модули ПО;

- устойчивость к атакам типа "искажение" и "удаление".

Во **второй главе** проведён анализ характеристик программного обеспечения, допускающих их использование для внедрения водяного знака. Основная идея предлагаемого в рамках диссертационной работы подхода к внедрению водяного знака в программное обеспечение заключается в использовании статистических характеристик машинного кода, обладающих постоянными, то есть в малой степени зависящими от реализованной в программе функциональности, свойствами. Для анализа статистических характеристик машинного кода были выбраны приложения, ориентированные на IA32-архитектуру, что объясняется её чрезвычайной распространённостью.

В результате проведённого исследования определены следующие статистические характеристики IA32 машинного кода, обладающие постоянными свойствами:

- соотношение частот использования определённого подмножества ассемблерных команд;
- автокорреляция последовательности машинных команд;
- характеристика монотонности машинного кода.

На рис. 1 приведены частоты использования 36 мнемоник команд, для которых дисперсия частоты использования, подсчитанная для исследованного множества программ, не превышает двух процентов от среднего значения.

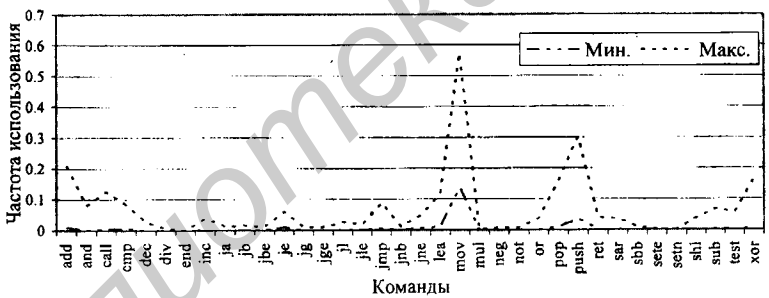



Рис. 1. Команды, обладающие наиболее стабильными частотами использования

Для машинного кода процессоров с IA32-архитектурой существует определённая закономерность или, другими словами, типичный вид характеристики, представляющей собой соотношение частот использования команд с различными мнемониками. Данная закономерность в некоторой мере аналогична закономерностям, присущим естественным языкам, для которых частоты использования отдельных букв примерно постоянны и практически не зависят от смысла текста. Для случая IA32 машинного кода, существует некоторое подмножество ассемблерных команд, частоты использования которых являются стабильными и практически не зависят от реализованной в программе функциональности.

Рассмотрим величину, определяемую как:

$$m_x = \sum_{i=1}^{N} I(C_i) \cdot f(C_i),$$

где $C_i$ – мнемоника машинной команды, расположенной в программе на $i$-й позиции; $I(C_i)$ – индекс (вес), соответствующий мнемонике $C_i$; $f(C_i)$ – частота использования команды с мнемоникой $C_i$; $N$ – общее количество в программе команд с мнемониками, принадлежащими выбранному подмножеству.

Полученные результаты анализа выбранного множества программных модулей свидетельствуют о том, что для программного модуля, содержащего 3000 и более машинных команд, значение величины $m_x$ с большой вероятностью будет располагаться в интервале 6 – 9. Данное свойство может быть рассмотрено как одно из постоянных, то есть не зависящих от функциональности программы, свойств IA32 машинного кода.

На рис. 2 изображены полученные экспериментально минимальные, максимальные и усреднённые значения величины $Sn(k)$, характеризующей автокорреляцию машинного кода и определяемой как:

$$Sn(k) = \frac{1}{N} \sum_{i=0}^{N-1} \begin{cases} 1, P(i) = P((i+k) \bmod N) \\ 0, P(i) \neq P((i+k) \bmod N) \end{cases},$$

где $N$ – общее количество команд в программе; $P(i)$ – индекс, соответствующий мнемонике команды, расположенной в программе на $i$-й позиции; $k$ – величина сдвига; *mod* – взятие остатка от деления.

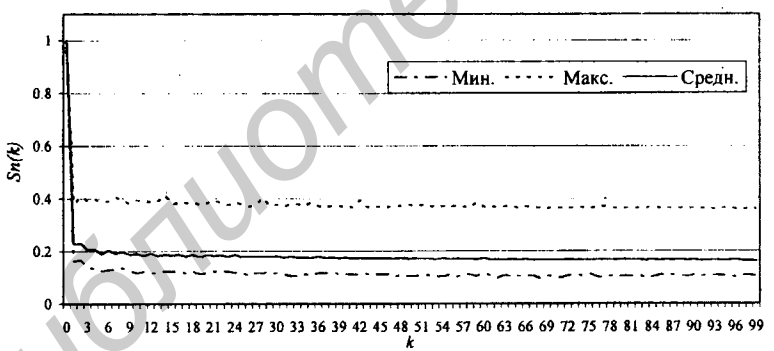



Рис. 2. Экспериментально полученные зависимости $Sn(k)$

Полученные результаты анализа данной характеристики позволяют рассматривать свойство IA32 машинного кода, которое заключается в том, что для программы, содержащей 3000 и более команд, значение отношения $Sn(86)/Sn(87)$ с большой вероятностью будет находиться в интервале 0.8 – 1.1, как постоянное и не зависящее от функциональности программы.

Рассмотрим бинарную последовательность, получаемую в результате интерпретации очерёдности следования независимых команд в программе. Для формирования данной последовательности используется банк пар независимых

команд, в котором каждой паре $T_i$ поставлено в соответствие двоичное значение $V_i$, определяемое по псевдослучайному закону. При обнаружении очередного вхождения в код программы одной из имеющихся в банке пар команд ($T_i$) в числовую последовательность добавляется новый элемент. Значение добавляемого элемента $B_j$ определяется порядком следования обнаруженных независимых команд. Если найденные команды следуют в таком же порядке, в котором они расположены в хранящемся в банке шаблоне $T_i$, то элемент принимает значение $V_i$. Иначе – элемент принимает значение равное $!(V_i)$. Одной из характеристик данной последовательности является величина $R(x)$ равная количеству содержащихся в ней монотонных (неубывающих) участков длины $x$. Проведённый анализ IA32 машинного кода показал, что характеристика монотонности машинного кода обладает постоянным свойством, которое заключается в том, что для программных модулей, содержащих более 1000 пар независимых команд, принадлежащих заданному набору, получаемая бинарная последовательность обязательно содержит неубывающие участки с длинами от двух до семи.

В рамках диссертационной работы проанализировано два основных подхода к преобразованию машинного кода с целью модификации вида рассмотренных характеристик. Первый подход заключается в использовании взаимозаменяемых команд или групп команд. Простейший пример модификации кода программы в соответствии с данным подходом показан на рис. 3.

```
add eax, 20        ⇔    sub eax, -20

mov eax, ebx       ⇔    xchg eax, ebx
mov ebx, ecx            mov ebx, ecx

ror eax, 17        ⇔    rol eax, 15

mov eax, 0         ⇔    xor eax, eax
```

Рис. 3. Преобразования кода, основанные на взаимозаменяемых командах

Второй подход заключается в перестановке независимых команд. В качестве примера можно привести следующую последовательность из трёх команд (рис. 4).

```
mov ebx, ecx      mov ebx, ecx      push ecx          mov esi, edi
push ecx    ⇔     mov esi, edi  ⇔   mov ebx, ecx  ⇔   push ecx
mov esi, edi      push ecx          mov esi, edi      mov ebx, ecx
```

Рис. 4. Пример перестановки независимых команд

Результаты проведённого в рамках данной работы анализа применения каждого из предлагаемых подходов для модификации характеристик машинного кода показали, что наиболее эффективным подходом для IA32-архитектуры

является использование механизма перестановок независимых команд, обеспечивающего наибольшую информационную ёмкость.

В **третьей главе** рассмотрен подход к модификации вида статистической характеристики машинного кода, основанный на адаптации метода Patchwork. Предложен ряд конкретных методов внедрения водяного знака за счёт модификации выделенных характеристик ПО. Рассмотрен метод внедрения "отпечатков пальцев" на уровне исходного кода программы, а также метод повышения устойчивости внедряемых водяных знаков к применению известных атак.

Среди методов внедрения информации об авторских правах в растровые изображения одним из наиболее значимых является предложенный Бендером, Грулом, Моримото и др. метод Patchwork. Для размещения водяного знака в данном методе используется скрытая модификация статистической характеристики изображения, что обеспечивает устойчивость внедряемой информации к попыткам обнаружения и уничтожения. Суть подхода, использованного в методе Patchwork, заключается в модификации статистической характеристики не всего защищаемого объекта, а некоторой его части. Преобразуемая часть объекта выбирается как репрезентативное подмножество элементов объекта, определяемое некоторым параметром. При этом характеристика всего объекта остаётся неизменной. Модифицированный вид характеристики выбранного подмножества элементов рассматривается как водяной знак. Обнаружение водяного знака возможно только при известном значении использованного при выборе преобразуемой части объекта параметра, который является секретным ключом.

Для внедрения однобитного водяного знака (признака авторства) в программное обеспечение предложена адаптация подхода, используемого в методе Patchwork, которая заключается в переходе от рассмотрения статистической характеристики изображения к рассмотрению одной из обладающих свойством постоянства статистических характеристик машинного кода.

Пусть $P = [I_1, I_2, \ldots, I_j, \ldots, I_N]$ – множество последовательных команд программы, где $N$ – общее количество команд в программе. Рассмотрим некоторое подмножество команд $L_k \in P$, определяемое как $L_k = [I_{a_1}, I_{a_2}, \ldots, I_{a_i}, \ldots, I_{a_M}]$, где $[a_1, a_2, \ldots, a_i, \ldots, a_M]$ – множество из $M$ позиций команд, формируемое генератором воспроизводимой случайной последовательности с начальным состоянием $k$, таких, что $1 \le a_i \le N$, $M << N$. Пусть $C$ – некоторая статистическая характеристика множества команд, такая, что $\forall k : C(L_k) \approx C(P)$.

Внедрение водяного знака $\omega$ в $P$ заключается в преобразовании подмножества команд $L_{f(\omega)} \in P$, сформированного с использованием значения односторонней функции $f$ от $\omega$ в качестве начального состояния генератора, таким образом, чтобы выполнялось условие $\left|C(P') - C(L_{f(\omega)})\right| > d$, где $P'$ – всё множество команд преобразованной программы; $d$ – некоторое пороговое значение.

В качестве конкретной статистической характеристики предлагается использовать среднее значение индекса команды ($m_x$).

Разработан метод внедрения однобитного признака авторства, основанный на модификации автокорреляционной характеристики последовательности машинных команд программы. В соответствии с предлагаемым методом, программа содержит водяной знак, если значение величины $Sn(k)$, вычисленное при определённом значении сдвига $k=k_0$, существенно отличается от значения этой же величины, вычисленной при значении сдвига равном $k_0+1$.

Внедрение водяного знака выполняется в три этапа.

*Этап 1.* В код программы в заданные позиции помещаются команды *nop* ("no operation"). Позиции вставок формируются и использованием генератора псевдослучайной последовательности, проинициализированного секретным ключом $K$.

*Этап 2.* Код программы, содержащий ранее размещённые команды *nop*, преобразуется таким образом, чтобы значение величины $Sn(k_0)$ для заданного сдвига $k_0$ существенно превышало значение $Sn(k_0+1)$.

*Этап 3.* Производится удаление ранее добавленных в код программы команд *nop*. Выполнение данного этапа позволяет нарушить полученный на втором этапе вид автокорреляционной характеристики и скрыть наличие водяного знака.

Для распознавания водяного знака с использованием секретного ключа $K$ выполняются те же действия, которые выполнялись на первом этапе внедрения. Далее анализируется автокорреляционная характеристика, которая, если использован правильный ключ, будет иметь специфический вид, обусловленный выполнением второго этапа внедрения, что интерпретируется как наличие в программе признака авторства, соответствующего ключу $K$ (рис. 5).

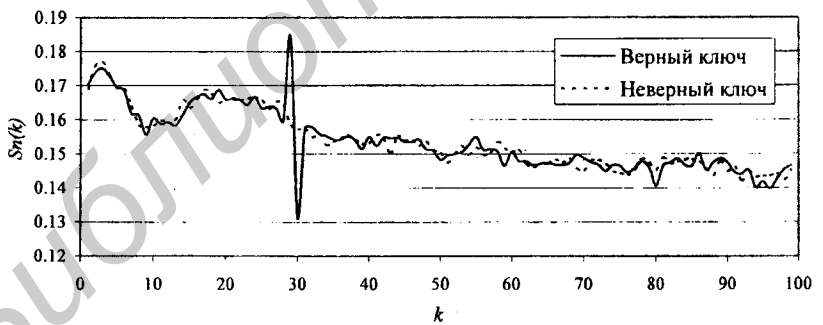


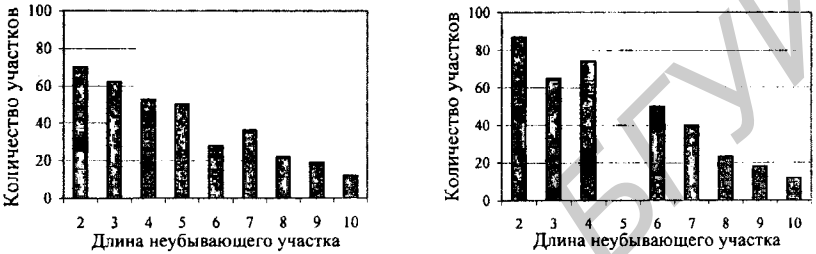

Рис. 5. Кривые функции $Sn(k)$ модифицированной программы

Значение ключа является функцией от данных, идентифицирующих разработчика (владельца) программы.

При попытке распознавания водяного знака с использованием ключа, отличного от того, который был использован при внедрении, вид получаемой зависимости $Sn(k)$ (см. рис. 5) не будет содержать аномалий, позволяющих судить о наличии в программе скрытого признака авторства. Это объясняется сильной

зависимостью автокорреляционной характеристики от взаимного положения команд в программе, которое меняется при выполнении вставок команд *nop*, позиции которых определяются параметром *K*.

Предложен метод внедрения признака авторства в программный модуль, основанный на модификации вида характеристики монотонности машинного кода. Суть данного метода заключается в модификации программного модуля таким образом, чтобы в распределении длин неубывающих участков анализируемой числовой последовательности отсутствовали либо присутствовали в чрезвычайно малом количестве неубывающие участки заданной длины (рис. 6).




а) б)

Рис. 6. Характеристика монотонности числовой последовательности: а) исходная программа; б) модифицированная программа

Последовательность бит, которой соответствует распределение, изображенное на рис. 6 б), несет дополнительную информацию, заключающуюся в отсутствии неубывающих участков длиной пять. Такая аномалия в характеристике монотонности, наличие которой может являться только результатом преднамеренных действий, рассматривается как водяной знак.

Для скрытия вносимой аномалии используются параметризируемые правила интерпретации модифицируемых элементов, представляющих собой пары независимых команд.

Для перехода от ассемблерного кода к бинарной последовательности используются правила интерпретации очерёдности следования независимых команд. Примером является следующее правило:

```
push %re1, %2 @ mov %re3, %re4
```

где “%2” обозначает целочисленную константу, а “%re1”, “%re3” и “%re4” – 32-разрядные регистры. Символ ‘@’ используется в качестве разделителя команд. На основании набора подобных правил формируется список всевозможных шаблонов команд, которые встречаются в правилах.

Каждая пара ($C_1$, $C_2$) последовательных независимых команд интерпретируется в соответствии со следующим выражением:

$$V(C_1,C_2)=\begin{cases}1,(I(C_1)-I(C_2))<0\\0,(I(C_1)-I(C_2))>0\end{cases},$$

где $I(C_1)$, $I(C_2)$ – номера шаблонов, соответствующих командам $C_1$, $C_2$ в упомянутом выше списке. Результат интерпретации независимых команд зависит от порядка расположения шаблонов в данном списке. Перед началом интерпретации независимых пар осуществляется "перемешивание" списка шаблонов с использованием генератора воспроизводимой случайной последовательности с начальным состоянием $K$. При внедрении и распознавании признака авторства используется очерёдность анализа элементов бинарной последовательности от младших адресов к старшим.

Секретный ключ $K$, параметризирующий процесс внедрения водяного знака, является результатом криптографического преобразования данных, идентифицирующих разработчика. Это необходимо для того, чтобы субъект, внедривший водяной знак в программу, имел возможность доказать, что именно он является автором. Доказательством является предъявляемый секретный ключ $K$, позволяющий обнаружить аномалию в одной из характеристик программы.

Для защиты ПО на уровне исходных текстов предложен подход, состоящий в одностороннем эквивалентном преобразовании алгоритма программы. Результатом такого преобразования является снижение читабельности текста программы и затруднение понимания логики её работы. Этим обеспечивается защита преобразованных участков кода от несанкционированной модификации, а также встраивание в исходный код компонентов водяного знака. При этом размещаемые данные характеризуются такой привязкой к логике работы программы, что их удаление или модификация обязательно повлечет за собой неправильное функционирование приложения.

В качестве конкретной реализации односторонних преобразований предлагается использовать замену выражений, вычисляемых в условных конструкциях, на эквивалентные, содержащие проверяемое условие в неявном виде. Допускающие подобную замену выражения присутствуют в любой программе в количестве, достаточном для внедрения водяного знака, содержащего информацию, идентифицирующую разработчика. Одностороннее преобразование условной конструкции производится путём замены стоящего в ней выражения на более сложное. Данное выражение содержит компоненту "водяного знака" и в преобразованном виде константу, на равенство с которой осуществляется проверка (рис. 7).

```
if Var = C then          =>   if E_Var(WM_i) = M then
...                           ...
end if                        end if
```

Рис. 7. Преобразование условной конструкции

Параметр $Var$ представляет собой локальную либо глобальную переменную; параметр $M$ – константу, вычисляемую как:

$$M = E_C(WM_i),$$

где $E$ – некоторая функция; $C$ – константа, изначально присутствовавшая в выражении; $WM_i$ – компонента водяного знака. В соответствии с приведённым

примером, после преобразования функция (в данном случае функция сравнения), вычисляемая в условном операторе, превращается в "черный ящик". Для того чтобы внести изменение в логику работы преобразованного участка кода, а именно модифицировать условие, при котором выражение в скобках будет истинным, необходимо корректно изменить значение параметра *M*. Однако для выполнения данной операции необходимо располагать значением константы *C*, которая после преобразования будет присутствовать в выражении только в неявном виде. Без знания *C* модификация логики преобразованного участка кода будет сложной задачей, поскольку потребует вычисления функции обратной *E*. В качестве функции *E* предлагается использовать целочисленную функцию, являющуюся кандидатом в односторонние.

Достоинством предложенного метода является то, что он позволяет защищать не только ПО, поставляемое конечному пользователю в виде исходных текстов, но и допускает его использование для защиты программ, компилируемых в "native" (машинно-зависимый) код, в комбинации с методами внедрения водяных знаков в ассемблерный код в качестве дополнительного уровня защиты.

Разработан метод защиты информации об авторе, внедряемой на уровне машинного кода, от атаки типа "искажение". Суть данного метода состоит в использовании преобразованных при внедрении компонент водяного знака условных конструкций для защиты кода внутри соответствующих им ветвей программы. Поскольку значение переменной *Var*, используемой в преобразованном условном выражении (см. рис. 7), остаётся неизвестным до тех пор, пока данная ветвь не начала выполняться, его можно использовать как ключ для шифрования ветви. В соответствии с предлагаемым методом, в начало каждой защищаемой ветви помещается вызов процедуры дешифрации её содержимого, которая выполняется перед первым выполнением ветви. Ключом дешифрации является значение переменной *Var*, при котором рассматриваемая ветвь выполняется.

В **четвёртой главе** рассмотрены вопросы реализации предложенных методов для защиты конкретных типов приложений. Представлен программный комплекс «EMWERS» (Extensible Multi-layer Watermark Embedding and Recognition System), реализующий предложенные методы внедрения водяных знаков. Приведены результаты экспериментального исследования эффективности разработанных методов и алгоритмов.

Разработанная реализация предложенных в работе методов внедрения водяных знаков на уровне машинного кода ориентирована на защиту программ, рассчитанных на работу в системах с архитектурой IA32 под управлением операционной системы Windows. Преобразование машинного кода защищаемых программ было реализовано с использованием механизма перестановок независимых команд, который, как показали проведённые экспериментальные исследования, является наиболее эффективным подходом к модификации исполняемого кода бинарных исполняемых модулей IA32-архитектуры. Для реализации локальных преобразований машинного кода был сформирован банк шаблонов пар независимых машинных команд, содержащий 972 элемента.

При реализации метода внедрения водяного знака, использующего характеристику монотонности машинного кода, в качестве параметра, определяющего результат интерпретации очерёдности следования независимых команд, использовано числовое значение, являющееся функцией от текстовых данных, представляющих собой информацию о разработчике, и секретного ключа. Данное числовое значение необходимо для задания начального состояния генератора воспроизводимой случайной последовательности, который используется для равномерного распределения значений '0' и '1' между *t* шаблонами. Для применённого в реализации банка шаблонов с количеством элементов *t* равным 972 количество вариантов интерпретации независимых пар команд составляет $C_{84}^{42} \approx 10^{291}$.

Выполнена реализация механизма защиты водяного знака первого уровня на основе преобразований алгоритма, связанных с размещением водяного знака второго уровня. В разработанной реализации данного механизма используются объектно-ориентированные возможности языка C++, что обеспечивает внесение минимальных модификаций в исходный текст программы.

Предложена реализация защиты фрагментов программ на языке C++ от обратного проектирования и модификации, основанная на одностороннем преобразовании условных конструкций, выполняемом при внедрении водяного знака второго уровня. Объём преобразований исходного текста минимизируется за счёт использования перегружаемых операторов языка C++.

Проведен сравнительный анализ эффективности реализованных методов и существующих реализаций подобных средств защиты. Результаты анализа показывают, что реализация новых методов внедрения водяных знаков по сравнению с лучшим из доступных аналогов обладает рядом преимуществ таких как:

- отсутствие необходимости наличия оригинала программы для извлечения ВЗ;
- наличие возможности применения к скомпилированным в native-код программным модулям;
- повышенная защищённость от атак типа "удаление" и "искажение", что обеспечивается одновременным использованием водяных знаков двух уровней;
- меньшее увеличение размера программного модуля в результате внедрения водяного знака.

Структура разработанной системы «EMWERS» включает три подсистемы, выполняющие соответственно функции внедрения "отпечатка пальцев" на уровне исходного кода (водяной знак второго уровня), внедрения водяного знака на уровне машинного кода (водяной знак первого уровня), обеспечения защиты водяного знака первого уровня при помощи механизма, используемого для размещения водяного знака второго уровня. Схема использования разработанных подсистем в режиме постановки средств защиты показана на рис. 8.


Рис. 8. Схема работы системы «EMWERS» в режиме внедрения водяного знака

Разработанный программный комплекс предназначен для работы в операционных системах Windows 9x/ME, Windows NT/2000/XP.

Программный комплекс «EMWERS» практически использован на предприятии в процессе создания дистрибутива разрабатываемого программного продукта. Схема использования системы «EMWERS» подразумевает хранение данных обо всех поставляемых клиентам копиях программного продукта (дистрибутивах). В рамках данной схемы система «EMWERS» обеспечивает:

- внедрение в исходные тексты компонент программного обеспечения (CPP и ASP файлы) информации о легальном пользователе, количестве лицензий и сроке их действия;
- размещение в бинарных исполняемых файлах компонент программного обеспечения (файлы EXE и DLL) информации о фирме-разработчике;
- защиту перечисленных объектов от обратного проектирования и несанкционированной модификации.

## ЗАКЛЮЧЕНИЕ

Основные результаты диссертационной работы состоят в следующем.

1. Определены характеристики программного обеспечения, допускающие их использование для внедрения информации об авторе (владельце): автокорреляционная характеристика машинного кода, соотношение частот использования машинных команд, характеристика числовой последовательности, получаемой в результате интерпретации очерёдности следования независимых машинных команд [1, 2, 5, 7].

2. Проведенное исследование методов модификации характеристик программных модулей, скомпилированных под архитектуру IA32, показало, что метод, основанный на перестановках независимых команд, является наиболее эффективным с точки зрения объёма скрываемой информации [3, 8, 11].

3. Разработанный метод внедрения водяного знака в программное обеспечение на уровне исходного текста, основанный на одностороннем преобразовании выражений, стоящих в условных конструкциях, является устойчивым к атаке типа "удаление" и "искажение" [14].

4. Разработаны методы внедрения водяного знака (признака авторства) на уровне машинного кода, основанные на преобразовании машинного кода, модифицирующем вид одной из выделенных статистических характеристик программного модуля, обладающие повышенной устойчивостью к обнаружению и применимые для защиты готовых бинарных исполняемых модулей [4, 6, 10, 12, 13].

5. Предложена адаптация метода Patchwork, ранее использованного для внедрения водяного знака в растровые изображения, позволяющая осуществлять скрытую модификацию характеристик машинного кода, выполняемую при размещении признака авторства [2, 9].

6. Предложена схема совместного использования разработанных методов в виде многоуровневой системы защиты, которая обеспечивает повышенную стойкость внедряемой информации и выполняет функцию защиты от обратного проектирования и модификации кода [15]. Предложенная схема реализована в виде программного комплекса, превосходящего лучшие из доступных аналогов по таким характеристикам как: границы применения, защищенность от известных атак и влияние на качество защищаемой программы. Разработанная система практически используется на предприятии для защиты исходных кодов и бинарных исполняемых модулей разрабатываемого программного обеспечения.

## СПИСОК ОПУБЛИКОВАННЫХ РАБОТ ПО ТЕМЕ ДИССЕРТАЦИИ

### *Статьи в научных журналах*

1. Ярмолик В.Н., Портянко С.С. Современные методы и средства защиты авторских прав разработчиков программного обеспечения // Доклады БГУИР.— 2004.— № 1 (5).— С. 126—135.

2. Ярмолик В.Н., Портянко С.С. Статистические свойства исполняемого кода приложений и их использование для защиты авторского права // Вести Института современных знаний.— 2004.— № 18.— С. 62—69.

3. Портянко С.С., Ярмолик В.Н. Внедрение водяного знака в исполняемый код на основе подстановок машинных инструкций // Информатика.— 2005.— № 1(5).— С. 132—138.

4. Портянко С.С., Ярмолик В.Н. Статистический подход для внедрения водяных знаков в исполняемый код // Доклады БГУИР.— 2005.— №1(9).— С. 98—103.

### *Материалы конференций и тезисы докладов*

5. Портянко С.С. Внедрение водяного знака в ПО на основе использования статистических свойств исполняемого кода // Доклады БГУИР: Тез. докл. науч. конф., Минск—Нарочь, 19—23 мая 2003 г. / БГУИР.— Минск, 2003. — Т. 1.— № 2/1.— С. 14—15.

6. Портянко С.С., Ярмолик В.Н. Защита программных модулей от несанкционированного использования при помощи водяных знаков // Известия Белорусской инженерной академии: Мат. докл. науч. конф., Нарочь, 29 сент. — 3 окт. 2003 г. / Бестпринт.— Минск, 2003.— № 1(15)/3.— С . 163—165.

7. Yarmolik V.N., Portyanko S.S. State of the Art in Software Ownership Protection // Computer Information Systems and Industrial Management Applications / Editors: Kh. Saeed, R. Mosdorf.— Bialystok, Poland, 2003.— P. 188—195.

8. Портянко С.С. Внедрение скрытой информации в код программ с использованием механизмов подстановок инструкций // Доклады БГУИР: Тез. докл. науч. конф., Минск—Нарочь, 17—21 мая 2004 г. / БГУИР.— Минск, 2004.— № 5.— С. 29—30.

9. Partsianka S.S. Protection of copyright with the information transmission in telecommunication networks // Известия Белорусской инженерной академии: Мат. докл. науч. конф., Браслав, 2—8 июля 2004 г. / Бестпринт.— Минск, 2004.— № 1(17)2.— С. 154—156.

10. Портянко С.С. Статистический подход к внедрению признака авторства в библиотечные модули // Известия Белорусской инженерной академии: Мат. докл. науч. конф., Нарочь, 27 сент. — 1 окт. 2004 г. / Бестпринт.— Минск, 2004.— № 2(18)/1.— С. 84—86.

11. Портянко С.С., Бавдей А.В. Независимые команды ассемблера как средство встраивания признака авторства в программные модули // Известия Белорусской инженерной академии: Мат. докл. науч. конф., Нарочь, 27 сент. — 1 окт. 2004 г. / Бестпринт.— Минск, 2004.— № 2(18)/1.— С. 83—84.

12. Yarmolik V., Partsianka S. Software IP Protection Based on Watermarking Techniques // Information Processing and Security Systems.— Springer Science+Business Media Inc., 2005.— P. 227—234.

13. Портянко С.С. Внедрение средства защиты авторских прав в аппаратно-программные комплексы на этапах их проектирования и реализации // Современные проблемы радиоэлектроники: Сб. тр. науч. конф., Красноярск, Россия, 5—6 мая 2005 г. / Под ред. А.И. Громыко.— Красноярск: ИПЦ КГТУ, 2005.— С. 478—480.

14. Портянко С.С. Техническое обеспечение защиты авторских прав при распространении программного обеспечения через Internet // Новые информационные технологии в научных исследованиях и в образовании: Тез. докл. науч. конф., Рязань, 20—22 апр. 2005 г. / РГРТА.— Рязань, 2005.— С. 159—160.

15. Портянко С.С. Многоуровневая система защиты компонент программного обеспечения от обратного проектирования и повторного использования // Информационные технологии в XXI веке: Сборник докладов и тезисов III-го Молодежного научно-практического форума, Днепропетровск, 27—28 апр. 2005 г. / УГХТУ.— Днепропетровск, 2005.— С. 163—164.

## РЭЗЮМЭ

**ПАРЦЯНКА Сяргей Сяргеевіч**

## АБАРОНА ПРАГРАМНАГА ЗАБЕСПЯЧЭННЯ ПРЫ ДАПАМОЗЕ ЛІЧБАВЫХ ВАДЗЯНЫХ ЗНАКАЎ

***Ключавыя словы:*** праграмнае забеспячэнне, аутарскія правы, вадзяны знак.

***Аб'ект даследавання*** – праграмнае забеспячэнне. ***Прадмет даследавання*** – метады тэхнічнай абароны аўтарскіх правоў распрацоўшчыкаў праграмнага забеспячэння. ***Мэта працы*** – распрацоўка эфектыўных метадаў абароны праграмнага забеспячэння ад несанкцыянаванага выкарыстання, зваротнага праектавання і мадыфікавання.

***Атрыманыя рэзультаты і іх навізна:*** вызначаны характэрыстыкі праграмнага забеспячэння, дазвалаючыя іх выкарыстанне для ўкаранення інфармацыі аб аўтары; выкарыстаны новы падыход да ўкаранення вадзяных знакаў у праграмнае забеспячэнне, заснаваны на адаптацыі метада ўкаранення вадзянога знака ў растравыя выяўленні (метад Patchwork), характэрызуемага павышанай устойлівасццю ўкараняемай інфармацыі да знаходжання і выдалення; прапанаваны метады ўкаранення прызнака аўтарства ў машынны код, заснаваныя на пераўтварэнні машыннага кода, мадыфікуючым выгляд аўтакарэляцыйнай характэрыстыкі, а таксама пераўтварэнні, мадыфікуючым выгляд характэрыстыкі аднастайнасці машыннага кода; распрацаваны метад укаранення вадзяных знакаў у праграмнае забеспячэнне на ўзроўне зыходнага тэкста, дазваляючы распазнаванне вадзянога знака на ўзроўне машыннага кода і маючы павышаную ўстойлівасць да спроб выдалення і скажэння; прапанавана тэхналогія абароны праграмнага забеспячэння, заснаваная на шматузроўневай сістэме вадзяных знакаў, характэрызуемай павышанай устойлівасццю да вядомых відаў атак, і выконваючай дадатковую функцыю абароны ад зваротнага праектавання і мадыфікавання; распрацаваны праграмны комплекс для ўкаранення ў праграмнае забеспячэнне сродкаў абароны на этапе стварэння і на этапе распаўсюджвання, заснаваны на шматузроўневай сістэме вадзяных знакаў і "адбіткаў пальцаў".

***Выкарыстанне і галіна прымянення:*** распрацаваная сістэма абароны праграмнага забеспячэння выкарыстоўваецца на прадпрыемстве "ЛАППА" для абароны распрацоўваемага праграмнага забеспячэння; атрыманыя ў рамках дысерацыйнай работы вынікі ўкаранены ў навучальны працэс, аб чым маюцца акты ўкаранення.

## РЕЗЮМЕ

**ПОРТЯНКО Сергей Сергеевич**

## ЗАЩИТА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРИ ПОМОЩИ ЦИФРОВЫХ ВОДЯНЫХ ЗНАКОВ

***Ключевые слова:*** программное обеспечение, авторские права, водяной знак. ***Объект исследования*** – программное обеспечение. ***Предмет исследования*** – методы технической защиты авторских прав разработчиков программного обеспечения. ***Цель работы*** – разработка эффективных методов защиты программного обеспечения от несанкционированного использования, обратного проектирования и модификации.

***Полученные результаты и их новизна:*** определены характеристики программного обеспечения, допускающие их использование для внедрения информации об авторе; использован новый подход к внедрению водяных знаков в программное обеспечение, основанный на адаптации метода внедрения водяного знака в растровые изображения (метод Patchwork), характеризуемый повышенной устойчивостью внедряемой информации к обнаружению и удалению; предложены методы внедрения признака авторства в машинный код, основанные на преобразовании машинного кода, модифицирующем вид его автокорреляционной характеристики, а также на преобразовании, модифицирующем вид характеристики монотонности машинного кода; разработан метод внедрения водяных знаков в программное обеспечение на уровне исходного текста, допускающий распознавание водяного знака на уровне машинного кода и обладающий повышенной устойчивостью к попыткам удаления и искажения; предложена технология защиты программного обеспечения, основанная на многоуровневой системе водяных знаков, характеризуемой повышенной устойчивостью к известным видам атак, и выполняющей дополнительную функцию защиты от обратного проектирования и модификации; разработан программный комплекс для внедрения в программное обеспечение средств защиты на этапе создания и на этапе распространения, основанный на многоуровневой системе водяных знаков и “отпечатков пальцев”.

***Использование и область применения:*** разработанная система защиты программного обеспечения используется на предприятии “ЛАППА” для защиты разрабатываемого программного обеспечения; полученные в рамках диссертационной работы результаты внедрены в учебный процесс, о чём имеются акты внедрения.

## SUMMARY

**PARTSIANKA Siarhei Siarheevich**

## SOFTWARE PROTECTION BY MEANS OF DIGITAL WATERMARKS

***Key words:*** software, authorship rights, watermark.

***Object of research*** is software. ***Subject of research*** is methods of technical protection of software developer authorship rights.

***The purpose of work*** – development of effective methods of software protection against unauthorized use, reverse engineering and modification.

***Received results and their novelty:*** the software characteristics allowing their use for authorship information embedding have been determined; the new approach to software watermarking based on adaptation of the raster image watermarking method (Patchwork method), which is characterized by enhanced persistence of embedded information to disclosure and removal is employed; the method of authorship sign embedding into machine code based on machine code transformation, which modifies the view of its autocorrelation characteristic, and transformation, which modifies the characteristic of machine code monotony is proposed; software watermarking method on the level of source code, which allows watermark recognition on the level of machine code and has enhanced persistence to attempts of removing and distorting is developed; the software protection technology based on multi-level watermarks system, which is characterized by enhanced persistence to known kinds of attacks and provides additional functionality – defense against reverse-engineering and modification is proposed; the system for embedding the aids of defense in software on implementation and distribution phases based on multi-layer system of watermarks and fingerprints is developed.

***Utilization and field of application:*** developed software defense system is used to protect the software, which is produced by "LAPPA" enterprise; the results of research are being applied in educational process that is documented by corresponding utilization acts.

ПОРТЯНКО Сергей Сергеевич

# ЗАЩИТА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРИ ПОМОЩИ ЦИФРОВЫХ ВОДЯНЫХ ЗНАКОВ

Специальность 05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук



| | | |
|---|---|---|
| Подписано в печать 05.07.2006. | Формат 60x84 1/16. | Бумага офсетная. |
| Гарнитура Times. | Печать ризографическая. | Усл. печ. л. 1,63. |
| Уч.-изд. л. 1,4. | Тираж 60 экз. | Заказ 468. |

Издатель и полиграфическое исполнение: Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
ЛИ №02330/0056964 от 01.04.2004. ЛП №02330/0131666 от 30.04.2004.
220013, Минск, П. Бровки, 6.