На правах рукописи

Шариков Павел Иванович

РАЗРАБОТКА СТРАТИФИЦИРОВАННЫХ МЕТОДИК СОЗДАНИЯ И ВЛОЖЕНИЯ УСТОЙЧИВОГО К АТАКАМ ДЕКОМПИЛЯЦИЕЙ И ОБФУСКАЦИЕЙ ЦИФРОВОГО ВОДЯНОГО ЗНАКА В БАЙТ-КОД CLASS-ФАЙЛОВ JAVA-ПРИЛОЖЕНИЙ И ИНФОРМАЦИОННЫХ СИСТЕМ

2.3.6. Методы и системы защиты информации, информационная безопасность

Автореферат
диссертации на соискание ученой степени
кандидата технических наук

Санкт-Петербург – 2023

Работа выполнена в федеральном государственном бюджетном образовательном учреждении высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича» на кафедре защищенных систем связи.

Научный руководитель: кандидат технических наук, доцент
Красов Андрей Владимирович

Официальные оппоненты:
Лаута Олег Сергеевич,
доктор технических наук, доцент,
Государственный университет морского и речного флота имени адмирала С.О. Макарова»,
кафедра комплексного обеспечения информационной безопасности, профессор кафедры

Жуковский Евгений Владимирович,
кандидат технических наук, доцент,
Санкт-Петербургский политехнический университет Петра Великого, Высшая школа кибербезопасности,
доцент

Ведущая организация: Федеральное государственное бюджетное учреждение науки «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук»,
г. Санкт-Петербург

Защита состоится 06 марта 2024 года в 14.00 на заседании объединенного диссертационного совета 99.2.038.03, созданного на базе Федерального государственного бюджетного образовательного учреждения высшего образования «Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова», Федерального государственного автономного образовательного учреждения высшего образования «Санкт-Петербургский государственный университет аэрокосмического приборостроения», Федерального государственного бюджетного образовательного учреждения высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича» по адресу: Санкт-Петербург, пр. Большевиков, д. 22, корп. 1, ауд. 554/1.

С диссертацией можно ознакомиться в библиотеке СПбГУТ по адресу Санкт-Петербург, пр. Большевиков, д. 22, корп. 1 и на сайте www.sut.ru.

Автореферат разослан 29 декабря 2023 года.

Ученый секретарь
диссертационного совета 99.2.038.03,
канд. техн. наук, доцент А.Г. Владыко

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования. На текущий момент, самый популярный язык программирования – Java, по данным множества Интернет-ресурсов, которые специализируются на отслеживании количества кода того или иного языка программирования, величины сообщества в интернете и на специализированных ресурсах. Данный язык используются в самых разных сферах, начиная от касс в супермаркетах и заканчивая банковской сферой. Следствием этого является повышенный интерес недобросовестных пользователей и злоумышленников к java-приложениям, частям различных проектов или информационных систем, которые написаны на Java или работают на JVM.

Class-файлы легко анализируются. При должной подготовке, специалисту удастся получить исходный код программы на Java, что позволит использовать его в своих целях. Возможность получить исходный код программы, написанной на Java, является задокументированным и правовым свойством языка и виртуальной машины Java. Отказаться от использования данного языка программирования, значит максимально сократить круг поиска специалистов на проекты, что экономически не выгодно. Также, существует дополнительный ряд факторов, сдерживающих от прекращения использования языка программирования Java. Например, репутация языка у разработчиков, самое большое и профессионально подготовленное сообщество, другими словами, язык Java – экономически целесообразный выбор компаний.

У компаний и отдельных разработчиков возникает потребность защитить свои программные продукты в огромном потоке данных от нарушения интеллектуальной собственности, атак, направленных на нарушение работоспособности, целостности. Существуют различные варианты, но в данной работе рассматривается вариант цифровых водяных знаков, которые должны быть легко идентифицируемы для юридического обоснования прав на программу компанией-разработчиком и сложно поддающиеся уничтожению.

Научная деятельность, связанная с проверкой работоспособности исполняемого class-файла java-приложения, с вложением цифрового водяного знака в class-файлы на разных аппаратных средах при разных условиях и конфигурациях, с защитой скрытого вложения в class-файл является исследованием динамического характера. Развитие языка Java, как и виртуальной машины Java, не стоит на месте, а активно улучшается и дорабатывается.

Изучением вопросов скрытого вложения цифровых водяных знаков в class-файлы и их защитой занимались многие ученые, в том числе: Котенко И.В., Красов А.В., Штеренберг С.И., J. Palsberg, A. Monden, Seiji Umatani, Krishan Kumar, Hamilton James, Gupta G., Pieprzyk J и др.

В более широком смысле, применимым не только к class-файлам, но и к изображениям, мультимедиа, изучением цифровой стеганографии занимались такие

ученые, как: Коржик В.И., Яковлев В.А., Оков И.Н., Грибунин В.Г., Туринцев И.В., Нечта И.В.

Произведен анализ и исследование работ заграничных коллег на данную тематику, либо косвенно относящиеся к ней.

Как, например, исследование J. Palsberg, S. Krishnaswamy, K. Minseok, D. Ma, Q. Shao, Y. Zhang. Experience with Software Watermarking. Авторы, рассматривают возможность расположения водяного знака в структуре данных, которые появились во время выполнения программы. Для вложения используется просто число, которое может быть представлено, как граф, который построен как структура объекта Java программы. Авторы производят вложение цифрового водяного знака в такие данные как: изменение размера кода, время выполнения программы, использование пространства «кучи». Работа косвенно относится к текущему исследованию. В связи с тем, что в ней предлагается только устаревшая, на текущий момент времени, методика для вложения цифрового водяного знака в java-приложение.

Авторы A. Monden, H. Iida, K. Matsumoto, K. Inoue, K. Torii, «A practical method for watermarking Java programs», представляют методику скрытого вложения информации в байт-код class-файла собственной разработки. Основой предлагаемой методики является замена эквивалентных операционных кодов байт-кода в скомпилированном class-файле Java. Таким образом, просматривается весь байт-код class-файла, далее согласно спецификации JVM, эквивалентные опкоды заменяются друг другом, образуя другую последовательность бит. Перебирая данные последовательности бит и изменяя только необходимые опкоды, авторы данной статьи предлагают производить вложение ЦВЗ в байт-код class-файла.

Авторы Seiji Umatani, Tomoharu Ugawa, Masahiro Yasugi в работе «Design and Implementation of a Java Bytecode Manipulation Library for Clojure», исследуют различные программы, которые могут анализировать и изменять байт-код Java. Большинство из них работают с class-файлами так же, как и DOM (document object model) HTML или XML. Они генерируют дерево, которое отражает структуру формата class file. Авторы рассказывают про собственную разработку BcMacro, главным отличием которого является то, что дерево шаблонов является первым классом объекта Clojure.

Авторы Minyoung Sung, Soyoung Kim, Sangsoo Park, Naehyuck Chang, Heonshik Shin в своей работе «Comparative performance evaluation of Java threads for embedded applications: Linux Thread vs. Green Thread» производят сравнение скорости работы «green threads» (потоки выполнения, управление которыми вместо операционной системы выполняет виртуальная машина) и потоков операционной системы Linux в Java приложениях. Во время исследования запускаются несколько тестовых задач, которые используют многопоточность, замеряется время синхронизации потоков и время выполнения программ.

Авторы Krishan Kumar, Viney Kehar, Prabhpreet Kaur Tulsi провели исследование «An Evaluation of Dynamic Java Bytecode Software Watermarking Algorithms».

В своем исследовании авторы, произвели вложение цифрового водяного знака, используя динамические алгоритмы, в 35 class-фалов. Далее проверили реакцию файлов на атаки. Использовали 2 алгоритма: алгоритм Арбойта (используются ранг утверждений (предикатов) в библиотеке, у каждого есть своё значение (число), которое используется как ЦВЗ) и алгоритм Коллберга-Томпсона (используется число, которое представлено в виде графа структуры объекта во время выполнения).

Авторы Hamilton James, Sebastian Danicic, «An Evaluation of the Resilience of Static Java Bytecode Watermarks Against Distortive Attacks». В данном исследовании поместили цифровой водяной знак в 60 файлов разными алгоритмами вложения. Провели ряд атак на данные файлы. Использовали 14 различных статичных алгоритма вложения на 3 разных системах. В итоге было применено 37 разных методик атаки изменения программы. Все используемые jar-файлы являются плагинами для текстового редактора jEdit, так как данные файлы минимального размера, но передают весь спектр java-программ. Результаты: всего 840 фалов, успешно цифровой водяной знак вложен в 671. Из 671 файла с цифровым водяным знаком, до атаки удалось успешно обнаружить 588 вложения. К каждому файлу применили все методы атак. В сумме вышло 26169 атак.

Авторы Gupta G., Pieprzyk J. в своей работе «Source code watermarking based on function dependency oriented sequencing» рассматривают алгоритм вложения цифрового водяного знака в файлы C/C++. Идея алгоритма цифрового водяного знака заключается в том, чтобы модифицировать последовательность так, чтобы функции этой последовательности были представлены в виде последовательно зашифрованного цифрового водяного знака.

Также, существует несколько отечественных работ, связанных с проверкой быстродействия JVM на различных платформах или с различными ОС.

Первая подобная работа за авторством Хашковского В. В., Лутай В.Н., Юрченко В.В. Сравнительная оценка времени выполнения программ на различных платформах была проведена в 2009 году. В ней авторы сравнивают скорость выполнения программы в Windows XP и Linux Open SUSE при использовании одинаковых аппаратных платформ. Исследователи используют возможности JVM по подсчету времени исполнения программы.

Вторая работа украинских коллег Дидух А. И., Тищенко В. В. Сравнение быстродействия Java на микрокомпьютере Raspberry Pi проведена уже в 2015 году и сравнивает быстродействие программ на Java при изменении частоты процессора.

На текущий момент большинство работ устарело, в следствии того, что внутренние алгоритмы работы Java и виртуальной машины Java претерпели значительные изменения и оптимизации. Также, во время исследования некоторых работ были выявлены недостатки методик, значительное снижение эффективности,

следствием чего было, также изменение внутренних алгоритмов работы виртуальной машины Java.

К основным вопросам, рассматриваемым в работе, необходимо отнести: разработка методики создания и эффективного вложения цифрового водяного знака в байт-код class-файла, учитывающего различные версии виртуальной машины Java, различные версии платформ; разработка методики скрытого вложения цифрового водяного знака в байт-код class-файла java-приложения, с исследованием возможности защиты вложения от атак на предмет разрушения цифрового водяного знака посредством декомпиляции; разработка методики создания и вложения цифрового водяного знака в class-файлы информационной системы, устойчивого к атакам обфускацией; анализ разрушающих факторов различных типов атак на цифровой водяной знак вложенный различными методиками в байт-код class-файла и способы их предотвращения.

Цель работы и задачи исследования. *Цель* работы состоит в разработке методик, позволяющих наиболее эффективно произвести создание и вложение цифрового водяного знака увеличенного объема в байт-код class-файлов java-приложений и информационных систем на Java, с увеличенной устойчивостью к различного рода атакам, направленных на разрушение или повреждение цифрового водяного знака, в интересах защиты авторских прав на исходный код.

Для достижения поставленной цели необходимо решить следующие *задачи*:

1. Разработка методики создания и вложения цифрового водяного знака увеличенного объема в байт-код class-файла. Исследование произведенного вложения, средний объем информации возможной к вложению в class-файл от его объема в процентах, сравнение с другими методиками.

2. Разработка методики создания и вложения цифрового водяного знака в class-файлы java-приложения устойчивого к атакам декомпиляцией направленных на его разрушение. Исследование устойчивости произведенного вложения к атакам декомпиляцией.

3. Разработка методики создания и вложения цифрового водяного знака в class-файлы информационной системы устойчивого к атакам обфускацией направленных на его разрушение. Исследование устойчивости произведенного вложения к атакам обфускацией.

Научная новизна результатов исследования заключается в разработке новых более эффективных методик создания и вложения цифровых водяных знаков большего объема и устойчивых к атакам в class-файлы java-приложений и информационных систем. Разработанные методики:

1. Разработана методика создания и вложения цифрового водяного знака в байт-код class-файлов, основанная на архитектурных особенностях языка Java и виртуальной машины Java.

2. В отличие от известных методик, отличительной особенностью данной методики является то, что:

– Создание цифрового водяного знака происходит на основе анализа структуры байт-кода class-файла. Таким образом, значительно повышается невозможность изъятия цифрового водяного знака из покрывающего сообщения.

– Возможно создание и вложение цифрового водяного знака по объему превосходящего другие ЦВЗ в исполняемые файлы, за счет использования дополнительных наборов операционных команд в байт-коде и их связи.

3. Разработана методика создания и вложения цифрового водяного знака в байт-код class-файлов java-приложения, устойчивого к атакам декомпиляцией. Разработанная методика, в отличие от известных позволяет:

– Произвести создание цифрового водяного знака на основе анализа структуры байт-кода class-файлов java-приложения. Поиска наименьшего возможного к созданию цифрового водяного знака, на основе анализа всех class-файлов java-приложения, для произведения тиражируемого вложения во все class-файлы java-приложения. Таким образом, можно произвести тиражируемое вложение цифрового водяного знака в различном объеме во все class-файлы java-приложения.

– Создать и произвести тиражируемое вложение цифрового водяного знака в каждый class-файл java-приложения, который устойчив к атакам направленной перекомпиляции, как отдельных class-файлов, так и полноценного java-приложения.

4. Разработана методика, позволяющая произвести создание и вложение цифрового водяного знака, устойчивого к атакам обфускацией, в байт-код class-файлов информационной системы. Особенности разработанной методики:

– Впервые инструменты обфускации рассматриваются не как способ оптимизации исходного кода class-файла и его защиты от разбора и кражи логики, а как инструмент для повреждения и уничтожения вложенного в class-файл цифрового водяного знака.

– Создать и произвести тиражируемое вложение цифрового водяного знака в каждый class-файл информационной системы, который устойчив к атакам обфускацией, как отдельных class-файлов, так и полноценного java-приложения.

– Создать и произвести вложение цифрового водяного знака регистратора в class-файлы с ключевой логикой, каждого java-приложения информационной системы, который позволяет производить проверку целостности информационной системы.

Теоретическая и практическая значимости. *Теоретическая значимость* результатов работы заключается ее вкладом в развитие теории методик создания и вложения цифрового водяного знака в исполняемые файлы, а именно: в доказательстве возможности использования расширенного набора операционных команд байт-кода для создания и вложения цифрового водяного знака в class-файлы посредством эквивалентных замен; в установлении возможности вложения устойчивого к атакам декомпиляцией цифрового водяного знака в class-файлы java-приложения; в

доказательстве устойчивости цифрового водяного знака к атакам декомпиляцией; в установлении возможности вложения устойчивого к атакам обфускацией цифрового водяного знака в class-файлы информационной системы; в доказательстве устойчивости цифрового водяного знака к атакам.

Практическая значимость работы заключается в следующих результатах:

1. Предложенная методика создания и скрытого вложения цифрового водяного знака в байт-код class-файла на основе недекларированных возможностей виртуальной машины Java заключается в использовании расширенного набора операционных команд виртуальной машины Java для эквивалентных замен в исполняемых class-файлах с целью вложения цифрового водяного знака увеличенного объема равному 0,2-5,7% от общего объема исполняемого class-файла, в отличие от существующих методик позволяющих произвести вложение цифрового водяного знака объема равного 0,2-1,5% от общего объема исполняемого class-файла; методика доведена до реализации алгоритма программы для ЭВМ;

2. Предложенная методика создания и вложения цифрового водяного знака в class-файлы java-приложения устойчивого к атакам декомпиляцией направленных на его разрушение заключается в повышенной устойчивости цифрового водяного знака к атакам декомпиляцией; эффективность методики достигает 95%, что на 50-55% больше по сравнению с существующими методиками;

3. Предложенная методика создания и вложения цифрового водяного знака в class-файлы информационной системы устойчивого к атакам обфускацией направленных на его разрушение заключается в возможности вложения цифрового водяного знака распределенного по частям информационной системы обладающего устойчивостью к атакам обфускацией; вероятность устойчивости цифрового водяного знака к атакам обфускацией достигает 76%, что на 66% больше по сравнению с существующими методиками; методика доведена до реализации алгоритма программы для ЭВМ.

Полученные результаты подтверждаются тестовыми расчетами и проведенными экспериментами. Также, результаты данной работы использовались в НИР «Построение доверенной вычислительной распределенной среды», заказчик ООО Ланк 2018-19 гг., №10200 ИИ28037, кафедры Защищенных Систем Связи под научным руководством А.В. Красова.

Методы исследования. При решении поставленных задач использовались методы цифровой стеганографии, информационной безопасности в программных средах, теории управления, современные методы и средства разработки программных средств, теории графов, теории алгоритмов.

Положения, выносимые на защиту

1. Методика создания и скрытого вложения цифрового водяного знака в байт-код class-файла на основе недекларированных возможностей виртуальной машины Java.

2. Методика создания и вложения цифрового водяного знака в class-файлы java-приложения устойчивого к атакам декомпиляцией направленных на его разрушение.

3. Методика создания и вложения цифрового водяного знака в class-файлы информационной системы устойчивого к атакам обфускацией направленных на его разрушение.

Степень достоверности и апробация результатов. *Достоверность* результатов, выносимых на защиту диссертационного исследования, выводов научного характера подтверждаются системным подходом к решению поставленных задач, обоснованием выбранных контейнеров, операционных кодов языка программирования Java, функций языка программирования для затруднения декомпиляции, оценке эффективности декомпиляторов и обфускаторов доказательствами и результатами экспериментальной проверки разработанных методик, анализом работ существующих зарубежных и отечественных практик решения аналогичных задач, апробацией результатов работы на международных и российских конференциях, а также подтверждением о внедрении предложенных методик в организациях и предприятиях.

Апробация результатов исследования. Основные положения работы докладывались и обсуждались на международных и республиканских конференциях, в том числе: II всероссийская научно-техническая конференция «Теоретические и прикладные проблемы развития и совершенствования автоматизированных систем управления военного назначения»; Региональная научно-техническая конференция студентов, аспирантов и молодых ученых «Студенческая весна» (Санкт-Петербург, 2016, 2017); Международный конгресс (CTN-2016) «Коммуникационные технологии сети»; Международная научно-техническая и научно-методическая конференция «Актуальные проблемы инфотелекоммуникаций в науке и образовании» (Санкт-Петербург, 2018, 2019); VI Всероссийская научно-техническая конференция «Проблема комплексного обеспечения информационной безопасности и совершенствование образовательных технологий подготовки специалистов силовых структур» – КОНФИБ; 13th International Symposium on Intelligent Distributed Computing IDC 2019; 2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT) 5-7 Oct. 2020; The 13th International Congress' workshop on Ultra Modern Telecommunications and Control Systems / The 7th young researchers international conference on the internet of things and its enablers "Artificial Intelligence in Communication Networks" 25-27 October, 2021.

Внедрение в учебный процесс по дисциплине «Программно-аппаратные средства защиты информации» (рабочая программа дисциплины, регистрационный № 22.05/381-Д) для старших курсов обучения бакалавров по направлению подготовки 10.03.01 «Информационная безопасность». Подготовлено учебно-методическое пособие «Разработка защищенных приложений» (Россия, Санкт-Петербург, СПбГУТ).

Публикации. По теме диссертации опубликовано 23 печатных работ, 8 из которых – в изданиях из перечня рецензируемых научных журналов ВАК при Минобрнауки России, в том числе 4 из них без соавторства; 4 – в международных изданиях, индексируемых в базах данных Web of Science и Scopus; 1 – монография; 3 свидетельства о государственной регистрации программы для ЭВМ; 7 работ, опубликованных в других изданиях.

Соответствие паспорту специальности. Все результаты, выносимые на защиту, соответствуют п.п. 7 и 17 паспорта научной специальности 2.3.6. Методы и системы защиты информации, информационная безопасность.

Личный вклад автора. В работе предложены методика создания и скрытого вложения увеличенного объема цифрового водяного знака в байт-код class-файла, методика создания и вложения цифрового водяного знака в class-файлы java-приложения устойчивого к атакам декомпиляцией направленных на его разрушение, методика создания и вложения цифрового водяного знака в class-файлы информационной системы устойчивого к атакам обфускацией, разработаны программы для ЭВМ, реализующие предложенные методики. В том опубликовано 4 печатные работы в изданиях из перечня рецензируемых научных журналов ВАК при Минобрнауки России без соавторства. Перечисленные результаты получены автором лично.

Объем и структура диссертации. Диссертационная работа состоит из введения, 4 глав, заключения, списка литературы и 5 приложений. Материал изложен на 248 страницах, включает 15 таблиц, 42 рисунков и схем. Список литературы содержит 172 наименования.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении проведен анализ проблемы вложения цифрового водяного знака в байт-код class-файла с возможностью устойчивости к атакам. Обоснована актуальность работы. Сформулированы цель и задачи работы, приведены основные научные и практические результаты.

В первой главе рассмотрена компьютерная стеганография в целом, рассмотрены варианты, когда использование цифрового водяного знака необходимо, сделан обзор возможных контейнеров для цифрового водяного знака, рассмотрены способы вложения информации в исполняемые файлы, произведен и обоснован выбор типа контейнера для текущего исследования.

Также, рассмотрены современные способы отбора контейнеров для скрытого вложения, рассмотрены методы суррогатной, селективной и конструирующей стеганографии. На основе имеющихся данных была произведена корректировка способов отбора контейнера для вложения информации в исполняемые class-файлы

текущего исследования для отбора контейнеров и произведения скрытого вложения в них.

Произведен подробный разбор основных свойств выбранного языка программирования и его окружения, рассмотрены принципы работы виртуальной машины Java, рассмотрены основные команды виртуальной машины Java и наиболее значимая для использования часть ее спецификации и команд. Произведен обзор структуры исполняемых файлов языка программирования Java и анализ их байт-кода.

Приведены примеры исполнения байт-кода виртуальной машиной Java и указаны места, где возможно редактирование байт-кода программы без ущерба ее работоспособности, для вложения цифрового водяного знака.

Исследована безопасность языка программирования и его виртуальной машины Java. Рассмотрены принципы работы верификатора байт-код виртуальной машины Java. Рассмотрены существующие методики вложения цифрового водяного знака в исполняемые файлы Java.

Во второй главе разработана методика создания и скрытого вложения цифрового водяного знака увеличенного объема в байт-код class-файла основанная на использовании дополнительных наборов операционных кодов байт-кода. Методика позволяет производить скрытое вложение цифрового водяного знака увеличенного объема в байт-код class-файлов на 60-80% по сравнению с известными. Методика доведена до реализации алгоритма программы для ЭВМ.

Преимущества методики:

- Позволяет производить большого объема вложение в исполняемые class-файлы
- Не нарушается логика работы исполняемых файлов
- Не увеличивается объем class-файлов
- Возможность вложения, тиражируемого цифрового водяного знака
- Возможность автоматизации процесса

Для полноценного разбора, как устроен выбранный язык программирования, и как устроена экосистема его компонентов, составлено формализованное множественное описание компонентов, принципов их работы и взаимосвязей друг с другом в теоретико-множественную модель языка программирования Java, продемонстрированную на рисунке 1.

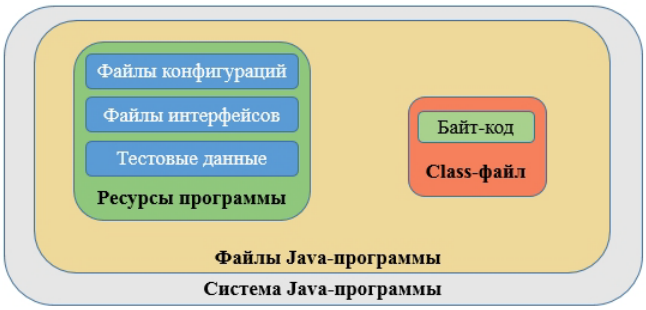

Рисунок 1 – Теоретико-множественная модель

В модели стеговложения java-приложения рассматриваются, как машины Тьюринга ТМ. Следовательно, при реализации стеговложения для семейства ТМ M необходима такая PPT ST, которая удовлетворяет следующим условиям:

1. **Функциональная эквивалентность.** Для любой ТМ q, q∈M, в результате стеговложения ST(q) строится ТМ, эквивалентная исходной машине q.

2. **Полиномиальные издержки.** Для любой ТМ q, q∈M, размер и быстродействие любой ТМ ST(q) отличается от размера и быстродействия ТМ q не более чем полиномиально, т.е. | ST(q)|= poly(|q|) и time(ST(q (x))) = poly(time(q (x))).

Эффективная неотличимость программ. Для любой пары эквивалентных ТМ q_1 и q_2 из класса M, имеющих одинаковый размер, распределения вероятностей случайных величин $ST(q_1)$ и $ST(q_2)$ вычислительно неотличимы, для любой PPT D справедливо соотношение $|Pr[D(ST(q_1)) =1] – Pr[D(ST(q_2)) =1]| = ≤ neg(|q|)$.

Рассмотрены существующие методики отбора контейнера для вложения информации в исполняемые class-файлы, сделаны выводы о необходимости применения комплексной методики на основе существующих.

По результатам анализа был определен расширенный набор эквивалентных опкодов байт-кода языка программирования Java, доступный для эквивалентных замен с целью вложения цифрового водяного знака. Фрагмент базы правил для эквивалентных замен опкодов приведен в таблице 1.

Были определены методики для сравнения class-файлов с цифровым водяным знаком и без. Для того, чтобы определить изменяет ли цифровой водяной знак такие параметры class-файла как запуск, логика, операторы и другие. Число опкодов для эквивалентных замен в class-файле высчитывается по формуле:

$$\log_2(\sum_{i=1}^{n} O_i!).$$

Для оценки относительной сложности class-файла используется метрика Джилба:

$$c_l = \frac{C_L}{N},$$

где C_L – абсолютная сложность java-приложения, характеризующуюся количеством циклов, операторов условных и безусловных переходов в нем, а N – общее количество операторов программного кода.

Для оценки информационная прочность class-файла используется метрика Чепина:

$$Q = P + 2M + 3C + 0{,}5T.$$

Таблица 1 – Фрагмент базы правил методики для эквивалентных замен опкодов в class-файлах

Мнемоническое представление опкода	Байт-код (HEX)	Байт-код (BINARY)	Возможные эквивалентые опкоды (мнемоника)
ifeq	0x99	10011001	ifne
ifne	0x9a	10011010	ifeq
iflt	0x9b	10011011	ifle, ifgt, ifge
ifle	0x9c	10011100	iflt, ifgt, ifge
ifgt	0x9d	10011101	iflt, ifle, ifge
ifge	0x9e	10011110	iflt, ifle, ifgt
ifnull	0xc6	11000110	ifnonnull
ifnonnull	0xc7	11000111	ifnull
if_icmpeq	0x9f	10011111	if_icmpne
if_icmpne	0xa0	10100000	icmpeq
if_icmplt	0xa1	10100001	if_icmple, if_icmpgt, if_icmpge
if_icmple	0xa4	10100100	if_icmplt, if_icmpgt, if_icmpge
if_icmpgt	0xa3	10100011	if_icmplt, if_icmple, if_icmpge
if_icmpge	0xa2	10100010	if_icmplt, if_icmple, if_icmpgt
if_acmpeq	0xa5	10100101	if_acmpne
if_acmpne	0xa6	10100110	if_acmpeq
short	-	-	int, long
int	-	-	short, long
long	-	-	short, int
float	-	-	double
double	-	-	float
iadd	0x60	1100000	ladd
ladd	0x61	1100001	iadd
fadd	0x62	1100010	dadd
dadd	0x63	1100011	fadd
iload	0x1a	11010	lload
lload	0x16	10110	iload
fload	0x17	10111	dload
dload	0x18	11000	fload
istore	0x36	110110	lstore
lstore	0x37	110111	istore
fstore	0x38	111000	dstore
dstore	0x39	111001	fstore

Также, на основе методики разработан алгоритм, продемонстрированный на рисунках 2-3, доведенный до реализации программы.

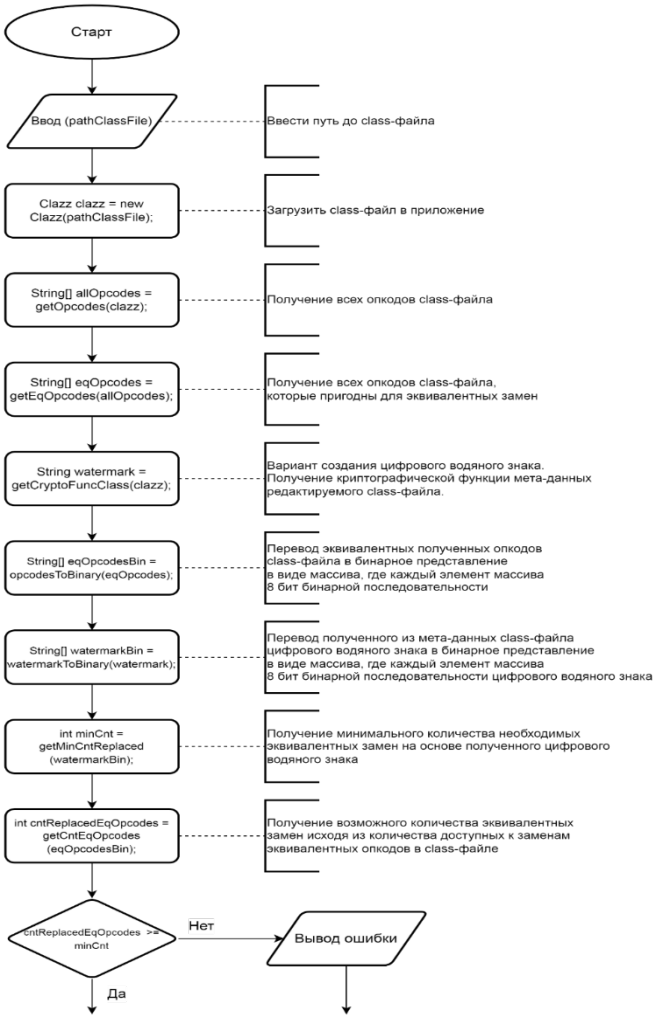

Рисунок 2 – Пример блок-схемы алгоритма создания и вложения цифрового водяного знака в class-файл. Часть 1

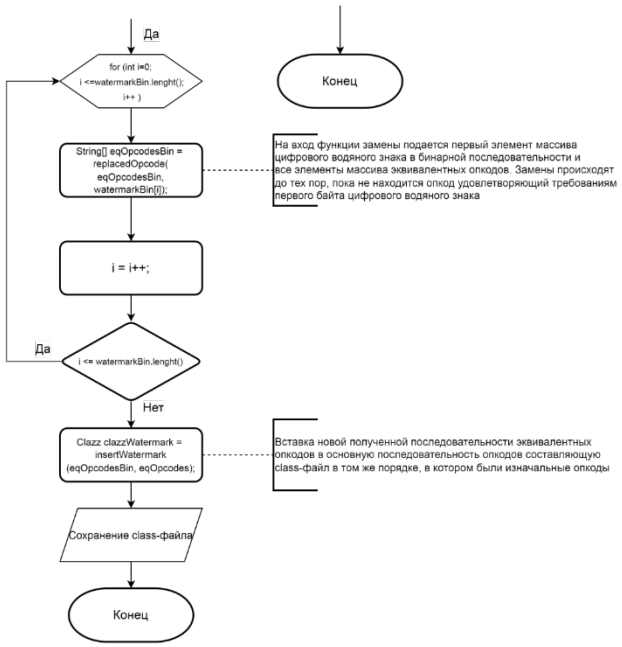

Рисунок 3 – Пример блок-схемы алгоритма создания и вложения цифрового водяного знака в class-файл. Часть 2

Проведен эксперимент создания и вложения цифрового водяного знака в простой class-файл с минимальным количеством логики и операционных команд пригодных для эквивалентных замен. Результаты показали, что объем успешно вложенного цифрового водяного знака в class-файл составил 0,4% от общего объема class-файла, что на 30% больше по сравнению с известными методиками и позволяет доказать выполнение поставленной в настоящем диссертационном исследовании задачи. Эффективность методики в сравнении с существующими продемонстрирована в таблице 2.

Таблица 2 – Эффективность методики вложения ЦВЗ увеличенного объема в class-файлы

Показатель	Известные методики	Разработанная методика
Предельное количество используемых групп опкодов	1 – 2	4 – 5
Средний объем вложения от общего объема class-файла, %	0,2 – 1,5	0,2 – 5,7
Зависимость типа информации от методики	Сильная зависимость, необходимость добавления блоков кода в исходный код	Минимальная зависимость, ограниченная только количеством кодировок для представления данных

В третьей главе разработана методика создания и скрытого вложения цифрового водяного знака, устойчивого к атакам декомпиляцией, в байт-код class-файлов java-приложений. Методика основана на использовании результатов анализа class-файлов java-приложения, выявления class-файлов с ключевой логикой для использования шаблонов и методов проектирования кода, затрудняющих декомпиляцию class-файла, с последующим вложением цифрового водяного знака. Методика в отличие от известных позволяет произвести вложение цифрового водяного знака устойчивого к атакам перекомпиляцией, в class-файлы java-приложения в 80-90% случаев. Методика доведена до реализации алгоритма программы для ЭВМ.

Преимущества методики:

- Учитывает декомпиляцию и повторную компиляцию class-файлов, как инструмент атаки на цифровой водяной знак
- Позволяет производить вложение функций, затрудняющих декомпиляцию, как на этапе написания исходного кода, так и во время работы с скомпилированным class-файлом
- Использует расширенный набор опкодов для эквивалентных замен в байт-коде class-файлов
- Возможность автоматизации процесса

Разработанная методика охватывает весь цикл разработки программного обеспечения в виде java-приложения и представляет собой ряд шагов,

предпринимаемых на различных этапах разработки java-приложения. Разработанная методика состоит из трех этапов, продемонстрированных на рисунках 4-7.

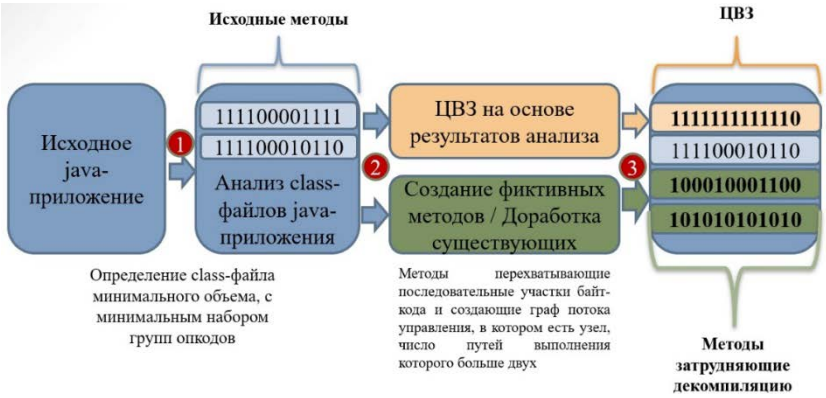

Рисунок 4 – Методика создания и вложения ЦВЗ в class-файлы java-приложения

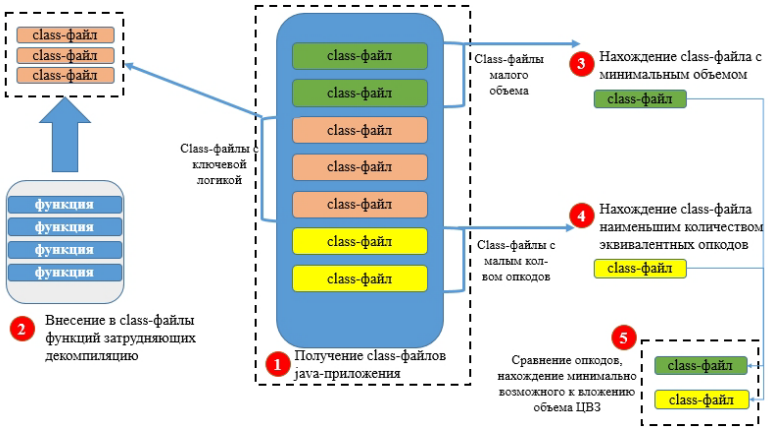

Рисунок 5 – Подготовительный этап

Подготовительный этап нацелен на анализ class-файлов java-приложения, разделение их по категориям, выявление наименьшего доступного объема во всем java-приложении для вложения цифрового водяного знака.

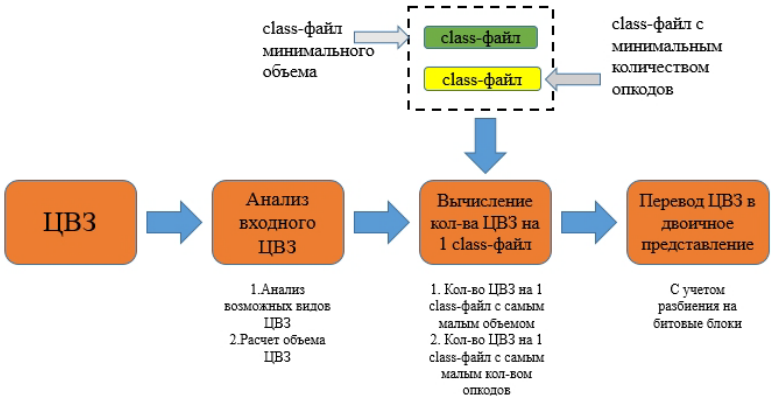

Рисунок 6 – Действия этапа создания цифрового водяного знака

Этап создания цифрового водяного знака основан на результатах анализа classфайлов java-приложения и позволяет создать цифровой водяной знак для каждой из категорий class-файлов, полученной на предыдущем этапе.

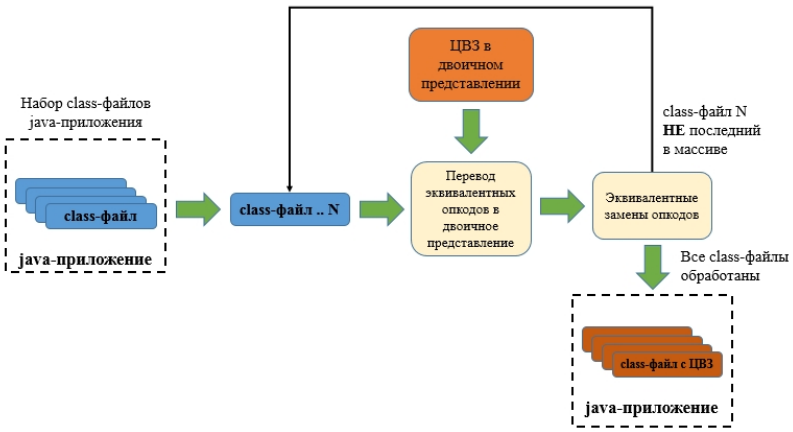

Рисунок 7 – Действия этапа вложения цифрового водяного знака

Таким образом, совокупность эквивалентных замен нескольких групп опкодов и паттернов, функций программирования не однозначно декомпилируемых или декомпилируемых неверно, полностью покрывает риск уничтожения или повреждения цифрового водяного знака посредством атак перекомпиляцией. Также дополнительным фактором являются методы, препятствующие декомпиляции, вызывающие ошибку в работе декомпилятора или получение некорректного исходного кода после работы декомпилятора. Также разработанная методика доведена до реализации алгоритма.

Были подробно проанализированы существующие декомпиляторы class-файлов Java и их возможности. Произведен анализ и сравнение декомпиляторов, описаны их сильные и слабые стороны. Проведено исследование устойчивости цифрового водяного знака к атакам декомпиляцией. Сделан вывод о невозможности повреждения или удаления цифрового водяного знака с помощью атак перекомпиляцией различными декомпиляторами и различными версиями компиляторов Java.

Для оценки class-файлов java-приложения до вложения цифрового водяного знака и после используется модель Нельсона. Вероятность приемлемого результата исполнения class-файла java-приложения рассчитывается по формуле:

$$P = 1 - \sum_{i=1}^{n} y_i \, .$$

А надежность java-приложения на основе равного количества запусков на одинаковых выборках входных данных рассчитывается по формуле:

$$R(n) = (1 - P_1)(1 - P_2) \dots (1 - P_n) = \prod_{j=1}^{n} (1 - P_j).$$

Результаты эксперимента по использованию функций затрудняющих декомпиляцию class-файлов совместно с созданием и вложением цифрового водяного знака в class-файлы java-приложения продемонстрировали, что методика увеличивает вероятность неуспешной декомпиляции java-приложения и его class-файлов до 60-80%, позволяет увеличить количество отказов при нелегитимной декомпиляции java-

приложения, увеличить устойчивость цифрового водяного знака к атакам декомпиляцией до 80-90% , что позволяет доказать выполнение поставленной в настоящем диссертационном исследовании задачи. Эффективность методики продемонстрирована в таблице 3.

Таблица 3 – Эффективность методики вложения ЦВЗ устойчивого к атакам декомпиляцией

Показатель	Известные методики	Разработанная методика
Вероятность успешной декомпиляции java-приложения и его class-файлов, %	70 – 75	20 – 40
Вероятность аварийного завершения работы во время декомпиляции java-приложения с ЦВЗ, %	10 – 15	55 – 60
Вероятность неработоспособности исходного кода после атаки декомпиляцией, %	20 – 25	30 – 35
Успешное разрушение ЦВЗ атакой декомпиляции, %	50 – 60	0 – 5

В четвертой главе разработана методика создания и скрытого вложения цифрового водяного знака регистратора, устойчивого к атакам обфускацией, в байт-код class-файлов java-приложений информационной системы. Методикой в отличие от известных используется два вида цифровых водяных знаков: цифровой водяной знак регистратор информационной системы и тиражируемый цифровой водяной знак в class-файлы java-приложений информационной системы. Вложение ЦВЗ-регистратора в class-файлы с ключевой логикой, при атаке обфускацией увеличивает вероятность приведения java-приложения или class-файла в неработоспособное состояние с невозможностью получения оригинального исходного кода до 70-80%.

Преимущества методики:

- Учитывает инструменты обфускации, как инструменты атаки на цифровой водяной знак, а не оптимизации кода
- Позволяет проводить оценку целостности информационной системы и ее отдельных частей
- Использует два типа цифровых водяных знаков для информационной системы
- Позволяет произвести вложение цифровых водяных знаков таким образом, что применение инструментов продвинутой обфускации приведет к неработоспособности исходного кода
- Возможность автоматизации процесса

Разработанная методика основана на примере работы микросервисной архитектуры информационной системы, состоящей из набора java-приложений и использующей этапы: анализа информационной системы, анализа javaприложений, создания и вложения цифрового водяного знака на основе структуры байт-кода class-

файла, создания цифрового водяного знака регистратора единого для всей информационной системы. Методика продемонстрирована на рисунке 8.

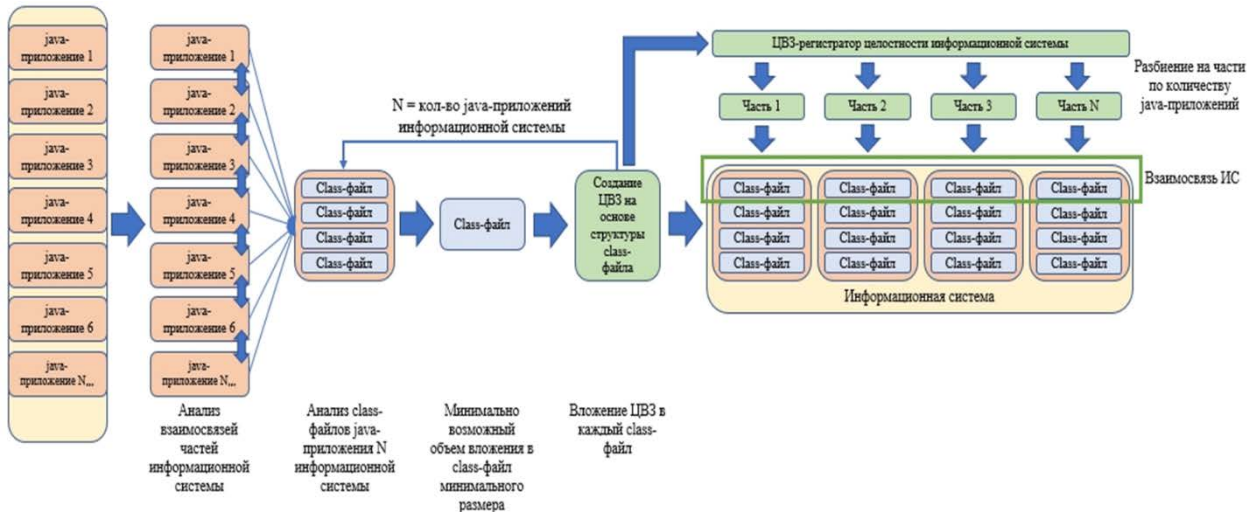

Рисунок 8. Методика создания и вложения цифрового водяного знака регистратора в информационную систему

Подробно разобраны существующие обфускаторы class-файлов Java. Произведен анализ и сравнение обфускаторов. Среди существующих методик вложения ЦВЗ в class-файлы ни у одной нет устойчивости к атакам обфускацией. Проведено исследование устойчивости цифрового водяного знака к атакам обфускацией. Анализ результатов работы 12 обфускаторов показал, что только инструменты продвинутой обфускации в отдельных исключительных случаях могут повредить цифровой водяной знак, однако, при этом в 100% случаев замедляется производительность всей информационной системы и увеличивается занимаемый объем class-файлов до 300%, в 80% случаев исполняемые class-файлы во время запуска в аварийном режиме завершают свою работу, в 100% невозможно производить отладку информационной системы и получить ее исходный код или произвести его редактирование для последующей повторной компиляции.

Произведен статистический анализ, основанный на выборке из 320 class-файлов с ключевым функционалом языка программирования Java, для определения зависимости приведения исходного кода в неработоспособное состояние посредством использования инструментов продвинутой обфускации и определения среднего объема цифрового водяного знака регистратора доступного к вложению.

Вероятность устойчивости ЦВЗ к атакам обфускацией рассчитывается по формуле Бернулли:

$$P_D = C_N^K p^K (1-p)^{N-K}.$$

Оценка устойчивости цифрового водяного знака к разрушающим атакам обфускацией рассчитывается по формуле:

$$P(Y) = 1 - \frac{D}{K}.$$

Оценка сложности потока управления java-приложения в информационной системе производится с помощью метрики Мак-Кейба:

$$V = e - n + 2.$$

Оценка сложности class-файла информационной системы рассчитывается по метрике Мак-Клура:

$$\mathrm{C}(o) = \frac{(D(o) + J(o))}{n}.$$

На основе результатов анализа функциональных особенностей обфускаторов и результатах исследования, сделаны выводы о сложности уничтожения цифрового водяного знака атаками обфускацией и о последствиях данных атак на работоспособность, как отдельных модулей, так и всей информационной системы в целом, что позволяет доказать выполнение поставленной в настоящем диссертационном исследовании задачи. Результаты приведены в таблице 4.

Таблица 4 – Результаты атак обфускацией цифрового водяного знака в 320 class-файлах

Показатель	Известные методики, %	Разработанная методика, %
Кол-во известных обфускаторов способных повредить ЦВЗ, %	90	24
Вероятность невозможности получения исходного кода после атак обфускацией, %	40	70
Вероятность аварийного завершения работы class-файла после обфускации, %	55	80

ЗАКЛЮЧЕНИЕ

Поставленная в диссертационном исследовании цель по увеличению объема и устойчивости к атакам декомпиляцией, обфускацией цифрового водяного знака в интересах защиты авторских прав на исходный код достигнута. Для достижения цели были поставлены и выполнены задачи, получены научные результаты, составляющие следующие итоги исследования:

1. Разработана методика создания и скрытого вложения цифрового водяного знака повышенного объема на основе байт-кода class-файлов.

– Позволяет производить вложения цифрового водяного знака увеличенного объёма в байт-код class-файла за счет расширения набора используемых опкодов;

– Позволяет связать цифровой водяной знак и class-файл, сделав их взаимозависимыми

2. Разработана методика создания и тиражируемого вложения, цифрового водяного знака в class-файлы java-приложения устойчивого к атакам перекомпиляцией направленных на уничтожение вложения.

– Позволяет создать устойчивый к атакам перекомпиляцией цифровой водяной на основе байт-кода class-файлов java-приложения.

– Позволяет произвести вложение цифрового водяного знака в class-файлы java-приложения без изменения его свойств, структуры, размера, работоспособности.

3. Разработана методика создания и тиражируемого вложения, цифрового водяного знака регистратора в class-файлы информационной системы, работающей на Java, устойчивого к атакам обфускацией направленных на деструктуризацию покрывающего сообщения или уничтожение вложения.

– Позволяет создать устойчивый к атакам обфускацией цифровой водяной на основе байт-кода class-файлов информационной системы.

– Позволяет произвести вложение цифрового водяного знака в class-файлы информационной системы без изменения ее свойств, структуры, размера, работоспособности.

СПИСОК РАБОТ, ОПУБЛИКОВАННЫХ ПО ТЕМЕ ДИССЕРТАЦИИ

В рецензируемых изданиях из перечня ВАК при Минобрнауки России

1. **Шариков П.И.** Методика обфускации байт-кода java-приложения с целью его защиты от атак декомпиляцией // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. 2022. № 1. С. 64-72.

2. Иванов А.В., Красов А.В., **Шариков П.И.** Исследование возможностей методики скрытого вложения цифрового водяного знака в class-файлы на виртуализированных платформах с отличающейся архитектурой // Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». 2018. № 2. С. 79-88.

3. Красов А.В., **Шариков П.И.** Методика защиты байт-кода JAVA-программы от декомпиляции и хищения исходного кода злоумышленником // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. 2017. № 1. С. 47-50.

4. **Шариков П.И.**, Красов А.В., Штеренберг С.И. Методика создания и вложения цифрового водяного знака в исполняемые java файлы на основе замен опкодов // T-Comm-Телекоммуникации и Транспорт. 2017. Т. 11. № 3. С. 66-70.

5. **Шариков П.И.**, Красов А.В. Исследование возможности использования java-агентов для вложения скрытого цифрового водяного знака непосредственно перед запуском java-приложения // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. 2019. №. 4. С. 14-18.

6. **Шариков П.И.** Исследование устойчивости цифрового водяного знака к атакам направленной декомпиляции составных частей java-приложения // Электронный сетевой политематический журнал «Научные труды КубГТУ». 2022. № 2. С. 100-112.

7. **Шариков П.И.** Исследование атаки обфускацией на байт-код java-приложения с целью разрушения или повреждения цифрового водяного знака // I-methods. 2022. Т. 14. № 1.

8. **Шариков П.И.** Методика создания и скрытого вложения цифрового водяного знака в байт-код class-файла на основе не декларированных возможностей виртуальной машины java // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и Технические Науки. 2023. № 07/2. С. 165-174.

Свидетельства о государственной регистрации программ для ЭВМ

9. Свидетельство о государственной регистрации программы для ЭВМ № 2020664301 Российская Федерация. Модификатор байт-кода java-программы для скрытого вложения цифрового водяного знака посредством автоматического редактирования байт-кода class-файла / А.В. Красов, А.И. Пешков, **П.И. Шариков**, Г.П. Жиркова; заявитель Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича» (СПбГУТ). № 2020663386; заявл. 30.10.2020; опубл. 11.11.2020.

10. Свидетельство о государственной регистрации программы для ЭВМ № 2020617872 Российская Федерация. Анализатор байт-кода java – программы для скрытого вложения цифрового водяного знака посредством автоматического редактирования байт-кода class-файла / А.В. Красов, А.И. Пешков, **П.И. Шариков**; заявитель Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича» (СПбГУТ). № 2020616770; заявл. 29.06.2020; опубл. 15.07.2020.

11. Свидетельство о государственной регистрации программы для ЭВМ № 2022613440 Российская Федерация. Программное обеспечение по реализации стеганографических методов при передаче сообщения / А.В. Красов, А.М. Гельфанд, **П.И. Шариков**, А.И. Катасонов; заявитель Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича» (СПбГУТ). № 2022612582; заявл. 24.02.2022; опубл. 14.03.2022.

В изданиях, индексируемых Web of Science и Scopus

12. **Sharikov P.I.** et al. Research of the possibility of hidden embedding of a digital watermark using practical methods of channel steganography // International Symposium on Intelligent and Distributed Computing. Springer, Cham, 2019. P. 203-209.

13. **Sharikov P.I.**, Krasov A.V., Volkogonov V.N. A study of the correctness of the execution of a class file with an embedded digital watermark in different environments // IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2020. V. 862. No. 5. P. 052052.

14. Krasov A., **Sharikov P.** A Technique for Analyzing Bytecode in a Java Project for the Purpose of an Automated Assessment of the Possibility and Effectiveness of the Hidden Investment of Information and its Volumes in a Java Project // 2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). IEEE, 2020. P. 258-263.

15. **Sharikov P.** et al. A Technique for Detecting the Substitution of a Java-Module of an Information System Prone to Pharming with Using a Hidden Embedding of a Digital Watermark Resistant to Decompilation // 2021 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). IEEE, 2021. P. 219-223.

В других изданиях

16. Красов А.В., Гельфанд А.М., Коржик В.И., Котенко И.В., Петрив Р.Б., Сахаров Д.В., Ушаков И.А., **Шариков П.И.**, Юркин Д.В. Построение доверенной вычислительной среды/ Санкт-Петербург: издатель Петрив Роман Богданович, 2019. 108 с. (монография)

17. **Шариков П.И.**, Красов А.В. Исследование возможности вложения цифрового водяного знака в байт-код путем замены уязвимого байт-кода Java класса // Информационная безопасность регионов России (ИБРР-2017). 2017. С. 499-500.

18. **Шариков П.И.**, Красов А.В. Исследование уязвимости сериализации и десериализации данных в java // Региональная информатика и информационная безопасность: сборник научных трудов. Санкт-Петербургское Общество информатики, вычислительной техники, систем связи и управления. 2017. С. 333-336.

19. Красов А.В., **Шариков П.И.** Обеспечение безопасности java программ посредством вложения программного водяного знака // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2018). VII Международная научно-техническая и научно-методическая конференция. Санкт-Петербург, 2018. С. 517-520.

20. Красов А.В., **Шариков П.И.** Метод использования самомодифицирующегося кода для защиты приложения с кодовым зашумлением // Телекоммуникационные и вычислительные системы Международный форум информатизации (МФМ-2016); Международный конгресс (CTN-2016) «Коммуникационные технологии сети». Труды конференции. 2016. С. 118-121.

21. **Шариков П.И.** Методика быстрой обфускации class-файлов java-приложения с минимальным использованием ресурсов // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2019). VIII Международная научно-техническая и научно-методическая конференция. Санкт-Петербург, 2019. С. 548-551.

22. **Шариков П.И.** Методика анализа байт-кода с целью оценки эффективности скрытого вложения информации и ее объемов в java-проект // Молодежная научная школа кафедры «Защищенные системы связи». 2020. Т. 1. № 1 (1). С. 30-34.

23. **Шариков П.И.** Алгоритм вложения цифрового водяного знака в исполняемые файлы необходимый для подтверждения авторства программы при судебной экспертизе // Молодежная научная школа кафедры «Защищенные системы связи». 2021. Т. 1. № 1 (3). С. 7-11.

Подписано в печать 22.12.2023. Формат 60×84 1/16.
1,0 а.л. Тираж 100 экз.

Отпечатано в СПбГУТ, 193232, Санкт-Петербург, пр. Большевиков, д. 22, корп. 1