

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ  
Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

**ОБФУСКАЦИЯ ИСХОДНОГО КОДА**  
**АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ**

студентки 2 курса 248 группы  
направления 09.04.03 – Прикладная информатика  
механико-математического факультета  
Иргискиной Екатерины Алексеевны

Научный руководитель  
доцент, к.ф.-м.н.

Р.Н.Фадеев

Заведующий кафедрой  
д.ф.-м.н., доцент

С.П.Сидоров

Саратов 2023

## Введение

Постоянно прогрессирующие в своем развитии современные компьютерные технологии делают актуальными различные задачи в сфере информационной безопасности. Одним из приоритетных направлений защиты информации является защита программных продуктов (ПП). ПП являются интеллектуальной собственностью, как и сама информация, имея схожие механизмы распространения.

Разработка наиболее эффективной методики защиты для различных видов ПП в настоящее время становится одной из важнейших задач большинства программистов, которые занимаются разработкой специализированного платного ПО, так как это позволяет им продавать свой труд и исключить возможности его нелегального использования среди потребителей.

Обфускация - процесс запутывания ПП, позволяющий видоизменять исходный код ПО таким образом, что пользовательский интерфейс и поведение программы останутся неизменными. Иначе говоря, обфускация - процесс приведения исходного текста или исполняемого кода программы к виду, сохраняющему ее функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию после процесса декомпиляции.

Таким образом, тема магистерской диссертации - обфускация программного кода - является актуальной.

Целью данной выпускной работы являлось написание программы, которая обфусцирует код. Для достижения указанной цели поставлены следующие задачи:

- Провести анализ литературы и интернет-источников, на основе которых выделить основные проблемы в данном вопросе, сформулировать требования к функциональности программного продукта
- Выбрать язык программирования, на котором будет написан обфускатор
- Написать программу для обфусцирования кода

- Протестировать данную программу на различных кодах.

## **1 Основное содержание работы**

Работа состоит из 9 глав:

1. Основные способы защиты ПП
2. Методы защиты ПП
3. История развития обфускации
4. Достоинства обфускации
5. Алгоритмы процесса обфускации
6. Виды обфускации
7. Выбор языка программирования
8. Практическая часть
9. Заключение

причем каждая глава содержит несколько разделов. В 1-7 главах описана теоретическая составляющая дипломной работы.

В работе рассматриваются два основных способа защиты программного продукта. Дается подробное описание техническому способу защиты.

Описывается история развития обфускации.

Также в работе приведены достоинства обфускации, такие как:

- обфускация не ограничивается государственными стандартами на длину ключа;
- при шифровании ПП одним из самых популярных способов защиты является шифрование байт-кода. Байт-код хранится в приложении в зашифрованном виде и расшифровывается только в момент выполнения. В уже рабочем приложении злоумышленник может получить доступ к оперативной памяти, памяти процессора и его регистрам. Поэтому любой фрагмент приложения, который был расшифрован, может быть сразу использован злоумышленником для анализа приложения. При использовании обфускации шифрования байт-кода не происходит, идет запутывание кода самого ПО, поэтому злоумышленник не получит никаких преимуществ от получения фрагментов приложения из оперативной памяти или регистров

- для того, чтобы приложение работало, требуется получить доступ к его байт-коду. Если байт-код изначально зашифрован, то для дальнейшей работы требуется его расшифровка. Данный процесс может быть времязатратным при использовании более надежных алгоритмов, тогда как при обфускации дополнительного времени на шифровку и расшифровку байт-кода не требуется

- при шифровании всех исходных кодов вместе и использовании поточных алгоритмов шифрования, становится случайный доступ к участкам исходного кода. Это означает, что для работы приложения требуется изначально расшифровать его полностью. При использовании запутывания кода данного недостатка не наблюдается - возможность случайного доступа к определенным участкам сохраняется.

Также в работе рассказано о лексической обфускации, обфускации данных, управления и превентивной обфускации.

В седьмой главе описаны наиболее популярные языки программирования, объясняется выбор в пользу Java.

Шестая глава - практическая часть дипломной работы. В рамках магистерской работы необходимо написать обфускатор кода.

Данная программа будет обфусцировать код с помощью переименования переменных, удаления однострочных и многострочных комментариев, добавления лишних циклов и т.д.

Однако в любой программе есть команды, которые нельзя заменять иначе смысл программы нарушится, например, “if”, “while” и т.д. Для этого создадим файл, в котором перечислим все команды. Аналогично поступим с типами данных.

В ходе работы был написан код программы.

Для тестирования можно запустить любую программу. В нашем случае - это программа нахождения наибольшего общего делителя,

В выводе получаем обфусцированный код. Видим, что были изменены переменные, удалены все комментарии, также был вставлен оператор `goto`, который эффективно выполняет первоначальную последовательность инструкций, но в обходном порядке, что затрудняет выполнение логического потока.

Также проделаны тесты на нескольких кодах: быстрой сортировки, бинарного поиска.

## **Заключение**

В дипломной работе были рассмотрены методы защиты программного продукта, а именно - обфускация.

Был написан обфускатор кода, а также протестирован на программах нахождения наибольшего общего делителя, быстрой сортировки, бинарного поиска.

Таким образом, была достигнута цель данной выпускной работы.

Также были достигнуты задачи:

- Провести анализ литературы и интернет-источников, на основе которых выделить основные проблемы в данном вопросе, сформулировать требования к функциональности программного продукта
- Выбрать язык программирования, на котором будет написан обфускатор
- Написать программу для обфусцирования кода
- Протестировать данную программу на различных кодах.