

УДК 004.056.53

Алгоритмы обфускации и перспективы их применения

Бабуркина Софья Сергеевна¹

s.baburkina@mail.ru

Швецова Анастасия Антоновна²

flamy.snow@gmail.com

¹ МГТУ им. Н.Э. Баумана, Москва, Россия² Лаборатория Касперского, Москва, Россия

Аннотация. В работе рассмотрены существующие алгоритмы и методы обфускации. Представлены примеры реализации различных методов обфускации. Приведен обзор актуальных сценариев применения обфускации. Рассмотрены методы и параметры оценки алгоритмов обфускации. Сформулированы гипотезы о дальнейшем направлении развития и применения обфусцирующих алгоритмов.

Ключевые слова: обфускация, стойкость обфускации, обратная разработка, защита ИИ

Введение. Понятие обфускации впервые было приведено в 1997 году в работе [1] — это такое преобразование исходного текста или исполняемого кода программы, что ее анализ будет значительно затруднен, а результат ее выполнения не изменится. Авторы рассматривали обфускацию в качестве метода защиты от кражи интеллектуальной собственности, но, помимо этого, обфускация кода также применяется авторами ВПО для избежания обнаружения программ средствами АВЗ.

Методы и материалы; результаты. Реализация обфускации программы происходит по некоторому алгоритму, определяющему набор и порядок применения различных методов обфускации. Обобщенным алгоритмом обфускации считается Алгоритм Колберга. Основная идея алгоритма — многократное применение набора трансформаций T к выбранным частям кода S с целью получения обфусцированного кода S' , сохраняющего логику работы исходного кода S [2].

Структура алгоритма и набор используемых методов обфускации зависят от целей обфускации, структуры оригинальной программы и технологий, используемых при ее разработке.

Конкретную реализацию обфускации можно классифицировать по ее уровню. В зависимости от того, над чем именно проводятся обфусцирующие операции, это может быть высший уровень — обфускация исходного кода программы; уровень промежуточного кода — обфускация абстрактного набора инструкций, генерируемого при компиляции программ, написанных на некоторых языках программирования; и низший уровень — обфускация машинного кода [4].

Методы обфускации тесно связаны с особенностями языка программы и выбранным уровнем обфускации, но их также предлагается разделить на три

обобщенные группы: обфускация данных, лексическая обфускация и обфускация потока управления.

Обфускация данных предполагает различные преобразования используемых значений переменных и параметров вызываемых функций: разделение одного значения на несколько переменных, преобразование кодировок и создание собственных структур данных.

К лексической обфускации относятся преобразования имен переменных и функций, комментариев и их взаимного расположения таким образом, чтобы чтение кода исходной программы было затруднено.

Обфускация потока управления позволяет скрыть структуру программы, сделав неочевидными расположение исполняемых инструкций, точек входа и выхода.

При выборе методов и формировании алгоритма обфускации программы необходимо решить две основные проблемы: сохранение логики исходной программы и приемлемое увеличение времени ее выполнения и объема затрачиваемых на это ресурсов. Оценку стойкости обфускации возможно выполнить по модели «черного ящика» — так обфускация считается стойкой в случае, если при анализе исходного кода программы за разумное время возможно получить не больше информации, чем при динамическом анализе программы как «черного ящика». Авторы работы кроме такой оценки предлагают вводить различные метрики для эффективности методов обфускации ПО: размер процедуры, сложность циклической структуры, сложность потока данных процедуры и проч.

Обратную разработку обфусцированных программ возможно проводить вручную, автоматизированно или с привлечением больших языковых моделей. В первых двух случаях человеку для проведения деобфускации необходимо самостоятельно опознать обфусцированные участки кода и используемые методы обфускации, и сформулировать для них обратные преобразования, если это возможно (удаление лишних блоков кода, переименование переменных и т. п.), но в работе демонстрируется возможность проведения такой классификации языковой моделью.

На сегодняшний день существует множество вариантов обфусцирующих алгоритмов — на платформе Gitlab размещены 1124 публичных проекта, посвященных обфускации, что подтверждает актуальность исследуемой темы.

В качестве перспективного направления использования обфускации предлагается рассмотреть ее применение к исходным кодам и параметрам языковых моделей.

Заключение. В работе были рассмотрены основные сферы применения обфускации, приведен обзор методов реализации, оценки эффективности и деобфускации. Было выдвинуто предположение об эффективности применения методов обфускации к исходным кодам больших языковых моделей, сформулирован предварительный список обфусцируемых блоков и параметров. Были предложены методы по снижению потерь производительности при использовании алгоритмов обфускации для языковых моделей.

Список источников

- [1] Ерболатов А.Н. Реверс-инжиниринг против искусственного интеллекта: анализ самозащищенных моделей ИИ. *Вестник науки*, 2025, № 2 (83), с. 436–443.
- [2] *Обфускация и защита программных продуктов*. URL: <https://citforum.ru/security/articles/obfus/> (дата обращения 25.09.2025).
- [3] Красов А.В., Зуев И.П., Карельский П.В., Радынская В.Е., Гераськина В.С. Алгоритмы и методы защиты программного кода на базе обфускации. *I-methods*, 2020, № 1 с. 1–12.
- [4] MDN Glossary: Base64. URL: <https://developer.mozilla.org/en-US/docs/Glossary/Base64> (accessed 25.09.2025).