(cc) BY

Бородин Андрей Викторович
канд. экон. наук, доцент, заведующий кафедрой
Долгушев Евгений Дмитриевич
студент
ФГБОУ ВО «Поволжский государственный
технологический университет»
г. Йошкар-Ола, Республика Марий Эл

DOI 10.21661/r-112829

ПОСТАНОВКА ЗАДАЧИ ОБФУСКАЦИИ ПУЛА КОНСТАНТ

Аннотация*: в работе раскрыта роль системы методов обфускации программ как одного из средств защиты авторских прав в области разработки как программного обеспечения, так и некоторых видов аппаратных средств. Исследован подход к сокрытию семантики программ, основанный на сокращении количества позиций прямого задания и исключении прямого использования констант в программе при совместной минимизации размера явно заданной части пула констант и сложности кода формирования остальной его части. Дана формальная постановка задачи в виде оптимизационной модели.*

Ключевые слова*: авторское право, интеллектуальная собственность, класс сложности, линейная комбинация, обфускация, оптимизационная задача, пул констант, цифровой продукт.*

Проблема защиты интеллектуальной собственности в последние годы становится все более и более актуальной. Связано это, как с взрывным ростом количества новых идей, продуктов и технологий, так и с упрощением процедур их описания и реализации. Информационные технологии здесь оказались катализатором инновационного процесса. Они предложили гигантское множество простых для пользователя инструментов поддержки инновационной деятельности и постарались, по возможности, предложить технологии защиты интеллектуальной собственности. В то же время, очевидно, что прогресс в развитии простых и

удобных в эксплуатации инструментальных средств поддержки инноваций существенно опережает создание и развитие инструментария защиты интеллектуальной собственности. В ряду инструментальных средств, для которых справедливо все выше сказанное, можно отметить продуктовую линейку Microsoft Office, многие популярные продукты компаний Corel, MindJet, CA Technologies и т. д. Эти программные средства благодаря поддержке языка программирования Visual Basic for Applications (VBA) [3] взяли на себя роль метасистем – инструментов создания инноваций в области программного обеспечения со стороны прикладных специалистов, не считавших себя ранее специалистами в области программирования. Еще ярче эта ситуация прослеживается в сфере аппаратного обеспечения. Появление и развитие таких языков описания аппаратуры как Verilog [5] и VHDL [1], с одной стороны, чрезвычайно упростило и ускорило создание множества аппаратных решений, а, с другой, сделало ряд весьма сложных, интеллектуально емких (уникальных) проектов уязвимыми к несанкционированному использованию.

Некоторым препятствием на пути утечки интеллектуальной собственности в описанном контексте являются технологии обфускации. Неформально под обфускацией понимается приведение исходного текста или исполняемого кода программы к виду, сохраняющему её функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию после декомпиляции [4]. Таким образом, разработка новых методов обфускации с различными заданными свойствами представляет значительный интерес.

В целом ряде случаев, знание констант, используемых в программе, способно существенно облегчить понимание семантики алгоритма. Поэтому сокрытие явного использования констант может стать одним из препятствий на пути рефакторинга программ, по крайней мере, при ручном анализе кода. Для автоматизированного анализа обфускация констант вряд ли может стать серьезным препятствием, но для непрофессионального ручного анализа она может создать заметные проблемы.

Заметим, что константы основных базовых типов языков программирования легко могут быть представлены целыми неотрицательными числами. Константа целого типа – это целое неотрицательное со знаком, вещественного типа – это целое неотрицательное со знаком для мантиссы и целое неотрицательное со знаком для порядка, константа строкового типа – это набор ординалов, соответствующих символам, иначе, набор целых неотрицательных чисел. Таким образом, задачу обфускации констант базовых типов можно свести к задаче обфускации целых неотрицательных констант [2].

Формализуем задачу. Пусть C – множество целых неотрицательных констант, используемых в обфусцируемой программе, $C \subset \mathbb{Z}_0^+$. Здесь $\mathbb{Z}_0^+$ – множество целых неотрицательных чисел. Пусть, далее, $m = \max C + 1$, то есть $C \subseteq \mathbb{Z}_m \underset{Def}{=} \{0, 1, \ldots, m-1\}$. Пусть R – множество (вспомогательных) констант, которые представлены в программе не только своими идентификаторами, но и явно записанными целыми числами без знака. Теперь можно ввести множество констант, используемых в программе и заданных явно,

$$C_0(R) = C \cap R.$$

На первом шаге, используя данное множество констант, можно его расширить с помощью ряда линейных комбинаций:

$$C_1(R, A_0, x_0) = C_0(R) \cup \left\{ c:\ c \in C \setminus C_0(R), c = \sum_{a \in A_0(c)} x_0(c,a)\, a \right\},$$

где $A_0:\ C \setminus C_0(R) \to 2^{C_0(R)}$ – некоторая функция, представляющая подмножество констант, из множества $C_0(R)$, используемых в линейной комбинации для порождения некоторых констант из множества $C \setminus C_0(R)$ на первой итерации, $x_0 : \left[C \setminus C_0(R)\right] \times C_0(R) \to C_0(R)$ – функция, которая при порождении некоторых констант c, из множества $C \setminus C_0(R)$, сопоставляет каждому члену линейной комбинации из множества $A_0(c)$ коэффициент из множества $C_0(R)$.

На последующих шагах мы можем постепенно расширять множество констант, определенных через константы предыдущих шагов:

$$C_{k+1}\left(R, A_k, x_k\right) = C_k\left(R, A_{k-1}, x_{k-1}\right) \cup$$

$$\cup \left\{ c:\ c \in C \setminus C_k\left(R, A_{k-1}, x_{k-1}\right), c = \sum_{a \in A_k(c)} x_k(c,a)\, a \right\},$$

где $A_k:\ C \setminus C_k\left(R, A_{k-1}, x_{k-1}\right) \to 2^{C_k\left(R, A_{k-1}, x_{k-1}\right)}$ – некоторая функция, представляющая подмножество констант, из множества $C_k\left(R, A_{k-1}, x_{k-1}\right)$, используемых в линейной комбинации для порождения некоторых констант из множества $C \setminus C_k\left(R, A_{k-1}, x_{k-1}\right)$ на $(k+1)$-й итерации,

$x_k:\ \left[C \setminus C_k\left(R, A_{k-1}, x_{k-1}\right)\right] \times C_k\left(R, A_{k-1}, x_{k-1}\right) \to C_k\left(R, A_{k-1}, x_{k-1}\right)$ – функция, которая при порождении некоторых констант c, из множества $C \setminus C_k\left(R, A_{k-1}, x_{k-1}\right)$, сопоставляет каждому члену линейной комбинации из множества $A_k(c)$ коэффициент из множества $C_k\left(R, A_{k-1}, x_{k-1}\right)$, $k = 1, 2, \ldots$.

Очевидно, что для некоторых R на некоторой итерации k' будет достигнуто определение всех необходимых констант:

$$C_{k'}\left(R, A_{k'-1}, x_{k'-1}\right) = C.$$

Определим множество возможных параметров порождения заданного множества констант при фиксированном множестве непосредственно заданных констант:

$$V(R) \underset{Def}{=} \left\{ \left(k', \left(A_k, x_k\right)_{k=0}^{k=k'-1} \right):\ C_{k'}\left(R, A_{k'-1}, x_{k'-1}\right) = C \right\}.$$

Теперь можно сформулировать ряд оптимизационных задач.

1. Задача минимизации количества операций при порождении заданного набора констант C при фиксированном множестве непосредственно определенных констант R:

$$\pi(R) \underset{Def}{=} \left(k'', \left(A'_k, x'_k\right)_{k=0}^{k=k''-1}\right)(R) = \underset{V(R)}{\arg\min} \sum_{k=1}^{k'} \sum_{c \in C_k\left(R,\, A_{k-1},\, x_{k-1}\right)} \left|A_k(c)\right|.$$

2. Задача минимизации количества непосредственно определенных констант среди вариантов с минимальной сложностью формул:

$$R' = \underset{R \in \mathrm{Dom}\,\pi}{\arg\min} \left|R\right|,$$

где $\mathrm{Dom}\,\pi$ – область определения функции π.

Значение функции $\pi(R')$ можно рассматривать как оптимальную (в определенном смысле) схему обфускации констант. Следует, однако, отметить, что алгоритм точного решения задачи вычисления $\pi(R')$ лежит, по-видимому, в классе *EXPSPACE* [8; 9]. Поэтому на практике задача вычисления $\pi(R')$ решается каким-либо приближенным эмпирическим методом.

Разработка эффективных (пусть и приближенных) алгоритмов решения поставленной задачи может оказать значительное влияние на качество обфускации кода и, следовательно, способна повысить степень защищенности авторских прав для целого ряда цифровых продуктов. Так, предложенный подход к обфускации пула констант был апробирован в ручном режиме поиска решений при разработке пакета прикладных программ (ППП) «МультиМИР» версии 1.1 [6; 7], реализованного на языке VBA, для защиты части программных модулей, содержащих элементы «know how», в связи с началом распространения данной версии ППП среди заказчиков.

Список литературы

1. Бибило П.Н. Основы языка VHDL / П.Н. Бибило. – М.: Книжный дом «Либроком», 2014. – 328 с.

2. Бородин А.В. Обфускация пула констант как задача построения минимальной системы целочисленных линейных комбинаций / А.В. Бородин, Е.Д. Долгушев // Образование, наука, бизнес: развитие и перспективы: материалы III международной научно-практической конференции (6 мая 2016 г.). – Саратов: Издательство ЦПМ «Академия Бизнеса», 2016. – С. 7–13.

3. Михеев Р.Н. VBA и программирование в MS Office для пользователей / Р.Н. Михеев. – СПб.: БХВ-Петербург, 2006. – 384 с.

4. Обфускация (программное обеспечение) // Википедия. Свободная энциклопедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%84%D1%83%D1%81%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) (дата обращения: 28.02.2016).

5. Стешенко В.Б. ПЛИС фирмы Altera: элементная база, система проектирования и языки описания аппаратуры / В.Б. Стешенко. – М.: ДМК Пресс, 2016. – 576 с.

6. Уразаева Т.А. О функциональности пакета прикладных программ «МультиМИР» / Т.А. Уразаева // Современные проблемы и перспективы социально-экономического развития предприятий, отраслей, регионов. – Йошкар-Ола: Поволжский государственный технологический университет, 2014. – С. 261–265.

7. Уразаева Т.А. Пакет прикладных программ «МультиМИР»: архитектура и применение / Т.А. Уразаева // NB: Кибернетика и программирование. – 2014. – №5. – С. 34–61. – DOI: 10.7256/2306–4196.2014.5.12962.

8. Papadimitriou C.H. Computational complexity / C.H. Papadimitriou. – New York: Addison-Wesley Publishing Company, Inc., 1994. – 523 p.

9. Sipser M. Introduction to the Theory of Computation / M. Sipser. – Boston: Thomson Course Technology, 2006. – 431 p.