Математика и информатика

УДК 681.3

М.К. БУЗА, Е.Н. ЛИВАК

ИДЕНТИФИКАЦИЯ КОМПЬЮТЕРНЫХ ПРОГРАММ

• The identification of computer programs for defending of authorities rights is based on supplying the program with code by reticent information about controling rights and is based on classical and specially deviced methods of cryptography and computer's steganography.

Защита программного продукта от несанкционированного воспроизведения, использования и/или модификации направлена прежде всего на поддержку авторских прав его создателя.

Для защиты программного обеспечения (ПО) используются, как правило, программные, аппаратные и программно-аппаратные средства. Традиционно они основаны на противодействии либо незаконному копированию программ, либо попыткам запуска и/или исполнения нелицензионных копий продукта, включая и меры предотвращения попыток нарушителя исследовать логику работы программы. Как показывает практика, такие методы в различной степени затрудняют нелегальное распространение и использование ПО, но не решают проблему.

Альтернативой традиционным подходам к защите авторских прав в области ПО является сопровождение программного кода (каждой копии программы) скрытой информацией об управлении правами, подтверждающей авторство на программу.

Основные понятия и определения

Информация об управлении правами – это информация, которая сопровождает программный продукт и идентифицирует его автора. Данное понятие определено Законом Республики Беларусь «Об авторском праве и смежных правах» [1, ст. 4]. Заметим также, что устранение или изменение любой электронной информации об управлении правами без разрешения правообладателя является нарушением авторского права [1, ст. 39, п. 5].

Внедрение в программы скрытой информации об управлении правами и ее сопровождение назовем *идентификацией программ*, а коды, в которых представлена скрытая информация об управлении правами, – *идентификационными метками*.

Аутентификация программы – это установление экспертом того факта, что программа, вероятно, не модифицирована и принадлежит данному автору.

Надежность идентификационных меток будем связывать с мощностью механизма их защиты от несанкционированной модификации или удале-

This document has been edited with Infix PDF Editor - free for non-commercial use.

To remove this notice, visit: www.iceni.com/unlock.htm

ния. *Мощность механизма защиты* – это его способность противостоять прямым атакам злоумышленника. Используются следующие градации мощности защиты: *базовая, средняя и высокая*, определенные в существующих стандартах по оценке безопасности информационных технологий.

Теоретические основы системы идентификации программ

В качестве практического средства идентификации программ предлагается автоматизированная система, предназначенная для создания и внедрения идентификационных меток в защищаемые программные продукты, а также для аутентификации программ. Автоматизированная система базируется на специально разработанной ***технологии идентификации программ***, в основе которой лежат классические и оригинальные методы криптографии и компьютерной стеганографии, алгоритмы теории кодирования. Объектом защиты является исполняемый файл программы.

Наряду с решением основной задачи (сопровождение объекта защиты информацией об управлении правами) технология идентификации обеспечивает надежность как самой системы идентификации программ, так и, в частности, идентификационных меток, т. е. гарантирует конфиденциальность, целостность и доступность информации об управлении правами.

Проблема стойкости идентификационных меток к попыткам их обнаружения и удаления (модификации) нарушителем решается с помощью методов и механизмов, используемых для создания и внедрения меток.

Отметим некоторые существенные особенности технологии идентификации программ.

1. В программный код внедряются идентификационные метки различного типа, количество которых определяется в соответствии с параметрами защищаемого объекта и авторской информации.

2. Идентификационные метки кодируются классическими и специально разработанными криптографическими алгоритмами.

3. Для внедрения идентификационных меток применяются как традиционные, так и оригинальные стеганографические методы.

4. Используется специфический психологический подход к защите идентификационных меток, базирующийся на индивидуальном психологическом барьере злоумышленника и его квалификации.

5. Идентификационные метки закодированы так, что позволяют обнаруживать факт их модификации неавторизованными лицами, а также (по желанию пользователя системы) способны самовосстанавливаться.

Структура автоматизированной системы идентификации программ

Система идентификации программных продуктов состоит из обеспечивающей и функциональной подсистем.

Обеспечивающая подсистема (рис. 1) идентифицирует объект защиты и выполняет следующие функции:

- создание идентификационных меток на основе авторской информации об управлении правами и других параметров;
- сохранение эталонной информации на отчуждаемый носитель;
- внедрение в защищаемый объект скрытой информации об управлении правами;
- внедрение в защищаемый объект специального модуля, обнаруживающего модификацию (удаление) меток и восстанавливающего их.

Функциональная подсистема производит аутентификацию программ и позволяет:

This document has been edited with Infix PDF Editor - free for non-commercial use.
To remove this notice, visit: www.iceni.com/unlock.htm

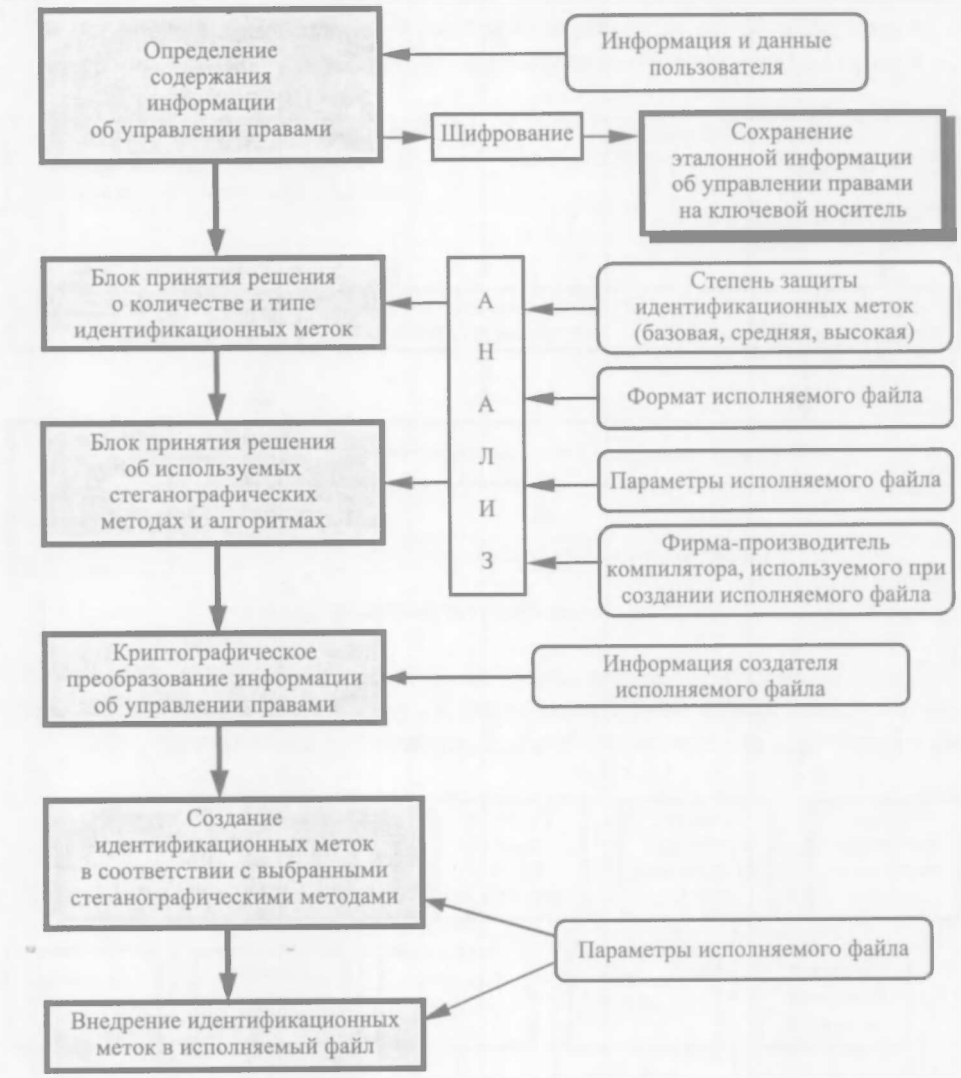

Рис. 1. Структура обеспечивающей подсистемы

• считывать информацию об управлении правами, сопровождающую защищенный объект;

• устанавливать факт авторства посредством сравнения с предъявляемой эталонной информацией об управлении правами.

На основе определения содержания информации об управлении правами на защищаемый программный продукт строится идентификатор программы – уникальный цифровой код (рис. 2), который записывается на специальный отчуждаемый ключевой носитель и хранится у автора. В качестве ключевого носителя в рассматриваемой версии системы используется гибкий магнитный диск. Заметим, что в случае дополнения системы аппаратными средствами любое современное устройство, предназначенное для хранения персональной информации (магнитная карта, электронный ключ, жетон и т. п.), может быть использовано как ключевой носитель.

Основной компонентой данной системы является блок «Анализ», который принимает следующие решения:

• об используемых типах идентификационных меток;

This document has been edited with Infix PDF Editor - free for non-commercial use.
To remove this notice, visit: www.iceni.com/unlock.htm

- о количестве таких меток, внедряемых в исполняемый файл;
- о применяемых стеганографических методах и алгоритмах.

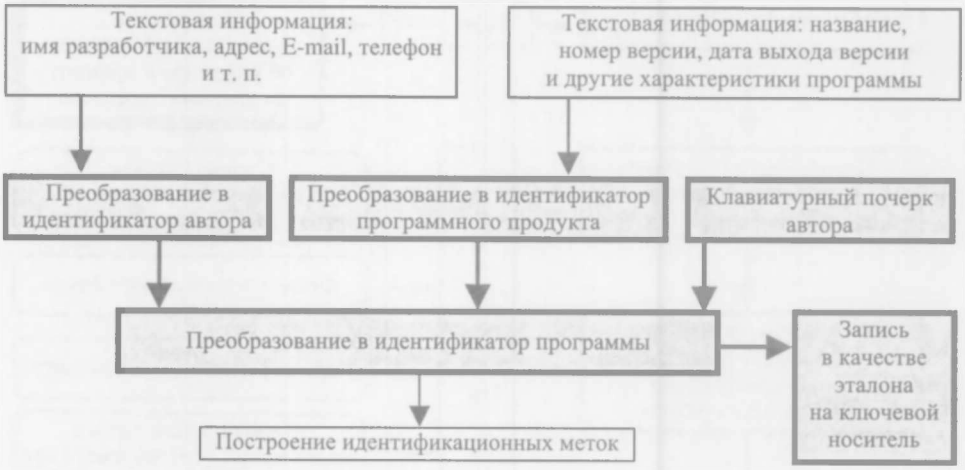

Рис. 2. Построение идентификатора программы

Часть подсистемы, производящая анализ параметров исполняемого файла, – одна из наиболее ответственных. На рис. 3 изображены анализируемые структуры исполняемого файла в формате Portable Executable (PE).

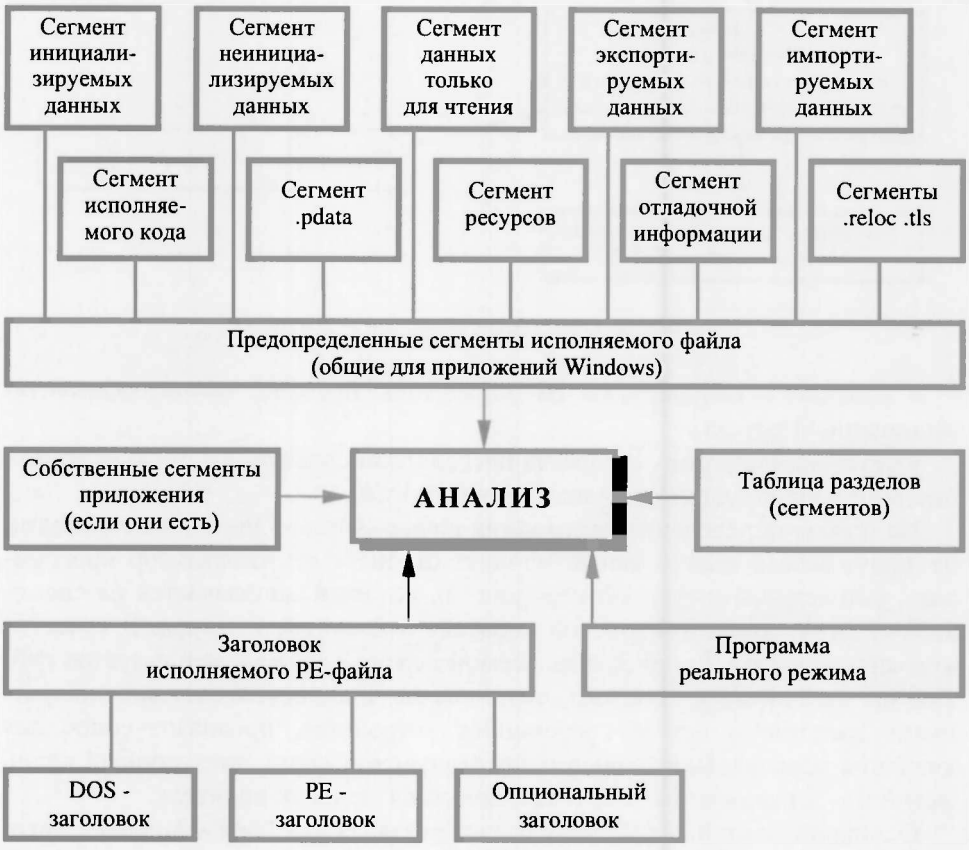

Рис. 3. Анализируемые параметры исполняемого PE-файла

This document has been edited with Infix PDF Editor - free for non-commercial use.
To remove this notice, visit: www.iceni.com/unlock.htm

В процессе анализа определяются общее количество байт, смещения и размер отдельных участков файла, доступных для внедрения меток.

Принципы создания и внедрения идентификационных меток

Технология идентификации программ предполагает внедрение идентификационных меток нескольких типов.

Построение и внедрение идентификационных меток первого типа

Они предназначены для обеспечения **базовой** мощности механизма защиты идентификационных меток и представляют собой зашифрованный с помощью одного из криптоалгоритмов идентификатор программы с последующим сжатием. В качестве ключа шифрования используется идентификатор автора.

Для внедрения идентификационных меток первого типа применяются стеганографические методы, основанные на наличии неиспользуемых (зарезервированных) полей заголовков файлов в форматах EXE (DOS) и PE (Windows, UNIX). Исследование форматов исполняемых файлов (DOS и Windows) показало, что общий размер зарезервированных полей является достаточным для внедрения некоторого количества идентификационных меток первого типа.

Построение и внедрение идентификационных меток второго типа

Данный тип меток предназначен для обеспечения **средней** мощности механизма защиты идентификационных меток.

Пусть идентификатор программы – последовательность десятичных цифр длины d: $\alpha = (\alpha_1 \alpha_2 \ldots \alpha_j \ldots \alpha_d)$, где $\alpha_j \in \{0..9\}$.

Каждую десятичную цифру в данной последовательности запишем соответствующей двоичной тетрадой. Получаем последовательность длины $4d$ в двоичном алфавите:

$$\varphi = (\varphi_1 \varphi_2 \ldots \varphi_j \ldots \varphi_{4d}), \text{ где } \varphi_j \in \{0, 1\}.$$

Идентификационная метка второго типа строится на основе идентификатора программы φ с помощью функции V_i (i – номер метки), осуществляющей выбор n символов (где n кратно 3) из последовательности φ по заданному правилу, и первоначально представляет собой последовательность символов длины n в двоичном алфавите:

$$V_i(\varphi) = (\gamma_1 \gamma_2 \ldots \gamma_l \ldots \gamma_n) = \gamma_i, \text{ где } \gamma_j \in \{0, 1\}, \; n < 4d. \qquad (1)$$

Затем n-битный блок γ_i подвергается операциям:

1) перестановки $P_i(\gamma_i)$,

2) поблочной подстановки $S(P_i(\gamma_i))$,

3) сложению по модулю 2 с определенными операцией W_i битами результата операции подстановки, примененной для двоичной последовательности φ.

Таким образом, идентификационная метка второго типа с номером i строится с помощью функции F_i, где

$$F_i(\varphi) = S(P_i(V_i(\varphi))) \oplus W_i(S(\varphi)).$$

Для внедрения идентификационных меток второго типа используются стеганографические методы, которые условно можно разделить на две группы:

1) методы, основанные на наличии полей заголовков файлов в форматах EXE и PE, изменение значений которых не влияет на корректную работу программы;

2) методы и алгоритмы, использующие принципы технологий внедрения вирусов в исполняемые файлы в форматах EXE и PE.

This document has been edited with Infix PDF Editor - free for non-commercial use.
To remove this notice, visit: www.iceni.com/unlock.htm

Построение и внедрение идентификационных меток третьего типа

Этот тип меток (самовосстанавливающиеся) предназначен для обеспечения **высокой** мощности механизма защиты идентификационных меток.

Самовосстанавливающиеся метки строятся на основе последовательности γ символов длины n в двоичном алфавите (1) следующим образом.

Последовательность γ разбивается на три части. Представим ее в виде:

$$\gamma = (\xi_1\xi_2 \dots \xi_k\psi_1\psi_2 \dots \psi_k\eta_1\eta_2 \dots \eta_k), \text{ где } k = n/3.$$

Обозначим $\xi = (\xi_1\xi_2 \dots \xi_k)$; $\psi = (\psi_1\psi_2 \dots \psi_k)$; $\eta = (\eta_1\eta_2 \dots \eta_k)$. Каждый из наборов ξ, ψ, η в отдельности подвергается дальнейшему кодированию:

$$\xi \to \chi_\xi; \quad \psi \to \chi_\psi; \quad \eta \to \chi_\eta.$$

Рассмотрим алгоритм кодирования на примере набора ξ.

Критическим участком исполняемого файла назовем такую последовательность байт, в которой каждый бит существенно влияет на работоспособность файла, т. е. при изменении хотя бы одного бита в данной последовательности исполняемый файл либо не работает, либо работает некорректно.

По определенному правилу из некоторого критического участка исполняемого файла выбирается последовательность k байт

$$\beta_i = (\beta_{i1}\beta_{i2} \dots \beta_{ij} \dots \beta_{i8}),\ i = [1..k]. \qquad (2)$$

К каждому набору β_i применяется функция Y_m, которая удаляет символ с заданным номером m, выбираемый на основе идентификатора автора:

$$\beta_i' = Y_m(\beta_i) = (\beta_{i1} \dots \beta_{i\,m-1}\,\beta_{i\,m+1} \dots \beta_{i8}),\quad m \in [1..8]. \qquad (3)$$

К каждому набору β_i' применяется функция I_j, которая для заданного j, выбираемого на основе идентификатора автора, преобразует бит β_{ij} по следующему правилу:

$$\beta_{ij} = 0, \text{ если } \beta_{ij} = \xi_i,$$
$$\beta_{ij} = 1, \text{ если } \beta_{ij} \neq \xi_i, \qquad (4)$$

где $\xi_i \in \xi$.

Получаем $I_j(\beta_i') = \beta_i''$, $i = [1..k]$, $j \in [1..7]$. Заметим, что в результате применения функции I_j наборы β_i' и β_i'' могут различаться только в символе с номером j. Длина набора β_i'' равна 7.

Далее для каждого β_i'' строится код Хемминга: последовательность β_i'' кодируется набором

$$\chi_i = (\chi_{i1}\chi_{i2}\beta_{i1}\chi_{i3}\beta_{i2}\beta_{i3}\beta_{i4}\chi_{i4}\beta_{i5}\beta_{i6}\beta_{i7}), \qquad (5)$$

где $\chi_{i1}, \chi_{i2}, \chi_{i3}, \chi_{i4}$ – контрольные биты; $\beta_{i1}, \beta_{i2}, \beta_{i3}, \beta_{i4}, \beta_{i5}, \beta_{i6}, \beta_{i7}$ – биты последовательности β_i''.

Так как значения информационных битов последовательности χ_i известны (в силу выбора набора β_i), то для кодирования набора $\xi = (\xi_1\xi_2 \dots \xi_k)$ будем использовать последовательность, состоящую только из контрольных битов кода Хемминга, построенного для каждого ξ_i:

$$\xi \to \chi_\xi = (\chi_{11}\chi_{12}\chi_{13}\chi_{14} \dots \chi_{i1}\chi_{i2}\chi_{i3}\chi_{i4} \dots \chi_{k1}\chi_{k2}\chi_{k3}\chi_{k4}). \qquad (6)$$

Таким образом, каждая тетрада последовательности χ_ξ контролирует соответствующий бит идентификационной метки. Заметим, что длина кода χ_ξ составляет $4k$ битов.

Утверждение. Код, построенный в соответствии с правилами (2)–(6), является самокорректирующимся.

Доказательство утверждения следует из построения кода. Действительно, из построения β_i'' следует, что при изменении значения бита ξ_i в представлении метки в последовательности β_i'' может измениться только

This document has been edited with Infix PDF Editor - free for non-commercial use.
To remove this notice, visit: www.iceni.com/unlock.htm

один бит, а код Хемминга, построенный для такой последовательности, позволяет обнаруживать и исправлять ошибку.

Идентификационные метки, внедряемые в исполняемый файл программы, строятся на основе наборов ξ, ψ, η, χ_ξ, χ_ψ, χ_η и могут быть представлены в следующем виде:

$$M_1 = (\xi\ \chi_\psi\ \chi_\eta);\ M_2 = (\chi_\xi\ \psi\ \chi_\eta);\ M_3 = (\chi_\xi\ \chi_\psi\ \eta).$$

Длина каждой метки составляет $9k$ битов.

Для внедрения самовосстанавливающихся меток используются оригинальные стеганографические методы включения информации в исполняемые файлы.

Применение автоматизированной системы идентификации программ

Предлагаемая система может быть использована, во-первых, в качестве практического средства для предварительной защиты авторских прав разработчиков программного обеспечения и, во-вторых, в качестве программного средства эксперта-криминалиста, специализирующегося по делам о компьютерных преступлениях и, в частности, связанным с нарушениями авторских прав создателей программного обеспечения.

1. Об авторском праве и смежных правах: Закон Республики Беларусь от 16 мая 1996 г. № 370-XIII // Ведомости Верховного Совета Республики Беларусь. 1996. № 20. С. 366; Ведомости Национального Собрания Республики Беларусь. 1998. № 31–32. С. 472; Изменения и дополнения: Закон от 11 августа 1998 г. № 194-З // Ведомости Национального Собрания Республики Беларусь. 1998. № 31–32. С. 472.

2. Яблонский С.В. Введение в дискретную математику. М., 1979.

Поступила в редакцию 06.04.2001.

Михаил Константинович Буза – доктор технических наук, профессор, заведующий кафедрой математического обеспечения ЭВМ.

Елена Николаевна Ливак – старший преподаватель кафедры информатики и вычислительной техники ГрГУ им. Я. Купалы.

УДК 519.10

Г.В. МАТВЕЕВ, В.М. ШИРЯЕВ

ОЦЕНКИ ДЛЯ ЧИСЛА НЕРАВЕНСТВ

Some new results on the number of unequatityes for arbitrary systems are obtained. Our approach is based on the arithmetical theory of quadratic forms.

При изучении подмножеств, свободных от сумм в абелевых группах [1], возникает следующая задача комбинаторного характера. Пусть $a_1, a_2, \ldots, a_m$ – упорядоченная система элементов конечного множества. Под неравенством в этой системе будем понимать бинарное соотношение $a_i \neq a_j$. При этом неравенства $a_i \neq a_j$ и $a_j \neq a_i$ не различаются. Систему $a_1, a_2, \ldots, a_m$ разобьем на классы одинаковых элементов $\{a_1, a_2, \ldots, a_m\} = X_1 \cup X_2 \cup \ldots \cup X_t$. Положим $|X_i| = x_i$, $i = 1, 2, \ldots, t$. Очевидно $x_1 + x_2 + \ldots + x_t = m$. Обозначим через n количество неравенств в системе.

Задача состоит в описании условий, при которых тройка натуральных чисел выступает в качестве системы параметров m, n, t.

Необходимым и достаточным условием реализуемости является разрешимость в целых положительных числах системы уравнений

This document has been edited with Infix PDF Editor - free for non-commercial use.

To remove this notice, visit: www.iceni.com/unlock.htm