Построение хрупкого цифрового водяного знака, применяемого коллективом авторов в исполняемых файлах

И. В. Нечта

*Сибирский государственный университет телекоммуникаций и информатики
Новосибирск, Россия*

Аннотация

Согласно данным статистических исследований, нарушение лицензионных соглашений ежегодно приносит огромные убытки компаниям, занимающимся разработкой программного обеспечения. С одной стороны, создаются нелегальные копии программного продукта, с другой стороны, несанкционированно используются какие-либо фрагменты программ третьими лицами. Еще одной важной проблемой является нарушение целостности программы, например в части блокировки функций проверки лицензионного ключа. В этой связи задача разработки методов защиты интеллектуальной собственности в программных приложениях является весьма актуальной. Известные ранее методы решают данную задачу при помощи хрупких водяных знаков. Большое внимание исследователи уделяют способам внедрения в программу цифровых водяных знаков (ЦВЗ) и методам их построения. Ниже предлагается метод построения хрупкого цифрового водяного знака, применяемого в исполняемых файлах. Рассматривается модель коллектива разработчиков, создающих защищенный ЦВЗ программный продукт. Применение данного метода позволит выявить факт нарушения целостности контейнера и позволит автору при необходимости подтвердить свое участие в создании и внедрении ЦВЗ. В данном методе используются математические свойства систем линейных алгебраических уравнений, электронно-цифровая подпись и криптографические хэш-функции. Схема базируется на теореме Кронекера – Капелли. Для нахождения группового пароля соавтор, входящий в группу, находит один из корней совместной системы линейных алгебраических уравнений. Указанная система состоит из n уравнений и содержит $n + 1$ неизвестную. Для постороннего, не входящего в группу, система уравнений не имеет единственного решения. Соавтор, входящий в группу, способен вычислить одну неизвестную системы на основе своих паспортных данных. Следовательно, система уравнений для такого соавтора имеет единственное решение. Проанализированы различные атаки на защищенную таким образом программу, показана эффективность предлагаемого метода. Разработанный метод может применяться в компаниях с большим коллективом разработчиков.

Ключевые слова

хрупкий цифровой водяной знак, стеганография, защита авторских прав, контроль целостности программ, коллективное авторское право, нарушение авторских прав, теорема Кронекера – Капелли, электронная цифровая подпись, защищенный программный продукт

Для цитирования

Нечта И. В. Построение хрупкого цифрового водяного знака, применяемого коллективом авторов в исполняемых файлах // Вестник НГУ. Серия: Информационные технологии. 2020. Т. 18, № 1. С. 65–73. DOI 10.25205/1818-7900-2020-18-1-65-73

Construction Fragile Digital Watermark Used by a Team of Authors in Executable Files

I. V. Nechta

*Siberian State University of Telecommunications and Information Sciences
Novosibirsk, Russian Federation*

Annotation

According to statistical research, a violation of license agreements annually causes huge losses to software companies. On the one hand, illegal copies of the software product are created, on the other hand, some fragments of the programs are used by third parties unauthorized. Another important problem is the violation of the program integrity, for example, in terms of blocking functions of the license key checking. In this regard, the task of construction methods for

protecting intellectual property in software applications is highly relevant. Previously known methods solve this problem by means of fragile digital watermarks. Below is presented a method for constructing a fragile digital watermark used in executable files. A model of a developers team creating software product protected by DWM is considered. The application of this method will allow to reveal the fact of the container integrity violation, on the one hand, and, on the other hand, will allow the author, if it is necessary, to confirm his participation in the development and embedding of the DWM. In this method we use mathematical properties of systems of linear algebraic equations, digital signature and cryptographic hash functions. The scheme is based on the Kronecker – Capelli theorem. To find the group password the co-author who is in the group finds one root of the consistent system of linear algebraic equations. The indicated system consists of n equations and contains $n + 1$ variables. For an outsider who is not in the group, the system of equations does not have a unique solution. The co-author of the group is able to calculate one variable in system based on their passport data. Therefore, the system of equations for such co-author has a unique solution. Various attacks on a protected by the new method are analyzed, and it is shown its efficiency. The constructed method can be applied in companies with a large team of developers.

Keywords

fragile digital watermark, steganography, copyright protection, program integrity control, collective copyright, violation of copyright rights, Kronecker – Capelli theorem, electronic digital signature, secure software product

For citation

Nechta I. V. Construction Fragile Digital Watermark Used by a Team of Authors in Executable Files. *Vestnik NSU. Series: Information Technologies*, 2020, vol. 18, no. 1, p. 65–73. (in Russ.) DOI 10.25205/1818-7900-2020-18-1-65-73

Введение

При создании и сопровождении цифрового контента одной из важных проблем является защита авторского права. Известно, что коммерческие организации несут значительные убытки вследствие несанкционированного копирования, распространения и неправомерного заимствования защищаемого цифрового объекта. Так, например, изображение, сделанное одним фотографом, может использоваться другим фотографом как собственное. Другим примером является незаконное использование фрагментов чужих программных приложений, например, библиотек.

Развитие стеганографии – науки, изучающей проблемы сокрытия факта передачи секретного сообщения, привело к созданию методов защиты объектов авторского права. Так, в файл может быть внедрено скрытое секретное сообщение, которое предназначено для идентификации автора. Если данный файл будет незаконно использоваться посторонним лицом, то настоящий автор может извлечь сообщение и продемонстрировать его в ходе судебных разбирательств. Таким образом будет установлено истинное авторство.

Встраиваемые в контейнер данные могут быть различными и зависят от решаемой задачи. В научной литературе существует два термина: *цифровой отпечаток пальца* (ЦОП) и *цифровой водяной знак* (ЦВЗ). ЦОП используется для идентификации владельца файла. Например, в каждую продаваемую копию программы автор встраивает паспортные данные лицензионного пользователя. При обнаружении пиратской копии программы, сообщение извлекается, и выявляется нарушитель лицензионного соглашения. С точки зрения правоприменительной практики (западных стран) данный факт является основанием для возбуждения следственных действий в отношении нарушителя.

Понятие ЦВЗ является более общим и включает в себя решение задач идентификации автора [1] и контроля целостности объекта данных или его фрагментов [2]. В случае с программами контроль целостности позволяет конечному пользователю удостовериться, что он использует неповрежденные вредоносными программами приложения.

Одна из задач стеганографии – это скрыть факт передачи секретного сообщения. В случае с защитой авторских прав это уменьшает вероятность успешной атаки злоумышленником, который явно не знает места размещения атакуемых данных. Поэтому сообщение внедряют в безобидный на вид объект данных, так называемый контейнер. В качестве контейнера могут быть текстовые, аудио-, видео- и исполняемые файлы. Каждый тип контейнера имеет свои свойства: вероятность обнаружения секретного сообщения, объем внедрения, устойчивость

внедрения к искажению контейнера и т. д. В силу большого внимания со стороны исследователей к стеганографии указанные свойства часто меняются при разработке новых методов стегоанализа (выявление факта внедрения).

На сегодняшний день существует большое число методов внедрения скрытых сообщений. Группа методов LSB внедрения, например, опубликованные в работах [3–6], применяются для встраивания сообщения в изображение и аудиоконтейнер. Используемый подход базируется на незначительном (с точки зрения восприятия человеком) изменении цвета изображения или звука. Например, запись сообщения производится в последний значащий бит цвета пикселя. Внедрение в текстовые контейнеры может осуществляться методом замены синонимов [7; 8]. Выбор синонима в предложении производится в соответствии со скрываемым сообщением. В данном случае меняется только форма записи смыслового содержания текста.

Для встраивания сообщений в исполняемые файлы используются статические [9; 10] и динамические цифровые водяные знаки [11; 12]. Статические ЦВЗ используют константы, различные формы записи команд (или групп команд) для хранения водяного знака. Однако такие методы не считаются устойчивыми к искажению. Например, исполняемый файл может быть упакован протектором [13], что сделает нечитаемым его водяной знак. Динамические водяные знаки создают некоторые динамические структуры данных в памяти, которые могут быть обнаружены по сигнатуре, известной только автору. Такие методы считаются устойчивыми не только к искажению, но и к стегоанализу.

Постановка задачи

Для повышения устойчивости внедрения в исполняемых файлах используются технологии запутывания кода программы (obfuscation) [14] и технологии контроля целостности (tamper proofing) [15]. Первая предполагает изменение программы так, чтобы устранить возможность анализа программы с целью выяснения логики ее работы. Вторая технология использует эквиваленты цифровой подписи защищаемых участков кода. Согласно определению, сформулированному в работе [2], ЦВЗ считается *хрупким*, если он разрушается при любом редактировании программы. Свойство хрупкости реализуется благодаря использованию хэш-функций, которые меняют свое значение при искажении кода программы даже на один бит. Наличие этих технологий в исполняемых файлах допускает снижение требований к ЦВЗ по устойчивости внедрения, так как устойчивость обеспечивается другими средствами.

Существуют ситуации, когда программное средство разрабатывается коллективом разработчиков. При этом возникает необходимость встроить ЦВЗ, содержащий информацию обо всех авторах. Для решения данной задачи существуют следующие подходы.

Метод Easter egg [16] заключается в создании дополнительного функционала программе. Например, при выборе пункта меню «Об авторах» отображается окно, содержащее список коллектива разработчиков. Такой подход прост в реализации, но является уязвимым. Злоумышленнику легче контролировать процесс искажения программы, так как в явном виде демонстрируются сведения, которые нужно исказить.

Другой подход предполагает использование «мертвого кода», определяющего водяной знак. Такой код встраивается в программу, но никогда не выполняется. Пример представлен на (см. рисунок).

Условие формируется из «непрозрачных предикатов» – логических выражений, которые всегда либо истинны, либо ложны и не могут быть легко проанализированы при дизассемблировании программы [17]. Таким образом, трудно отличить не выполнявшиеся полезные участки кода от мертвого. Недостатком такого метода является увеличение размера программы.

Как уже было отмечено, одним из важных моментов при построении ЦВЗ является вопрос о наполнении водяного знака. Обычно встраивают паспортные данные авторов или их фотографии. Известны методы, например [18], встраивающие значения хэш-функций участков кода.

Целью данной работы является создание метода построения ЦВЗ, применяемого коллективом авторов. Особое внимание уделяется содержимому ЦВЗ. Способ представления этого содержимого (в виде констант, графа, мертвого кода и т. д.) выходит за рамки данного исследования. Предполагается, что данный метод будет использоваться для исполняемого файла.

Рассматривается следующая модель. Пусть имеется организация, создающая программный продукт. Коллектив разработчиков формирует и публикует итоговый исполняемый файл, содержащий ЦВЗ, который включает в себя некоторые параметры и электронную цифровую подпись (ЭЦП) организации (нотариуса).

При необходимости любой автор может подтвердить свое участие в разработке программы при помощи извлечения и анализа ЦВЗ. Целью злоумышленника является либо устранение данных о каком-либо авторе из ЦВЗ, либо замена / добавление новых данных.

Сформулируем предъявляемые требования к ЦВЗ (далее – Требования). Часть Требований к цифровым водяным знакам была уже ранее сформулирована, например, в работе [9].

1. Возможность обеспечить проверку целостности контейнера.
2. Посторонний не может выдать себя за автора.
3. Невозможность произвольно исключать данные автора из ЦВЗ.

Другая часть Требований сформулирована в рамках данной статьи в соответствии с рассмотренной выше моделью.

4. Автор способен подтвердить свои права без раскрытия конфиденциальных сведений о других авторах. Раскрытие конфиденциальных сведений об авторах (например, паспортных данных) возможно только с их согласия, что не считается безопасным действием.

5. Возможность объединять авторов в группы. Данное требование адаптировано к современной практике при создании программ, когда авторы объединены в группы, ответственные за определенный круг задач. Следовательно, появляется возможность определить долю участия автора.

6. Каждая группа имеет возможность реализовать механизм проверки ЦВЗ внутри программы. Здесь предусматривается ситуация, когда злоумышленниками являются члены какой-либо группы и нотариус. При выполнении данного требования достаточно одной группы, не содержащей злоумышленника, для корректной проверки ЦВЗ.

Последний пункт Требований накладывает ограничение на используемый метод встраивания скрытых данных, который должен позволять беспрепятственное считывание ЦВЗ.

Описание предлагаемого метода

Проанализируем требования к ЦВЗ. Достижение поставленной ранее цели может быть осуществлено классической ЭЦП, однако подпись всегда поставляется отдельно от контейнера и требует хранения дополнительных сведений. В отличие от ЭЦП предлагаемый водяной знак позволяет объединять авторов в группы. Указанное свойство требуется для определения доли участия. Например, программа состоит из нескольких модулей, реализованных разными подразделениями организации. ЦВЗ позволяет сгруппировать пользователей по подразделениям.

Рассмотрим предлагаемый метод более подробно. Пусть имеется коллектив авторов из N человек и известно их разбиение на M групп. Обозначим i – номер группы ($i \in [1, M]$) и k_i – размер группы i . Каждому автору ставится в соответствие некоторый элемент $a_{i,j}$, где j – номер автора в i -й группе ($j \in [1, k_i]$). Значение $a_{i,j}$ вычисляется как хэш-функция от паспортных данных автора. Хэш-функция должна обладать свойствами криптографических хэш-функций.

Введем элемент f_i , необходимый для разделения групп. Знание значения f_i есть подтверждение факта вхождения в группу (в таком смысле это групповой пароль). Вычисление происходит следующим образом¹:

$$f_i = \sum_j a_{i,j}.$$

Для каждой группы, состоящей из нескольких авторов, формируется система линейных алгебраических уравнений:

$$\begin{cases} f_i \oplus a_{i,j} = B_{i,j} \\ a_{i,j} \oplus a_{i,j+1} = B_{i,j+1} \\ \dots \\ a_{i,k-1} \oplus a_{i,k} = B_{i,k} \end{cases}$$

Для каждой группы, состоящей из одного автора, формируется уравнение, содержащее a_{null} , рассчитанное для заранее оговоренных паспортных данных (например, нулевые значения серии и номера):

$$a_{i,j} \oplus a_{null} = B_{i,j}.$$

Рассмотрим для примера разбиение на группы:

$$A = \{[a_{1,1}, a_{1,2}], [a_{2,1}, a_{2,2}, a_{2,3}], [a_{3,1}, a_{3,2}, a_{3,3}, a_{3,4}]\}.$$

Получаются следующие системы уравнений:

$$\begin{cases} f_1 \oplus a_{1,1} = B_{1,1} \\ a_{1,1} \oplus a_{1,2} = B_{1,2} \end{cases}$$

¹ Здесь и далее используется сложение по модулю 2.

$$\begin{cases} f_2 \oplus a_{2,1} = B_{2,1} \\ a_{2,1} \oplus a_{2,2} = B_{2,2} \\ a_{2,2} \oplus a_{2,3} = B_{2,3} \end{cases}$$

$$\begin{cases} f_3 \oplus a_{3,1} = B_{3,1} \\ a_{3,1} \oplus a_{3,2} = B_{3,2} \\ a_{3,2} \oplus a_{3,3} = B_{3,3} \\ a_{3,3} \oplus a_{3,4} = B_{3,4} \end{cases}$$

Обозначим H – значение хэш-функции от пустого контейнера, введем $Sign(x)$ – электронную цифровую подпись сообщения x . Предполагается, что ЭЦП использует ключ нотариуса. Конкретизация ЭЦП, которая должна удовлетворять требованиям ГОСТ, выходит за рамки исследования.

Опишем процесс формирования ЦВЗ.

Шаг 1. Нотариус формирует ЭЦП $Sign(H \oplus T)$, где $T = \sum_i B_{i,j}$.

Шаг 2. Последовательно составляются значения: ЭЦП, необходимых данных для ее проверки (например, для RSA – открытая часть ключей) и множество значений $B_{i,j}$ из систем уравнений.

Шаг 3. Полученные значения внедряют в исполняемый файл.

На этапе составления систем уравнений все расчеты доступны внутри группы, следовательно, автор должен запомнить позицию начала собственной группы и позицию собственных данных внутри группы. Для предотвращения добавления ложного автора нотариусом необходимо контролировать число элементов B с помощью средств, реализованных внутри программы.

Рассмотрим действия, необходимые для подтверждения авторства.

Шаг 1. Извлечение ЦВЗ.

Шаг 2. Расчет значения хэш-функции от пустого контейнера. Ранее отмечалось ограничение в выбираемых методах внедрения. Для статических и динамических ЦВЗ пустой контейнер получается путем перезаписи ЦВЗ нулевыми значениями, при этом нужно только знать места его хранения в программе. Это требование не является проблемой для разработчиков программы.

Шаг 3. Вычисление значения $a_{i,j}^*$ на основе собственных паспортных данных и остальных неизвестных $(a_{i,j} : a_{i,j} \neq a_{i,j}^*)$ в системе уравнений соответствующей группы.

Шаг 4. Проверяется корректность подписи $Sign(H \oplus T)$ и числа значений B , если ошибки нет, то переходим к Шагу 5. В противном случае считаем, что контейнер был искажен и авторство не подтверждается.

Шаг 5. Если для группы выполняется условие $\sum a_{i,j} = f_i$, то авторство признается подлинным, иначе ложным.

Анализ выполнения Требований

Проанализируем выполнение предложенным ЦВЗ Требований (в том же порядке).

1. Проверка целостности контейнера осуществляется при помощи вычисления хэш-функции от пустого контейнера и сравнения полученного значения с эталоном. Известно, что хэш-функция изменит свое значение даже при незначительных изменениях в контейнере, в том числе одного бита, что обеспечивает контроль целостности.

2. Посторонний злоумышленник не сможет выдать себя за автора, так как значение хэш-функции от его паспортных данных будет отличаться от используемых в ЦВЗ. Рассмотрим подробнее действия злоумышленника. Предположим, что злоумышленник не входит в состав авторов и хочет предъявить свои права на содержимое контейнера. Тогда ему необходимо указать группу, в которой находятся его данные. На основе паспортных данных злоумышленника вычисляется значение a_{calc} и остальные переменные соответствующей системы уравнения. Использование в методе криптографической хэш-функций делает маловероятным коллизии (совпадений значений хэшей) при различных паспортных данных.

Если группа состоит из 1 человека, то рассчитанное значение B не будет совпадать с имеющимся в ЦВЗ, так как $(a_{i,1} \oplus a_{null}) \neq (a_{злой} \oplus a_{null})$. Если группа состоит из нескольких человек, то злоумышленник не сможет назвать значение f_i , доказывающее принадлежность к группе. Действительно, для нахождения f_i необходимо решить систему уравнений. Уравнения составляются так, что число неизвестных больше на единицу, чем количество уравнений. В таком случае согласно теореме Кронекера – Капелли система не имеет единственного решения. Таким образом, атака постороннего не будет успешной.

3. Произвольное исключение или изменение данных об авторе из ЦВЗ невозможно, так как наличие и корректность этих значений контролируются программой. Рассмотрим случай, когда злоумышленником является один из авторов. Его целью является исключение данных кого-либо из авторов («жертвы») из водяного знака. Допустим, злоумышленник решает обнулить сведения жертвы. Тогда меняются соответствующие значения $B_{i,j}$ и T , следовательно, подпись $Sign(H \oplus T)$ становится неверной. Подделать подпись злоумышленник не может. При выявлении факта разрушения ЦВЗ программа может становиться неработоспособной (например, генерируется программное исключение), что недопустимо для злоумышленника.

Предположим, что нотариус является злоумышленником, тогда он может попытаться либо добавить, либо заменить, либо удалить кого-либо из авторов. Добавление или удаление данных как одного нового автора, так и группы, в ЦВЗ вызовет изменение числа элементов $B_{i,j}$, что отслеживается. Замена данных о каком-либо авторе, во-первых, не дает знания группового пароля f_i , а во-вторых, приведет к изменению значений $B_{i,j}$ в соответствующей системе уравнений. И даже при правильной ЭЦП будет отслежено участниками группы.

Предположим, что злоумышленниками являются один член группы авторов и нотариус. Тогда злоумышленники могут заменить данные какого-либо автора в группе без нарушения ЭЦП. Однако в таком случае также меняется значение группового пароля, что отслеживается средствами контроля целостности ЦВЗ.

4. При доказательстве авторства необходимы только конфиденциальные сведения заявителя. Остальные элементы уравнения (из которого строится ЦВЗ) представляют собой хэш-значения и не раскрывают данных других авторов.

5. Метод позволяет добавлять сведения об авторах, объединенных в группы, как из одного человека, так и из нескольких.

6. Каждая группа имеет возможность реализовать механизм проверки ЦВЗ внутри программы. Действительно, разрабатываемый группой авторов код может содержать в себе функции проверки. Расчет используемых хэш-значений и составление уравнений не является трудной задачей для авторов, чьи данные встраиваются в ЦВЗ.

Таким образом, мы видим, что все сформулированные пункты Требований выполняются.

Заключение

Предложенный в данной работе метод позволяет создавать ЦВЗ, применяемый коллективом авторов, с возможностью подтверждения авторства над исполняемым файлом. Предложенная схема базируется на математических свойствах систем линейных алгебраических уравнений. Применение криптографических хэш-функций и ЭЦП реализует свойство хрупкости водяного знака, что позволяет контролировать целостность программы.

Предложенный метод обладает следующими особенностями. Во-первых, требуется хранение меньшего объема информации в отличие от схемы с ЭЦП. Во-вторых, не требуется регистрация и обслуживание ЭЦП для каждого автора в установленном порядке. В-третьих, метод позволяет объединять авторов в группы, что может быть полезным, например, для установления доли участия автора при разработке программы.

В работе были рассмотрены различные атаки злоумышленника. Показано, что, несмотря на активные действия со стороны злоумышленника, Требования, предъявляемые к ЦВЗ, выполняются. Предложенный подход может применяться при создании систем защиты авторского права.

Список литературы / References

1. **Dey A., Bhattacharya S., Chaki N.** Software watermarking: Progress and challenges. *INAE Letters*, 2019, vol. 4, no. 1, p. 65–75.
2. **Chen Z., Wang Z., Jia C.** Semantic-integrated software watermarking with tamper-proofing. *Multimedia Tools and Applications*, 2018, vol. 77, no. 9, p. 11159–11178.
3. **Zhou X., Gong W., Fu W., Jin L.** An improved method for LSB based color image steganography combined with cryptography. In: *IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, 2016, p. 1–4.
4. **Muhammad K., Ahmad J., Rehman N., Jan Z., Sajjad M.** CISSKA-LSB: color image steganography using stego key-directed adaptive LSB substitution method. *Multimedia Tools and Applications*, 2017, vol. 76, no. 6, p. 8597–8626.
5. **Liu J., Tian Y., Han T., Yang C., Liu W.** LSB steganographic payload location for JPEG-decompressed images. *Digital Signal Processing*, 2015, vol. 38, p. 66–76.
6. **Atoum M. S.** A Comparative Study of Combination with Different LSB Techniques in MP3 Steganography. *Information Science and Applications*, 2015, vol. 339, p. 551–560.
7. **Xiang L., Li Y., Hao W., Yang P., Shen X.** Reversible natural language watermarking using synonym substitution and arithmetic coding. *Computers, Materials and Continua*, 2018, vol. 55, no. 3, p. 541–559.
8. **Xiang L., Wang X., Yang C., Liu P.** A novel linguistic steganography based on synonym run-length encoding. *IEICE transactions on Information and Systems*, 2017, vol. 100, no. 2, p. 313–322.
9. **Collberg C., Sahoo T. R.** Software watermarking in the frequency domain: implementation, analysis, and attacks. *Journal of Computer Security*, 2005, vol. 13, no. 5, p. 721–755.
10. **Gong D., Liu F., Lu B., Wang P., Ding L.** Hiding information in java class file. *International Symposium on Computer Science and Computational Technology*, 2008, vol. 2, p. 160–164.
11. **Ma H., Jia C., Li S., Zheng W., Wu D.** Xmark: Dynamic Software Watermarking using Collatz Conjecture. *IEEE Transactions on Information Forensics and Security*, 2019, vol. 14, no. 11, p. 2859–2874.
12. **Wang Y., Gong D., Lu B., Xiang F., Liu F.** Exception handling-based dynamic software watermarking. *IEEE Access*, 2018, vol. 6, p. 8882–8889.

13. **Hai N. M., Ogawa M., Tho Q. T.** Packer identification based on metadata signature. In: Proceedings of the 7th Software Security, Protection, and Reverse Engineering / Software Security and Protection Workshop, 2017, p. 1–11. DOI 10.1145/3151137.3160687
14. **Behera C. K., Bhaskari D. L.** Different obfuscation techniques for code protection. *Procedia Computer Science*, 2015, vol. 70, p. 757–763.
15. **Ahmadvand M., Pretschner A., Kelbert F.** A taxonomy of software integrity protection techniques. *Advances in Computers*, 2019, vol. 112, p. 413–486.
16. **Nematollahi M. A., Vorakulpipat C., Rosales H. G.** Software Watermarking. *Digital Watermarking*, 2017, vol. 11, p. 131–144.
17. **Xu D., Ming J., Wu D.** Generalized dynamic opaque predicates: A new control flow obfuscation method. *International Conference on Information Security*, 2016, vol. 9866, p. 323–342.
18. **Ahmadvand M., Hayrapetyan A., Banescu S., Pretschner A.** Practical Integrity Protection with Oblivious Hashing. In: Proceedings of the 34th Annual Computer Security Applications Conference, 2018, p. 40–52.

*Материал поступил в редакцию
Received
08.11.2019*

Сведения об авторе

Нечта Иван Васильевич, кандидат технических наук, доцент, начальник отдела подготовки кадров высшей квалификации, Сибирский государственный университет телекоммуникаций и информатики (Новосибирск, Россия)
ivannechta@gmail.com
ORCID 0000-0003-0361-2742
Researcher ID B-1604-2014

Information about Author

Ivan V. Nechta, PhD (Eng), Head of the department of training highly qualified personal Siberian State University of Telecommunications and Information Sciences (Novosibirsk, Russian Federation)
ivannechta@gmail.com
ORCID 0000-0003-0361-2742
ResearcherID B-1604-2014