УДК 004.056.53

А.В. Козачок, Л.М. Туан

Подход к защите файлов документальных форматов от несанкционированного доступа на основе применения неразличимой обфускации программного кода

Предложен подход к защите файлов документальных форматов от несанкционированного доступа на основе применения формата защищенного контейнера. Приведено описание функциональной модели системы контролируемого разграничения доступа к данным, отличающейся применением процедуры неразличимой обфускации.

Ключевые слова: защита от несанкционированного доступа, неразличимая обфускация, контейнер, исполняемый код.

В настоящее время угрозы информационной безопасности, связанные с утечкой конфиденциальных данных, являются одними из наиболее опасных для любой организации, так как приводят к материальному ущербу, утрате интеллектуальной собственности, снижению репутации организации. По мере развития информационных технологий и предоставляемых услуг связи число потенциальных каналов утечки информации растет [1].

Целью проводимого исследования является построение комплекса моделей процесса контролируемого разграничения доступа к данным, позволяющего осуществить защиту от несанкционированного доступа к информации за счет применения неразличимой обфускации программного кода [2].

Исходя из вышеизложенного, разработана обобщенная функциональная модель процесса контролируемого разграничения доступа к данным на основе обфускации программного кода. При этом субъектами доступа в модели выступают пользователи, идентифицируемые учетными записями, а объектами являются файлы документальных форматов. Правила разграничения доступа субъектов к объектам задаются в виде матрицы доступа, учитывающей метки конфиденциальности.

В основу предлагаемой модели положен подход схожий с моделью системы военных сообщений [3], когда используется понятие контейнера для обработки структурированных данных. Отличительная особенность заключается в том, что контейнер является защищенным на основе метода неразличимой обфускации [4].

Обфускацией программы называется всякое ее преобразование, которое сохраняет вычисляемую программой функцию, но при этом придает программе такую форму, что извлечение из текста программного кода ключевой информации об алгоритмах и структурах данных, реализованных в этой программе, становится трудоемкой задачей.

Обфусцированной программой называется программа, которая после применения обфусцирующих преобразований на всех допустимых для исходной программы входных данных выдает тот же самый результат, что и оригинальная программа, но более трудна для анализа, понимания и модификации [5].

В настоящее время исследования в области обфускации программного кода проводятся по двум направлениям [4]:

– системное программирование;

– математическая криптография.

С позиции системного программирования обфускация программы может использоваться для защиты авторских прав на программное обеспечение, для предотвращения реверс-инжиниринга программ, для создания и защиты водяных знаков, обеспечения безопасности мобильных агентов в информационных сетях, для проведения безопасного поиска в потоках данных и защиты баз данных. Однако существенным недостатком данного подхода является отсутствие обоснования гарантированной стойкости. В случае применения методов динамического анализа программ и привлечения квалифицированных экспертов в области системного программирования стойкости существующих средств обфускации программ оказывается недостаточно.

С позиции математической криптографии разработка эффективных алгоритмов позволит решить целый ряд серьезных вопросов, например, с ее помощью можно преобразовать криптосистему с секретным ключом к криптосистеме с открытым ключом, проводить вычисления над зашифрованными данными, реализовывать системы функционального шифрования, доверенные схемы перешифрования и электронно-цифровой подписи, создавать верифицируемые системы тайного голосования и схемы двойственного шифрования.

Для построения эффективного метода защиты файлов документальных форматов, внедренных в защищенный контейнер, предлагается использовать математический аппарат неразличимой обфускации программного кода, активно развивающийся в настоящее время в рамках направления математической криптографии [6]. Исследования в области неразличимой обфускации, проводимые в настоящее время, как российскими учеными (Н.П. Варнавский, В.А. Захаров, Н.Н. Кузюрин), так и зарубежными (S. Garg, C. Gentry, S. Halevi, B. Barak, J.S. Coron, T. Lepoint, M. Tibouchi), базируются на возможности обфускации булевых функций. Процедуру проверки прав доступа пользователя к документу, внедренному в контейнер, можно рассматривать как точечную функцию, поскольку результатом ее выполнения является значение из множества {0,1}, поэтому применение неразличимой обфускации для защиты данной проверки является также корректным. Исследование [7] посвящено модификации существующих подходов к осуществлению неразличимой обфускации с целью устранения ряда ограничений, обусловленных применяемыми механизмами, моделями и алгоритмами [8, 9], а также обоснованию возможности применения данного математического аппарата для решения задачи защиты от несанкционированного доступа.

Разработанная функциональная модель позволяет хранить данные в унифицированном виде вне зависимости от исходного формата файла и обеспечивает единый метод доступа к данным для всех типов. Для безопасного хранения данных используется формат защищенного контейнера, в котором данные хранятся в обфусцированном виде. Контейнер представляет собой исполняемый файл, обладающий рядом заданных свойств и функций, позволяющих однозначно идентифицировать пользователя, разграничивать доступ к данным (права: читать, писать, передать права), обеспечивать защиту конфиденциальности внедренного документа. Формат контейнера обеспечивает его безопасное хранение и передачу по сети.

В предлагаемой модели рассматриваются две базовые операции:

– создание нового документа;

– открытие существующего документа, хранящегося в автоматизированной системе.

Рассмотрение следует начать с первой операции, когда пользователю необходимо создать новый документ. На рис. 1 представлена функциональная модель процесса создания документа и внедрения его в защищенный контейнер.

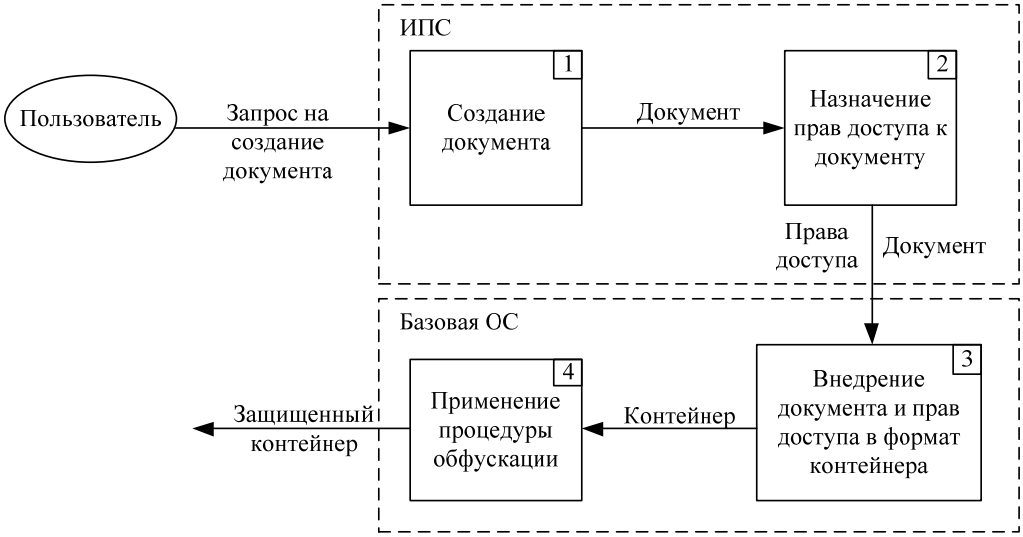

Рис. 1. Функциональная модель процесса создания документа и внедрения его в защищенный контейнер

Данная модель включает в себя следующие этапы:

1. Создание документа. На первом этапе осуществляется создание нового документа с использованием соответствующего приложения пользователя в базовой операционной системе (ОС).

2. Назначение прав доступа к документу. По окончании работы с приложением пользователю необходимо задать определенные права доступа к созданному документу. В модели предусмотрены четыре типа доступа: отсутствие прав, право на чтение, право на изменение (запись), право на передачу прав. Полные права на документ означают, что пользователь имеет полный доступ к документу, в том числе он может передавать права доступа другому пользователю в системе. Результатом выполнения данного этапа является документ с определенными правами на доступ к нему.

Важно отметить, что два вышеперечисленных этапа осуществляются в изолированной программной среде. Под изолированной программной средой (ИПС) в данном случае будем понимать изолированную среду выполнения, для которой реализована защита данных в оперативной памяти и на жестком диске, также исключается доступ в нее процессов, функционирующих в базовой операционной системе, тем самым обеспечивая конфиденциальность и целостность данных, обрабатываемых внутри ИПС.

3. Внедрение документа и прав доступа в формат контейнера. Как было обозначено ранее, контейнер представляет собой файл исполняемого формата, структура которого позволяет внедрять данные различных типов. Задачу формирования контейнера выполняет специализированное программное обеспечение, с помощью которого создается шаблон исполняемого файла в двоичном формате, к которому затем добавляются созданный документ и матрица доступа.

4. Применение процедуры обфускации. Для обеспечения эффективного метода защиты от анализа и модификации созданного контейнера, а также защиты конфиденциальности и целостности данных в работе использовалась процедура неразличимой обфускации исполняемого кода контейнера. Выходом данного блока является защищенный контейнер, устойчивый к различным методам статического и динамического анализа. Формат контейнера разрешает его передачу по сети и хранение на различных сетевых ресурсах.

Вторая операция, осуществляемая в рамках модели, – открытие пользователем существующего документа, который хранится в формате защищенного контейнера в автоматизированной системе. Функциональная модель доступа к защищенному контейнеру приведена на рис. 2. Для доступа и работы с защищенными контейнерами необходимо наличие в операционной системе специализированного программного обеспечения.

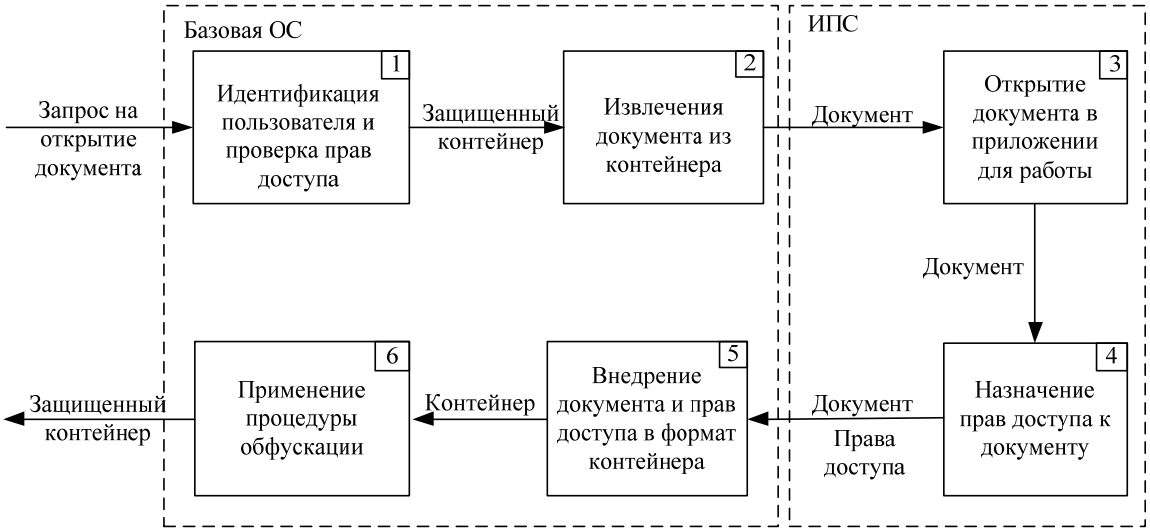

Рис. 2. Функциональная модель доступа к защищенному контейнеру

Данная модель включает в себя следующие этапы:

1. Идентификация пользователя и проверка прав доступа. Поскольку исполняемый файл, выбранный в качестве основы контейнера, является активным элементом, в его коде реализуются функции по идентификации пользователя. Затем в случае успешной идентификации осуществляется проверка прав на доступ к контейнеру идентифицированного пользователя на основе внедренной в контейнер матрицы доступа.

2. Извлечение документа из контейнера. При успешной идентификации пользователя и проверке прав доступа документ извлекается из защищенного контейнера в область памяти, защищенную ИПС.

3. Открытие документа в приложении для работы. После извлечения документа осуществляется запуск приложения пользователя для работы с документом в ИПС.

4. Назначение прав доступа к документу. Данный этап осуществляется только в том случае, если по окончании работы пользователь, имеющий полные права доступа к документу, принимает решение о назначении или передаче прав доступа другому пользователю.

Документ, которому были назначены права доступа, инкапсулируется в формат защищенного контейнера, а затем применяется процедура неразличимой обфускации к данному контейнеру для защиты от несанкционированного доступа и анализа. Полученный обфусцированный контейнер готов к передаче и дальнейшей работе с ним.

На рис. 3 представлена обобщенная модель процесса контролируемого разграничения доступа к данным на основе обфускации программного кода. Она включает в себя совокупность операций по созданию нового документа и открытию существующего.

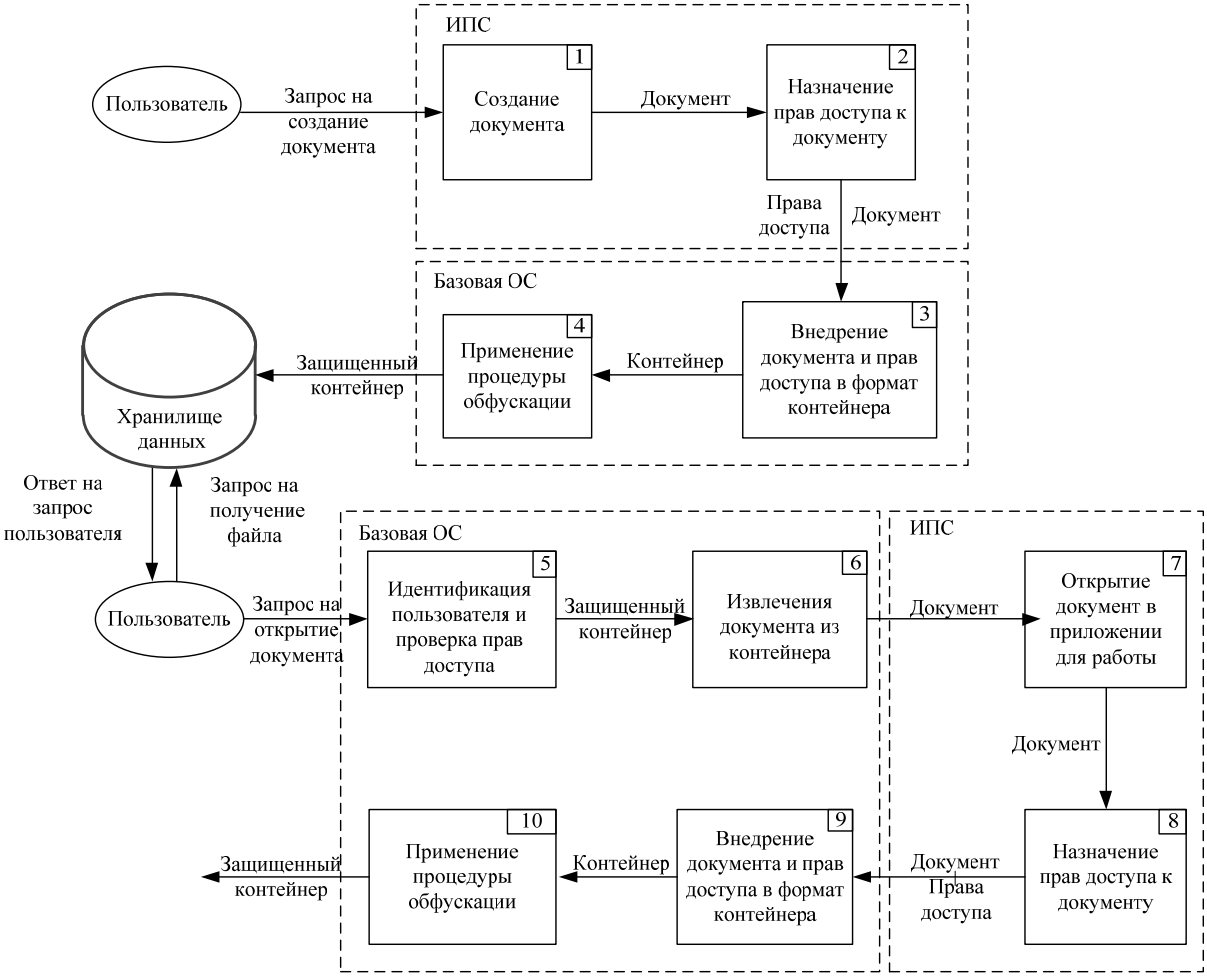

Рис. 3. Обобщенная модель процесса контролируемого разграничения доступа к данным на основе обфускации программного кода

В качестве примера и для рассмотрения механизма работы предложенных моделей были выбраны следующие исходные данные: пользователи User_1, User_2, …, User_N; группа пользователей CoWorkers.

Процесс контролируемого разграничения доступа к данным и обеспечения конфиденциальности документов на основе неразличимой обфускации программного кода реализуется следующим образом.

Пользователь User_1 создает документ для отправки его пользователю User_2. Создание документа и назначение прав доступа происходит в ИПС, как было показано ранее. По окончании работы с документом пользователь назначает права доступа пользователю User_2. Права могут задаваться как для отдельных пользователей, так и для групп пользователей. Варианты задания права доступа в обобщенной модели представлены в таблице.

Из анализа таблицы можно сделать вывод, что данное представление вариантов доступа позволяет комбинировать мандатное и дискреционное разграничение доступа. Первая и третья строки таблицы представляют собой классический вариант дискреционных правил разграничения доступа. Вторая строка – вариант представления правила мандатного разграничения доступа.

При этом для пользователя User_2 были назначены полные права доступа, для группы пользователей CoWorkers – права только на чтение, а пользователям с меткой конфиденциальности «2» – права доступа на чтение и запись.

Пример вариантов задания прав доступа к документу

Метка конфиденциальности	Имя пользователя	Имя группы	Права доступа		
			Чтение	Запись	Полные
*	User_2	*	1	1	1
2	*	*	1	1	0
*	*	CoWorkers	1	0	0

После назначения прав доступа происходит процесс внедрения документа в контейнер. Для защиты конфиденциальности и целостности документа, находящегося внутри контейнера, запускается механизм обфускации содержимого полученного контейнера. Поле этого защищенный контейнер помещается в хранилище данных.

При получении документа из хранилища данных механизм обеспечения безопасного доступа работает по следующему алгоритму.

Пользователь User_2 запускает полученный контейнер на исполнение. Встроенный в исполняемый файл механизм идентификации осуществляет проверку прав пользователя User_2 на доступ к документу, инкапсулированному в контейнер.

При успешной идентификации и проверке прав доступа пользователь User_2 получает доступ к документу, извлеченному из контейнера в ИПС. В соответствии с таблицей для пользователя User_2 были назначены полные права на доступ к документу, поэтому по окончании работы он сможет переназначить права доступа. Затем осуществляется процесс внедрения документа в формат контейнера, применения обфускации и сохранения защищенного контейнера.

При этом защищенный контейнер помещается в хранилище данных. При запросе на открытие защищенного контейнера, встроенным механизмом защиты считывается идентификатор пользователя, осуществляется идентификация, анализируются заданные правила и принимается решение о возможности открытия контейнера. По окончании работы осуществляются все действия, аналогичные предыдущим примерам.

Таким образом, исходя из представленного комплекса моделей, можно сделать вывод, что применение подхода к защите файлов документальных форматов на основе применения неразличимой обфускации программного кода позволит добиться, с учетом ряда ограничений, предотвращения возможности несанкционированного доступа, а тем самым возможности нарушения конфиденциальности и целостности обрабатываемой в автоматизированной системе информации.

Литература

1. Козачок В.И. Факторы определяющие информационную безопасность корпорации / В.И. Козачок, С.А. Власова // Среднерусский вестник общественных наук. – 2014. – Вып. 5(35). – С. 30–34.

2. Козачок А.В. Обоснование возможности применения неразличимой обфускации для защиты исполняемых файлов / А.В. Козачок, Л.М. Туан // Перспективные информационные технологии: сб. трудов междунар. науч.-техн. конф.– Самара, 2015. – Т. 1. – С. 269–272.

3. Девянин П.Н. Модели безопасности компьютерных систем: учеб. пособие / П.Н. Девянин. – М.: Изд. центр «Академия», 2005. – 144 с.

4. Современное состояние исследований в области обфускации программ: определения стойкости обфускации / Н.П. Варновский, В.А. Захаров, Н.Н. Кузюрин, А.А. Шокуров // Труды Института системного программирования. – М.: ИСП РАН, 2014. – Т. 26, № 3. – С. 167–198.

5. Козачок А.В. Комплекс алгоритмов контролируемого разграничения доступа к данным, обеспечивающий защиту от несанкционированного доступа / А.В. Козачок, Л.М. Туан // Системы управления и информационные технологии. – Воронеж, 2015. – № 3(61). – С. 58–61.

6. Candidate indistinguishability obfuscation and functional encryption for all circuits / S. Garg, C. Gentry, S. Halevi et al. // IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS), 2013. – Berkley, USA: IEEE, 2013. – P. 40–49.

7. Аналитическая модель защиты файлов документальных форматов от несанкционированного доступа / А.В. Козачок, М.В. Бочков, Р.Р. Фаткиева, Л.М. Туан // Труды СПИИРАН. СПб., 2015. – Вып. 43. – С. 228–252.

8. Варновский Н.П. Математические проблемы обфускации / Н.П. Варновский, В.А. Захаров, Н.Н. Кузюрин // Математика и безопасность информационных технологий: матер. конф. в МГУ 28–29 октября 2004 г. – М., 2005. – С. 65–91.

9. On the (im)possibility of program obfuscation / B. Barak, O. Goldreich, R. Impagliazzo et al. // Advances in Cryptology. Lecture Notes in Computer. – Science, Paris, 2001. – Vol. 2139. – P. 1–18.

Козачок Александр Васильевич
Канд. техн. наук, сотрудник Академии федеральной службы охраны РФ, Орёл
Тел.: +7 (486-2) 54-99-33
Эл. почта: totrin@mail.ru

Туан Лай Минь
Сотрудник Академии федеральной службы охраны РФ, Орёл
Тел.: +7 (486-2) 54-99-33
Эл. почта: totrin@mail.ru

Kozachok A.V., Tuan L.M.

Approach to protect documentary file formats from unauthorized access based on indistinguishable program code obfuscation

The approach to protect documentary file formats from unauthorized access based on protected container format is described. Functional model aimed to control data access restriction system providing protection from unauthorized access and applying indistinguishable obfuscation is presented .

Keywords: protection from unauthorized access, indistinguishable obfuscation, container, program code.