УДК 004.087

АППАРАТНО-ПРОГРАММНЫЕ СРЕДСТВА И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ ЗАЩИТЫ СИСТЕМ ПРЕДОСТАВЛЕНИЯ ПОЛЬЗОВАТЕЛЯМ ДОСТУПА К ПРОГРАММНЫМ РЕСУРСАМ

Кравченко А.С. [1], Родин С.В. [2], Смоленцева Т.Е. [3]

[1] ФКОУ ВПО «Воронежский институт ФСИН России», Воронеж, Россия (394072, г. Воронеж, ул. Иркутская, 1 «а»), e-mail: vifsin@mail.ru
[2] ФГКОУ ВПО «Воронежский институт Министерства внутренних дел Российской Федерации», e-mail: rosvment@mail.ru
[3] ФГБОУ ВПО «Липецкий государственный педагогический университет», e-mail: kaf-inf@stu.lipetsk.ru

В статье рассматриваются приемы защиты от несанкционированного использования программы – подключаемые к портам вычислительной машины устройства (аппаратные ключи). В настоящее время аппаратные средства применяются для защиты дорогостоящего или специализированного программного обеспечения. Использование аппаратных ключей предполагает разработку приложений в соответствии с требованиями обеспечения безопасности. Наиболее действенными способами обхода защиты с применением аппаратных ключей является изменение кода защищаемой программы для удаления из алгоритма обращений к ключу или подделки его ответов на запросы, второй способ основан на программной симуляции наличия аппаратного устройства, при этом программа не модифицируется. Наиболее качественную защиту обеспечивают ключи с модифицируемым алгоритмом. Рассмотрены два варианта проверки не только наличия, но и соответствия ключа приложению: генерация электронной цифровой подписи и ее проверка, генерация массива ответов ключа на запросы программы по одному из алгоритмов шифрования. Аппаратные ключи в значительной степени усложняют процесс копирования и нелегального использования программного обеспечения, но наиболее эффективны при применении в комплексе с организационными методами защиты информации.

Ключевые слова: аппаратный ключ, защита программного обеспечения, ключ безопасности, ключ с секретным алгоритмом, ключ с памятью, ключ с модифицируемым алгоритмом.

HARDWARE AND SOFTWARE AND INFORMATION PROCESSES PROTECTION SYSTEMS GIVE USERS ACCESS TO SOFTWARE RESOURCES

Kravchenko A.S.[1], Rodin S.V.[2], Smolentseva T.E.[3]

[1] Voronezh Institute of the Russian Federal Penitentiary Service, Voronezh, Russia (394072, Voronezh, street Irkutskaya 1-a.), vifsin@mail.ru
[2] Voronezh Institute of the Russian Interior Ministry, Voronezh, Russia (394065,Voronezh, avenue Patriots 53), rosvment@mail.ru
[3] Lipetsk state technical University, Lipetsk, Russia. (398600, Russia, Lipetsk, St. Moscow, 30) kaf-inf@stu.lipetsk.ru.

In the article the methods of protection against unauthorized use of the program is connected to the ports of a computing device (hardware keys). Currently the hardware used to protect expensive or specialized software. Using hardware keys involves the development application in accordance with the security requirements. The most effective ways to bypass the protection with the use of hardware keys is to change the code of the protected program to remove from the algorithm of references to the key or tampering with his answers to the queries, the second method is based on a software simulation of the presence of a hardware device, and the program is not modified. The best quality protection provide the keys with the modified algorithm. We consider two variants of checking not only the existence but also the compliance of key application: generation of a digital signature and its verification, generation of an array of answers key at the prompts on one of the encryption algorithms. Hardware keys greatly complicate the process of copying and illegal use of the software, but is most effective when used in combination with the organizational methods of information protection.

Keywords: hardware key, software protection, security key, the key with a secret algorithm, the key memory key with the modified algorithm.

Для защиты программ от несанкционированного копирования и использования применяются аппаратные ключи защиты.

Ключи представляют собой аппаратные устройства, функции которых невозможно реализовать исключительно программными методами.

Современные программы распространяются с учетом использования владельцем строго определенного числа копии. Злоумышленник практически всегда может копировать содержимое любого носителя информации без искажений. Для защиты программных продуктов от несанкционированного использования возможен единственный путь – сделать невозможным выполнение программного кода на максимально большом числе вычислительных машин, не обладающих каким-либо уникальным свойством, на которое ориентировался разработчик. В большинстве случаев такое уникальное свойство необходимо придать системе путем внедрения в нее физического устройства, которое невозможно копировать с учетом всех его особенностей.

Цель работы – рассмотрение базовых способов применения аппаратных ключей для защиты программных продуктов от нелегального использования и информационных процессов их применения в программном коде.

Материалы исследований. Надежным приемом защиты от несанкционированного использования программы становятся подключаемые к портам вычислительной машины устройства – аппаратные ключи. В настоящее время аппаратные средства применяются для защиты дорогостоящего или специализированного программного обеспечения.

Общий алгоритм работы вычислительной машины с установленным аппаратным ключом можно представить, как последовательность шагов:

1 Запрос на выполнение защищаемой программы и начало выполнения ее контролирующего кода;

2. Запрос контролирующего кода программы к аппаратному ключу;

3. Генерация ответа аппаратным ключом для контролирующей части защищаемой программы;

4 Проверка ответа аппаратного ключа на правильность и принятие на основе результатов анализа решения о прерывании выполнения защищаемой программы.

Использование аппаратных ключей предполагает разработку приложений в соответствии с требованиями обеспечения безопасности. Программная часть средств защиты информации предполагает наличие двух ступеней защиты.

Первая состоит в шифровании части защищаемой программы и внедрения в нее средства перехвата управления при запуске, перехват управления необходим для проверки наличия специальных средств обхода защиты (низкоуровневые средства отладки выполнения программ, средства трассировки выполнения программ). Нужно отметить существование различных вариантов функционирования защищенной среды приложения:

различные способы «засорения» информационного обмена между ключом и программой, динамическая антиотладочная проверка и другие.

Вторая программная составляющая – это набор высокоуровневых API-функций, реализующих низкоуровневые операции для работы с аппаратным ключом. К таким низкоуровневым операциям относятся проверка наличия ключа, запросы на шифрование, запросы на специализированную обработку данных в соответствии с возможностями устройства. Исходя из того же риска отладки и трассировки защищаемой программы, необходимо применять средства маскировки точек входа и выхода из функций.

Существует несколько вариантов реализации аппаратных ключей защиты программного обеспечения. Первые устройства такого рода подключались к LPT (LPT PRO) или COM порту компьютера «в разрыв» с оборудованием к нему подключенным. В настоящее время наиболее часто для подключения ключа используется последовательный USB порт компьютера (Guardant Sign, Guardant Code, Guardant SD). Встречаются также специализированные устройства, подключаемые к компьютеру как плата расширения PCI (АДМЗ Аккорд), которые в свою очередь взаимодействуют с ключами.

Для классификации аппаратных ключей можно использовать различные группирующие признаки. Однако наиболее значимые характеристики относятся все же к защитным функциям ключей.

Вопрос пригодности ключей для защиты приложений сводится к решению важной задачи: злоумышленник имеет в своем распоряжении программный продукт и легальный аппаратный ключ для его использования, его основная задача состоит в том, чтобы, не выходя за заданный уровень затрат ресурсов реализовать модификацию исследуемой программы и среды ее выполнения таким образом, чтобы снять требование обязательного наличия аппаратного ключа при ее выполнении.

Наиболее действенными способами обхода защиты с применением аппаратных ключей является изменение кода защищаемой программы для удаления из алгоритма обращений к ключу или подделки его ответов на запросы. Второй способ основан на программной симуляции наличия аппаратного устройства, при этом программа не модифицируется. Симуляция присутствия ключа в системе предполагает внедрение в среду исполнения программы виртуального устройства функционально идентичного алгоритмам ключа, при этом важно не полное соответствие алгоритма работы, а соответствие его ответов на запросы программы, ответам настоящего устройства.

Рассмотрим эволюцию защитных функций аппаратных ключей от наиболее простых.

Наиболее простые устройства — это так называемые ключи с памятью. Предполагается, что они имеют ограниченный объем памяти для записи каких-либо

параметров, при этом часть памяти может быть не перезаписываемой. Данные, записанные в памяти, используются для проверки подлинности ключа. Такое исполнение не может обеспечить должную защиту приложения, так как уязвимо к симуляции (копию области памяти ключа можно сделать без ресурсных затрат).

На втором месте по защищенности стоят ключи с жестким секретным алгоритмом обработки входящих сообщений. При использовании таких устройств на этапе отладки программы необходимо сгенерировать конечное множество возможных ответов ключа на запросы защищаемой программы и в каком-либо виде хранить их в тексте программы или ресурсных файлах.

Проверка наличия оригинального ключа осуществляется путем сравнения ответов на запросы программы, полученных от него, с хранимыми эталонными. Ограничение на конечность множества ответов наложено размером программного обеспечения.

Модификацией алгоритма с секретным алгоритмом является система с секретным алгоритмом с параметрами, которая позволяет изменять алгоритм просчета ответов на запросы защищаемой программы. Это повышает устойчивость к симуляции, и дает возможность распространять программу в одном виде, а в зависимости от параметров алгоритма генерировать значения для разных версий функциональности программы.

Наиболее качественную защиту обеспечивают ключи с модифицируемым алгоритмом. Данные решения достаточно дорогостоящи, в следствии сложной аппаратной платформы устройства. Наиболее важное достоинство состоит в возможности переноса некоторых процедур защищаемой программы в память ключа с исполнением их процессором ключа. Таким образом, возможно исключить наиболее действенные методы анализа поведения защищаемой программы в плане взаимодействия с ключом, используемые злоумышленниками.

Для защиты приложений, с практической точки зрения, можно воспользоваться двумя вариантами проверки не только наличия, но и соответствия ключа приложению: генерация электронной цифровой подписи и ее проверка, генерация массива ответов ключа на запросы программы по одному из алгоритмов шифрования.

Рассмотрим варианты практической реализации системы защиты приложения с использованием ключа Guardant.

Ключами Guardant поддерживается множество аппаратно реализованных в устройствах алгоритмов криптографического преобразования (GSII64, RAND64, ECC160, AES128 Encode, AES128 Decode), предоставляются стандартные функции для их использования.

Первый подход к защите предполагает создание электронной цифровой подписи, ее хранение в коде программы и последующую ее проверку. Ключи Guardant имеют возможность аппаратной генерации электронной цифровой подписи с помощью алгоритма ECC160 на основе математического аппарата эллиптических кривых, для этого используется функция GrdSign со списком параметров.

Нужно отметить, что многие аппаратно реализованные функции предполагают использование в качестве параметра алгоритма либо случайного числа, либо вектора, стандартные функции языков программирования (для Delphi это пара randomize, random), их нельзя считать безопасными, поэтому предпочтительно применять специализированные API функции. Так, в наборе MSDN Microsoft присутствует функция CryptGenRandom из набора Microsoft Crypto API.

Ответная функция проверки электронной цифровой подписи для ключей Guardant – GrdVerifySign, она не реализована аппаратно. Использование электронной цифровой подписи предполагает хранение в памяти аппаратного ключа открытого ключа шифрования, и как следствие необходимости присутствия открытого ключа в коде программы. В таком случае предусматриваются механизмы его шифрования, например, алгоритмом GSII64.

Такой подход не исключает модификации функции проверки электронной цифровой подписи, для того, чтобы она выдавала положительный результат независимо от исходных данных.

Второй вариант проверки подлинности ключа основан на использовании однонаправленных алгоритмов шифрования для создания массива ответов ключа на запросы программы. В ключах Guardant для этих целей можно, например, использовать аппаратный алгоритм шифрования AES, который вызывается функцией GrdCrypt с набором параметров. Для реализации защиты необходимо во время проектирования приложения создать таблицу достаточно большого размера «вопрос-ответ» с результатами шифрования данных, посылаемых приложением. В рабочем приложении с помощью надежного генератора случайных чисел выбирается одна из множества записей таблицы и расшифровывается (для Guardant – с помощью функции GrdCrypt с параметром Decrypt). Для повышения качества защиты нужно предусмотреть отсутствие прямого сравнения результата расшифровки и эталона, лучше это сделать в выражении или другой процедуре.

Стоит отметить, что применение какой-либо системы защиты должно в полной мере опираться на назначение программного обеспечения.

Комбинация описанных способов проверки подлинности ключа использована для защиты от запуска приложения «Модуль расчета кратчайших путей к материальным ценностям» из состава комплекса для обучения сотрудников силовых ведомств.

Задача, которую решает применение электронного ключа, состоит в разрешении запуска программного обеспечения только установленным кругом лиц, не предусмотрена защита конфиденциальных сведений. Организация защиты предполагает неограниченное использование программного комплекса в целом. Применение ключей Guardant позволило распределить контроль запуска программного обеспечения по модулям в зависимости от категории пользователя.

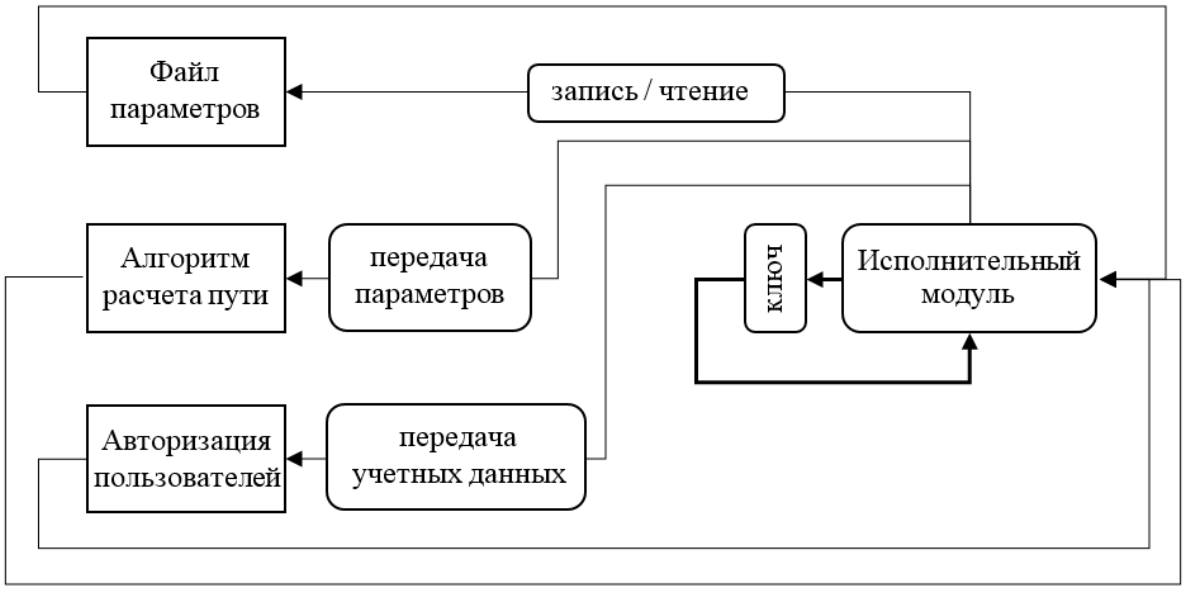

Рис. 1. Схема обращения исполнительного модуля к функциональным частям программного средства

На рисунке выше изображена схема взаимодействия исполнительного модуля программы с ее функциональными частями, работа которых должна быть защищена от возможного несанкционированного использования. Обращение к аппаратному ключу происходит при авторизации пользователей системы, при чтении и записи информации о состоянии программы в конфигурационный файл, перед началом расчета параметров стойкости защищаемого объекта.

Такое распределение проверок позволяет в полной мере ограничить несанкционированное использование программного средства. Организация защиты программ предполагает постоянно меняющуюся структуру и вид запросов к аппаратному ключу, чтобы создать наиболее неблагоприятную ситуацию для злоумышленника как в плане отладки программы в ассемблере, так и в плане создания программах эмуляторов аппаратных ключей.

Заключение. Аппаратные ключи могут в значительной степени усложнить процесс копирования и нелегального использования программного обеспечения, но наиболее

эффективны они при применении в комплексе с организационными, правовыми методами защиты информации.

Применение же таких средств для ограничения использования не дорогостоящих коммерческих программных продуктов, а программного обеспечения ведомственной специфики является важным компонентом системы разграничения доступа к программным ресурсам.

Список литературы

1. Аппаратная защита приложений от компьютерного пиратства на Microsoft Windows RT 8.1 [Электронный ресурс]. — Режим доступа: http://www.guardant.ru/press-center/publication/2013-11-06.html (дата обращения: 06 ноября 2013 года).
2. Варлатая С.К. Программно-аппаратная защита информации: учеб. Пособие /С.К. Варлатая, М.В. Шаханова. — Владивосток: Изд-во ДВГТУ, 2007.
3. Доля А. Аппаратные ключи - на страже программ [Электрон. ресурс] ‑ 16 января 2004. ‑ Режим доступа: http://fcenter.ru/online/softarticles/interview/8360
4. Панов А.С. Реверсинг и защита программ от взлома / А.С. Проскурин. – СПб.: БХВ-Петербург, 2006. – 256 с.: ил.
5. Проскурин В.Г. Защита программ и данных: учеб. пособие для студ. учреждений высшего проф. образования / В.Г. Проскурин. 2-с изд., стер. – М.: Издательский центр «Академия», 2012. 208 с. – (Сер. Бакалавриат).
6. Скляров Д.В. Искусство защиты и взлома информации / Д.В. Скляров — СПб.: БХВ-Петербург, 2004. — 288 с.
7. Современные технологии защиты программ: что выбрать разработчику? [Электронный ресурс]. — Режим доступа: http://www.guardant.ru/press-center/publication/2010-08-13.html (дата обращения: 13 августа 2010 года).

Рецензенты:

Душкин А.В., д.т.н., доцент, начальник кафедры управления и информационно-технического обеспечения ФКОУ ВПО «Воронежский институт ФСИН России», г. Воронеж;

Сумин В.И., д.т.н., профессор, профессор кафедры управления и информационно-технического обеспечения ФКОУ ВПО «Воронежский институт ФСИН России», г. Воронеж.