Эссе по курсу "Защита информации"
кафедра радиотехники
Московский физико-технический институт (ГУ МФТИ)
http://www.re.mipt.ru/infsec

Защита программ и данных с помощью электронных ключей

Малых Антон Владимирович

2005, г.Долгопрудный.

1. Содержание

2. Введение

Редкий коммерческий продукт выходит на рынок без какой-либо защиты: ввод серийного номера, активация через Интернет, ключевая дискета. Здесь вступает в силу постулат всех борцов с пиратством: «*любую защиту можно взломать*»; вопрос в том, какими силами и каково будет соотношение стоимости защиты к стоимости взлома. Считается уже хорошим результатом, если затраты на взлом превысили затраты на защиту.

В связи с таким положением дел, на мировом рынке появились компании, предлагающие простые для применения и в то же время мощные инструменты для защиты програмных продуктов от нелегального копирования. Одними из самых успешных таких инструментов оказались электронные ключи: устройства, подсоединяемые к порту *USB* либо *LPT* (в случае интерфейса *LPT* ключ становится «прозрачным» для других *LPT*-устройств), и взаимодействующие с защищенной программой.

3. Обзор технологий защиты ПО

Основная идея защиты ПО с помощью электронных ключей состоит в том, что в программу внедряется некий код, который взаимодействует с ключом, проверяет его подлинность, а также с помощью ключа кодирует, декодирует или вычисляет хэш-функции от некоторых данных, которые в дальнейшем используются программой. Такую функциональность обеспечивают встраиваемые в ключи программируемые логические матрицы (ПЛМ) или *ASIC*-чипы (заказные микросхемы).

Рассмотрим подробнее различные алгоритмы и технологии поддерживаемые ключами.

3.1. Уникальный идентификационный номер ключа (ID)

Эту возможность ключей вряд ли можно назвать технологией защиты. Идентификаторы были у самых первых ключей, появившихся на рынке в 1995 году: *Hardlock*, *Activator*, *GoldKey*. Как показала практика, защита программ, основанная только на опросе *ID* ключа не представляет серьезной проблемы для хакеров. Вскоре после выпуска программ, защищенных этими ключами (*AutoCAD*, *UniGraphics*, *3DSMax*, …) появились хакерские программы, способные полностью програмно эмулировать данные ключи.

Таким образом на сегодняшний день нельзя строить защиту с помощью идентификатора ключа, нужно обязательно задействовать какие-либо алгоритмы, зашитые в ключе.

3.2. Аппаратные функции y=f(x)

Все современные ключи имеют аппаратную реализацию функций вида *y=f(x)*, причем, как правило, не одну а несколько. Внутри ключа функции задаются дескрипторами, а разработчик может переопределить стандартные дескрипторы и использовать другие функции по своему усмотрению.

Например, ключ *GUARDANT STEALTH* имеет до 18 таких функций в одном корпусе. На вход этого ключа подается последовательность длиной от 4 до 255 байт, ключ возвращает преобразованную последовательность соответствующей длины. Преобразованные данные защищенная программа может использовать в своей работе. Конкретный вид каждого алгоритма преобразования данных задается его дескриптором длиной до 200 байт, которые хранится в энергонезависимой памяти *EEPROM*. Таким образом, для того чтобы только получить конкретный вид алгоритма, по которому преобразуется входная информация, хакеру придется перебрать порядка 2^{1600} комбинаций.

В старых ключах такие функции тоже были, например, функция кодов возврата, реализованная в ключах *MemoHASP-1*, работает по такому принципу: на вход подается последовательность длиной 2 байта, ключ возвращает последовательность длиной 8 байт. У ключей *Sentinel PRO* функция выглядит несколько иначе. Она работает по принципу: на запрос длиной 4 байта (2^{32} комбинаций) возвращается ответная последовательность длиной 4 байта. Вид алгоритма, по которому входная последовательность преобразуется в выходную, задается дескриптором длиной 4 байта (2^{32} комбинаций).

Оба ключа были взломаны в общем виде, т.е. было создано хакерское ПО, позволяющее получить из ключа значения дескрипторов и эмулировать для защищенной программы наличие и функциональность ключа. Это говорит о том, что универсальной защиты не существует и рано или поздно появятся программы, способные эмулировать самые современные ключи.

3.3. Алгоритмы шифрования

Многие производители ключей внедряют возможность аппаратного шифрования/дешифрования в свои ключи. Многие современные шифры практически невозможно взломать методом грубой силы, а значит программа будет надежно защищена при условии того что шифр не будет взломан.

Как известно, существует множество бесплатных библиотек, реализующих различные алгоритмы шифрования, поэтому шифрование для нужд программы стоит производить посредством таких библиотек, а не с помощью ключа. Встроенный в ключ алгоритм шифрования нужен для того чтобы защитить память ключа: как произовольные данные, записанные разработчиком в *EEPROM*, так и дескрипторы аппаратных функций.

Возникает вопрос: а откуда программа возьмет ключ (пароль) для шифра? Если он будет зашит в тело программы, то методом *reverse ingeneering* не составит труда хакеру заполучить его, пусть даже он будет зашифрован или замаскирован. Поэтому разумно получать ключ шифра из выхода хэш-функции: пропуская некоторые данные через аппаратную функцию, выход это аппаратной функции будет ключом шифра. Такой метод более устойчив ко взлому.

Напрмер, ключ *HASP HL* от *Aladdin Software Security* и ключ *SentinelPro* от *Rainbow Technologies* умеет шифровать и дешифровать данные по алгоритму *AES*, который выбран американским стандартом шифрования.

В ключах *Guardant Stealth II* используется алгоритм *GSII64*, специально разработанный компанией *Актив* для своих ключей. В отличие от однонаправленного алгоритма, реализованного в ключах *Guardant Stealth*, *GSII64* — алгоритм симметричный. Это означает, что с его помощью можно проводить как прямое преобразование данных (кодирование), так и обратное (декодирование). Эти операции выполняются на основе одного или нескольких секретных ключей, которые программист может записывать в память *Guardant Stealth II* самостоятельно — при помощи утилиты, входящей в комплект разработчика.

Алгоритм *GSII64* работает с дескрипторами (ключами) длиной 16 или 32 байт (128 или 256 бит) и

может выполнять следующие функции:

- блочное преобразование данных (размер блока — 8 байт);
- поточное преобразование данных произвольной длины;
- генерацию псевдослучайных чисел;
- вычисление Message Authentication Code (*MAC*).

Алгоритм *GSII64* поддерживает классические режимы работы блочных симметричных алгоритмов: *ECB*, *CBC*, *CFB*, *OFB*.

3.4. Защита памяти ключа

Для обеспечения должных норм безопасноти, память ключа не должна быть открыта каждому. Поэтому производители вводят ограничения на доступ к памяти ключа.

К примеру у ключей *Guardant Fidus* и *Stealth II* введена трехуровневая система доступа, и соответственно три пароля, по одному для:

- чтения памяти ключа и выполнения аппаратных алгоритмов;
- записи в память ключа;
- для выполнения с ключом специальных операций, таких как установка/удаление запретов на чтение/запись памяти ключа;

Кроме того в ключе можно установить аппаратный запрет на чтение или запись.

3.5. Защита сетевого ПО

Существуют ключи предназначенные исключительно для защиты сетевых приложений: *Guardant Net II*, *Aladdin HASP4 Net*, *Aladdin HASP HL Net*.

Для полноценной защиты и лицензирования сетевого продукта достаточно одного ключа на всю локальную сеть. Ключ устанавливается на любой рабочей станции или сервере сети. Сетевые электронные ключи обеспечивают не только защиту, но и лицензирование приложений. Задача лицензирования – ограничить число работающих в сети копий программы. Проще говоря, сколько лицензий приобрел пользователь, столько копий программы он сможет одновременно использовать. Число лицензий определяет поставщик и фиксирует в памяти ключа (сетевой ресурс ключа). Защищенное сетевое приложение обращается к ключу не напрямую, а через специальный программный сервер. Сервер ключа обеспечивает прохождение запроса от защищенного приложения (клиента) непосредственно к ключу и обратно. Загружается сервер на том же компьютере, к которому подсоединен ключ.

3.6. Защита от отладки

Для более эффективного противодействия хакерам производители разрабатывают технологии противодействия отладчикам, таким как *SoftIce*. Эти технологии работают с двух сторон: со стороны защищаемой программы и со стороны ключа. Со стороны программы внедряется код, который анализируя общее состояние системы проверяет запущен ли *SoftIce* или подобные ему отладчики и в случае если запущены прекращает нормальную работу. Ключ, в свою очередь, анализирует временные интервалы между запросами, и по ним эвристически определяет, работает ли программа в штатном режиме или в режиме отладки.

4. Обзор компаний, производящих электронные ключи

4.1. Aladdin Software Security

Компания на сегодняшний день – мировой лидер в сфере электронных ключей. *Aladdin Software Security* – подразделение израильской компании *Aladdin Knowledge Systems*. Компания выпускает два вида электронных ключей: *eToken* и *HASP*. Ключи и смарт-карты *eToken* разработаны для задач авторизации а аутентификации, поэтому далее они не рассматриваются.

Ключи *HASP*, в свою очередь, выпускаются двух версий: линейка *HASP4* (старая версия, но все еще выпускаемая), и линейка *HASP HL*.

Линейка *HASP4* состоит из базовой модели *Standart*, продвинутых *M1* и *M4*, сетевой *Net* и ключа с возможностью ограничивать работоспособность по времени *HASP4 Time*.

Линейка HASP HL иначе называется HASP пятого поколения. Возможности ключей значительно расширились по сравнению с четвертым поколением:

- публичный алгоритм AES для шифрования данных;
- разделение процесса защиты и лицензирования;
- единое HASP API для локальных и сетевых ключей;
- увеличение объема памяти ключей (4096 байт);
- полная обратная совместимость с ключами HASP4 и HASP3;
- использование RSA для системы удаленного перепрограммирования HASP HL;
- генератор исходных кодов;
- мастер ключ HASP HL.

4.2. Актив

С момента основания компания *Актив* занимается разработкой, производством и продажей систем защиты от компьютерного пиратства. Основным направлением работы компании является производство электронных ключей *Guardant*. На сегодняшний момент компания *Актив* — самый крупный российский производитель электронных ключей.

Младшим в семействе ключей от *Актива* является ключ *Guardant Fidus* с *USB* или *LPT* интерфейсом.

Самая эффективная защита обеспечивается ключом *Guardant Stealth II*, который пришел на смену ключу *Guardant Stealth*. Отличается от своего младшего собрата наличем аппаратных функций вида *y=f(x)* и алгоритмом шифрования *GSII64* с 128- или 256-битным ключом.

Завершает семейство ключ *Guardant Net II*, позволяющий эффективно защищать сетевые приложения.

4.3. Rainbow Technologies

Как сообщается на сайте www.rainbow.com, в марте 2004, *Rainbow* прекратила свое независимое существование и слилась с компанией *SafeNet*. Однако *SafeNet* до сих пор продолжает выпускать электронные ключи.

К сожалению, компания неактивно ведет свой бизнес на российском рынке, на российском сайте не удалось обнаружить никакой документации на ключи, производимые данной фирмой. Известно, лишь только что продукция состоит из ключей *SentinelSuperPro*, *SentinelSuperPro797* для подключения к порту *LPT* и *SentinelDuo* с интерфейсом *USB*, причем последний может работать как на платформе *Microsoft Windows*, так и на *Macintosh*.

Также компания выпускает продукт *SentinelLM*, который судя по описаниям похож на сетевую версию ключей, но из возможностей заявлено только управление лицензированием.

При написании статьи использовались материалы сайтов www.aladdin.ru и www.guardant.ru.