Программно-аппаратные средства обеспечения информационной безопасности

Лекция № 7

Современные программно-аппаратные средства обеспечения информационной безопасности. Часть 2

Рахманенко И.А.

План

- Персональные идентификаторы
- Электронные ключи

ПЕРСОНАЛЬНЫЕ ИДЕНТИФИКАТОРЫ

Персональные идентификаторы

- iButton
- eToken, JaCarta (уже рассмотрены)
- ruToken
- ESMART Token
- …

iButton

- iButton (Touch Memory) — это семейство многофункциональных микроэлектронных устройств, разработанных фирмой Dallas Semiconductor (USA) в настоящее время выпускаемых фирмой Maxim).

iButton

Области применения:

- средства идентификации;
- системы контроля доступа;
- системы компьютерной безопасности;
- температурный мониторинг и др.

iButton

- Каждое устройство iButton имеет уникальный номер (ID), записываемый в процессе изготовления.
- Все устройства iButton помещаются в стальной цилиндрический корпус **MicroCan**, выполнены по жестким стандартам и выдерживают серьезные механические и температурные нагрузки.

iButton

- **Обмен данными** с iButton производится через интерфейс 1-Wire. Информация в этом интерфейсе передается по единственному проводнику. Питание iButton получают из этого же проводника, заряжая внутренний конденсатор в моменты, когда на шине нет обмена данными.
- Скорость обмена достаточна для обеспечения передачи данных в момент касания контактного устройства.

iButton. Характеристики

Размер EEPROM-памяти	0 - 8192 байт (в зависимости от модели)
Рабочий диапазон температур	От -40°C до +70°C
Температура хранения	От -55°C до +85°C
Используемый интерфейс	1-Wire
Мин. время записи/стирания	5 мс
Рабочее напряжение питания	0,5 В - 7 В
Долговечность	1 000 000 циклов
Срок хранения данных в памяти	10 лет
Макс. потребляемый ток	5 мкА
Устройства чтения	Считыватели iButton
Назначение	Системы идентификации

ruToken

- Программные и аппаратные средства для идентификации и аутентификации пользователей, электронной подписи и безопасного хранения криптографических ключей.
- Разрабатываются и выпускаются российской компанией «Актив».

ruToken

- Токены Рутокен получили свое название, поскольку позиционируются, как отечественные решения — практически все их комплектующие российского производства и сборка устройств производится на территории России.
- На продукты Рутокен имеются сертификаты ФСБ и ФСТЭК России.

ruToken. Модели

- Рутокен ЭЦП
- Рутокен ЭЦП Flash (+flash память до 64 Гб)
- Рутокен ЭЦП Bluetooth (работает с моб. устр.)
- Рутокен PINPad
- Рутокен S (без ЭЦП)
- Рутокен Lite (ключевой носитель без криптографических функций)
- Рутокен Web

Рутокен ЭЦП

- Электронный идентификатор с аппаратной реализацией электронной подписи, шифрования и хеширования по ГОСТ. Также имеется поддержка RSA 2048. Существует модификация, Рутокен КП, используемая только с продуктами КриптоТРИ и КриптоПРО Рутокен CSP.

Рутокен PINPad

- Решение класса TrustScreen, позволяющее визуализировать подписываемый документ в доверенной среде непосредственно перед наложением электронной подписи.
- Документ отображается на экране и, в случае подтверждения пользователем корректности информации, подпись осуществляется непосредственно на самом устройстве.

Рутокен PINPad

- Данное решение защищает от фишинга, атак при помощи средств удаленного управления, подмены содержимого документа при передаче его на подпись (атака Man-in-the-browser).

Рутокен Web

- Основное назначение — замена небезопасной аутентификации по связке «логин-пароль» на двухфакторную аппаратную аутентификацию. Решение основано на технологии электронной подписи и позволяет свести к нулю риск кражи или неправомерного использования учетных записей пользователей Web-ресурса.

ESMART Token

- ESMART Token ГОСТ
- ESMART Смарт-карты
- Токены ESMART (без алгоритмов ГОСТ)

КЛЮЧИ ЗАЩИТЫ

Аппаратные ключи защиты

- Ключ защиты (электронный ключ, англ. dongle) — аппаратное средство, предназначенное для защиты программного обеспечения (ПО) и данных от копирования, нелегального использования и несанкционированного распространения.

Аппаратные ключи защиты

- Содержат специализированную микросхему, либо защищённый от считывания микроконтроллер, имеющие уникальные для каждого ключа алгоритмы работы.
- Имеют защищённую энергонезависимую память небольшого объёма
- Могут иметь встроенный криптопроцессор (для аппаратной реализации шифрующих алгоритмов), часы реального времени.

Форм-факторы

- Чаще всего USB-устройства.
- Также встречаются с LPT- или PCMCIA-интерфейсами.
- Возможно впайка микросхемы на плату на этапе производства техники.

Схема работы

1. Ключ присоединяется к определённому интерфейсу компьютера.
2. Защищённая программа через специальный драйвер отправляет ему информацию, которая обрабатывается в соответствии с заданным алгоритмом и возвращается обратно.

Схема работы

3. Если ответ ключа правильный, то программа продолжает свою работу.
4. В противном случае она может выполнять определенные разработчиками действия, например, переключаться в демонстрационный режим, блокируя доступ к определённым функциям.

Причина появления

- Создатели ПО используют различные программные модули, контролирующие доступ пользователей с помощью ключей активации, серийных номеров и т. д.
- Такая защита – дешёвая и не надежная.
- Возможно создание программ, позволяющих нелегально сгенерировать ключ активации (генераторы ключей) или заблокировать запрос на серийный номер/ключ активации (патчи, крэки).
- Легальный пользователь может обнародовать свой серийный номер.

Причина появления

- Решение проблемы - создание аппаратной компонента защиты ПО.
- Первые электронные ключи появились в начале 1980-х годов.

Комплект разработчика ПО

- Современные электронные ключи часто определяются как мультиплатформенные аппаратно-программные инструментальные системы для защиты ПО.
- Кроме ключа предоставляются SDK (комплект разработчика ПО).

Комплект разработчика ПО

- В SDK входит все необходимое — средства разработки, полная техническая документация, поддержка различных операционных систем, детальные примеры, фрагменты кода, инструменты для автоматической защиты.
- SDK может включать в себя демонстрационные ключи для построения тестовых проектов.

Технология защиты ПО

- Построена на реализации запросов из исполняемого файла или динамической библиотеки к ключу с последующим получением и, если предусмотрено, анализом ответа.

Примеры запросов

- проверка наличия подключения ключа;
- считывание с ключа необходимых программе данных в качестве параметра запуска (используется, в основном, только при поиске подходящего ключа, но не для защиты);

Примеры запросов

- запрос на расшифровывание данных или исполняемого кода, необходимых для работы программы;
- запрос на расшифровывание данных, зашифрованных ранее самой программой (позволяет отправлять каждый раз разные запросы к ключу и, тем самым, защититься от эмуляции библиотек API / самого ключа)

Примеры запросов

- проверка целостности исполняемого кода путём сравнения его текущей контрольной суммы с оригинальной контрольной суммой, считываемой с ключа;

Примеры запросов

- запрос к встроенным в ключ часам реального времени (при их наличии; может осуществляться автоматически при ограничении времени работы аппаратных алгоритмов ключа по его внутреннему таймеру);

Технология защиты ПО

- Некоторые ключи позволяют разработчику хранить алгоритмы или отдельные части кода и исполнять их в самом ключе на его собственном микропроцессоре.
- Кроме защиты от нелегального использования это позволяет защитить алгоритмы от изучения, копирования и использования в своих приложениях конкурентами.
- Однако для простого алгоритма (а разработчики часто совершают ошибку, выбирая для загрузки недостаточно сложный алгоритм) может быть проведен криптоанализ по методу анализа "черного ящика".

Основа защиты

- Алгоритм преобразования (криптографический или другой).
- Реализован аппаратно — практически исключает создание полного эмулятора ключа, так как ключ шифрования никогда не передается наружу, что исключает возможность его перехвата.

Алгоритм шифрования

- Может быть секретным или публичным.
- Секретные алгоритмы разрабатываются самим производителем средств защиты, в том числе и индивидуально для каждого заказчика. Главным недостатком использования таких алгоритмов является невозможность оценки криптографической стойкости.

Алгоритм шифрования

- Публичный алгоритм обладает криптостойкостью. Такие алгоритмы проверяются рядом экспертов, специализирующихся на анализе криптографии.
- ГОСТ 28147—89, AES, RSA, Elgamal

Виды защиты

- Автоматическая (с помощью специального ПО)
- С помощью API функций.

Автоматическая защита

- Автоматические инструменты (входящие в SDK), позволяющие защитить программу «за несколько кликов мыши».
- Файл приложения «оборачивается» в собственный код разработчика.
- Чаще всего код осуществляет проверку наличия ключа, контроль лицензионной политики, внедряет механизм защиты исполняемого файла от отладки и декомпиляции (например, сжатие исполняемого файла) и др.

Автоматическая защита

- Для использования автоматического инструмента защиты не требуется доступ к исходному коду приложения. Такой механизм защиты не позволяет использовать весь потенциал электронных ключей и реализовать гибкую и индивидуальную защиту.

Защита с помощью API

- Самостоятельная разработка защиты, интегрируя систему защиты в приложение на уровне исходного кода.
- Библиотеки для различных языков.
- API - набор функций, предназначенных для обмена данными между приложением, системным драйвером и самим ключом.
- Операции с ключом через API: поиск, чтение и запись памяти, шифрование и расшифровывание данных при помощи аппаратных алгоритмов, лицензирование сетевого ПО и т. д.

Защита с помощью API

- Обеспечивает высокий уровень защищённости приложений.
- Нейтрализовать защиту, встроенную в приложение, достаточно трудно вследствие её уникальности и «размытости» в теле программы.
- Необходимость изучения и модификации исполняемого кода защищенного приложения для обхода защиты является препятствием к ее взлому.

Обход защиты

Задача злоумышленника — заставить защищённую программу работать в условиях отсутствия легального ключа, подсоединённого к компьютеру.

Злоумышленник имеет доступ ко всем открытым интерфейсам, документации, драйверам и может их анализировать на практике с привлечением любых средств.

Обход защиты

При наличии ключа, злоумышленник может:

- перехватывать все обращения к ключу;
- протоколировать и анализировать эти обращения;
- посылать запросы к ключу и получать на них ответы;
- протоколировать и анализировать эти ответы;
- посылать ответы от имени ключа и др.

Обход защиты

2 вида обхода защиты:

- Внесение изменений в программный модуль (взлом);
- Эмулирование наличия ключа путем перехвата вызовов библиотеки API для обмена с ключом.

Обход защиты

- Современные электронные ключи (например, ключи Guardant поколения Sign и современные ключи HASP HL) обеспечивают стойкое шифрование протокола обмена электронный ключ - библиотека API работы с ключом.
- В результате наиболее уязвимыми местами остаются точки вызовов функций этого API в приложении и логика обработки их результата.

Обход защиты. Простое условие

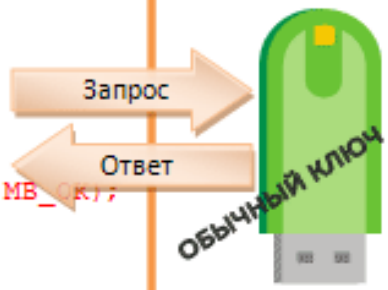
```
void _encrypt(
        LPVOID lpBuffer,
        UINT nBytes,
        UINT n1,
        LPBYTE lpTable,
        ULONG val)
{
  UINT    i, nbit, npos;
  BYTE    bit;
  ULONG   dw;
  LPBYTE  p = (LPBYTE)lpBuffer;

  i = nBytes * 8 - 1;

  if(IsDongleConnected() == FALSE)
  {
    MessageBox(NULL, "Ключ не найден", "Ошибка", MB_OK);
    return;
  }

  while(i != -1)
  {
    dw = (val >> i) & 0x01;
    nbit = lpTable[table4[n1] + i] & 0x07;
    npos = lpTable[table4[n1] + i] >> 3;
    if(dw)
    {
      bit = 1 << nbit;
      p[npos] |= bit;
    }
    else
    {
      bit = 1 << nbit;
      p[npos] &= ~bit;
    }
    i--;
  }
}
```

Если взломщик изменит это условие, то программа будет корректно работать без ключа.



Взлом программного модуля

- Злоумышленник исследует логику работы программы, анализируя код приложения, с целью выделения блока защиты и отключения его.
- Осуществляется с помощью отладки (или пошагового исполнения), дизассемблирования, декомпиляции и дампа оперативной памяти.
- Эти способы анализа исполняемого кода программы чаще всего используются злоумышленниками в комплексе.

Отладка

- Осуществляется с помощью отладчика.
- Позволяет по шагам исполнять любое приложение, эмулируя для него операционную среду.
- Важной функцией отладчика является способность устанавливать точки (или условия) остановки исполнения кода.
- С помощью них злоумышленник может отслеживать места в коде, в которых реализованы обращения к ключу (например, остановка выполнения на сообщении типа «Ключ отсутствует! Проверьте наличие ключа в USB-интерфейсе»).

Дизассемблирование

- Д - способ преобразования кода исполняемых модулей в язык программирования, понятный человеку — Assembler.
- В этом случае злоумышленник получает распечатку (листинг) того, что делает приложение.

Декомпиляция

- Д - преобразование исполняемого модуля приложения в программный код на языке высокого уровня и получение представления приложения, близкого к исходному коду.
- Может быть проведена только для некоторых языков программирования (например, для .NET приложений, создаваемых на языке C# и распространяемых в байт-коде - интерпретируемом языке относительно высокого уровня).

Дамп памяти

- Атака с помощью дапма памяти заключается в считывании содержимого оперативной памяти в момент, когда приложение начало нормально исполняться.
- В результате злоумышленник получает рабочий код (или интересующую его часть) в "чистом виде" (если, к примеру, код приложения был зашифрован и расшифровывается только частично, в процессе исполнения того или иного участка).

Противодействие отладке

Часто используются:

- нелинейность кода (многопоточность)
- недетерминированная последовательность исполнения
- обфускация кода (бесполезными функциями, выполняющими сложные операции, с целью запутать злоумышленника)
- использование несовершенства самих отладчиков и др.

Эмуляция ключа

- Воздействия на код программы не происходит.
- Эмулятор повторяет поведение реального ключа.
- Строятся на основе анализа перехваченных запросов приложения и ответов ключа на них.

Могут быть:

- табличными (содержать в себе все необходимые для работы программы ответы на запросы к электронному ключу);
- полными (полностью эмулируют работу ключа, так как взломщикам стал известен внутренний алгоритм работы).

Эмуляция ключа

- Построить полный эмулятор современного электронного ключа — это достаточно трудоёмкий процесс, требующий большого количества времени и существенных инвестиций.
- Примеры есть: компания Aladdin - в 1999 году разработан корректно работающий эмулятор ключей HASP3 и HASP4. Ключ использовал проприетарный алгоритм кодирования, который был взломан.

Эмуляция ключа

- Большинство современных ключей используют публичные криптоалгоритмы, поэтому злоумышленники предпочитают атаковать какой-то конкретный защищённый продукт, а не защитный механизм в общем виде.
- Существующие табличные эмуляторы основаны на неграмотном использовании функционала ключей разработчиками.
- Реализованы для отдельных приложений.

Примеры

- Sentinel HL (ранее - HASP)
- Guardant
- SenseLock

Sentinel

Задачи:

- защита готовых программ (.exe, .dll, .jar, .apk);
- гибкая защита программного обеспечения при помощи Sentinel LDK API;
- лицензирование и защита отдельных модулей и функций ПО;
- защита и лицензирование 1С-конфигураций;
- программная защита для продажи и активации программ через Интернет.

Модели Sentinel

- **Sentinel HL Basic**
- Наиболее простое и эффективное решение для защиты недорогих программ, не требующих управления лицензированием и сохранения в памяти ключа защиты параметров и настроек.
- Алгоритм: AES · Уникальный ID: нет · Память RW/R: нет · Кол-во лицензий: 1

Модели Sentinel

- **Sentinel HL Pro**
- Самая подходящая модель для защиты программного обеспечения с лицензированием по используемым компонентам. Наиболее популярные USB-ключи для защиты программ и данных.
- Алгоритм: AES · Уникальный ID: да · Память RW/R: 112/112 байт · Кол-во лицензий: 11-39

Модели Sentinel

- **Sentinel HL Max**
- Оптимален для защиты сложного программного обеспечения с лицензированием по функциональности и/или количественным показателям.
- Алгоритм: AES · Уникальный ID: да · Память RW/R/D*: 4/2/25 КБ · Кол-во лицензий: 240-2160

Модели Sentinel

- **Sentinel HL Max Micro**
- Благодаря миниатюрному корпусу практически не выступает из USB-порта, очень удобен при использовании в ноутбуках и планшетах. По функционалу аналогичен ключу Sentinel HL Max.
- Алгоритм: AES · Уникальный ID: да · Память RW/R/D*: 4/2/25 КБ · Кол-во лицензий: 240-2160

Модели Sentinel

- **Sentinel HL Time**
- Эффективное решение для организации аренды, лизинга, подписки на защищенное программное обеспечение, распространения пробных версий (trial). Аналогичен USB-ключу Sentinel HL Max, но дополнительно имеет часы реального времени, используемые при лицензировании программного обеспечения.
- Алгоритм: AES · Уникальный ID: да · Память RW/R/D*: 4/2/25 КБ · Кол-во лицензий: 240-2160

Модели Sentinel

- **Sentinel HL Net**
- Сетевые USB-ключи, разработанные специально для защиты корпоративного ПО, при этом USB-ключи могут работать и как локальные. Ключ защиты Sentinel HL Net позволяет ограничивать количество пользователей, одновременно работающих с защищенными программами, лицензировать компоненты, функциональность и другие количественные показатели.
- Алгоритм: AES · Уникальный ID: да · Память RW/R/D*: 4/2/25 КБ · Кол-во лицензий: 240-2160

Модели Sentinel

- **Sentinel HL NetTime**
- Совмещает в себе функции ключей Sentinel HL Net и Sentinel HL Time, оптимален для ограничения работы по времени защищенных программ совместно с контролем количества одновременно работающих пользователей.
- Алгоритм: AES · Уникальный ID: да · Память RW/R/D*: 4/2/25 КБ · Кол-во лицензий: 240-2160

Модели Sentinel

- **Sentinel HL Drive**
- Предназначен для распространения программного обеспечения на ключе со встроенной Flash-памятью. Совмещает в себе функции ключа Sentinel HL Max и Flash-накопителя.
- Алгоритм: AES · Уникальный ID: да · Память RW/R/D*: 4/2/25 КБ · Кол-во лицензий: 240-2160

Модели Sentinel

- **Sentinel HL Max Board**
- Ключ предназначен для производителей Embedded систем из готовых комплектующих. Sentinel HL Max Board монтируется непосредственно во внутренний USB разъём материнской платы, что позволяет скрыть ключ внутри корпуса Embedded устройств. По функционалу аналогичен ключу Sentinel HL Max.
- Алгоритм: AES · Уникальный ID: да · Память RW/R/D*: 4/2/25 КБ · Кол-во лицензий: 240-2160

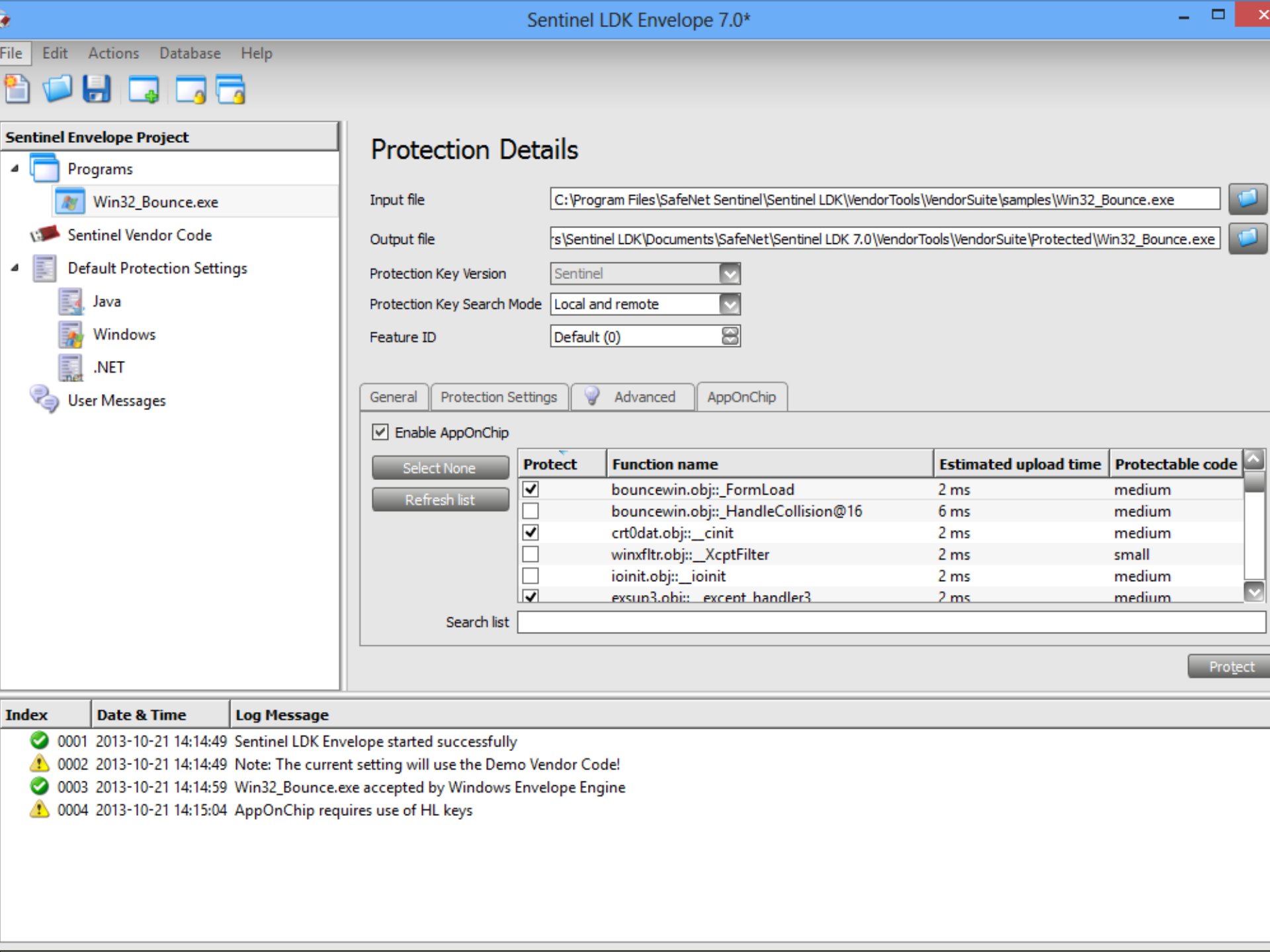
Модели Sentinel

- **Sentinel HL Max Chip**
- USB-ключ предназначен для разработчиков ПО, имеющих производство собственных комплектующих. Ключ имеет стандартный интерфейс USB-шины(SOIC8) для припайки на плату. По функционалу аналогичен ключу Sentinel HL Max.
- Алгоритм: AES · Уникальный ID: да · Память RW/R/D*: 4/2/25 КБ · Кол-во лицензий: 240-2160

Sentinel Envelope

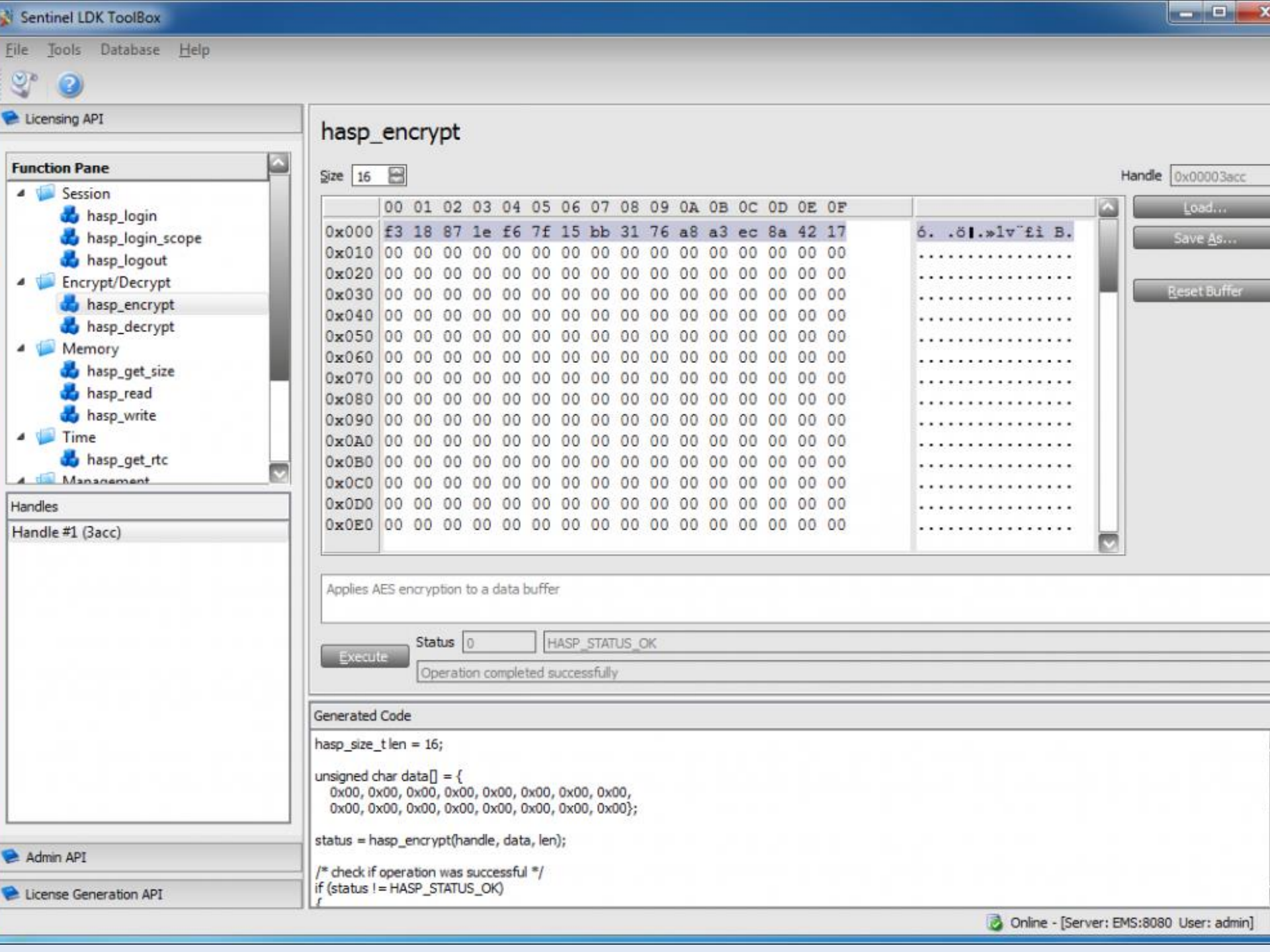
Защита программы устанавливается автоматически всего за три клика мышкой.

- Защита .Net-программ (exe/dll).
- Защита Java-программ.
- Защита для Windows (32/64-бит), Linux и Mac OS программ.
- ыAppOnChip - исполнение кода внутри аппаратного ключа защиты.
- Защита приложений для Android.


Sentinel LDK Envelope 7.0*
File
Edit
Actions
Database
Help
Sentinel Envelope Project
Programs
Win32_Bounce.exe
Sentinel Vendor Code
Default Protection Settings
Java
Windows
.NET
User Messages
Protection Details
Input file
C:\Program Files\SafeNet Sentinel\Sentinel LDK\VendorTools\VendorSuite\samples\Win32_Bounce.exe
Output file
s\Sentinel LDK\Documents\SafeNet\Sentinel LDK 7.0\VendorTools\VendorSuite\Protected\Win32_Bounce.exe
Protection Key Version
Sentinel
Protection Key Search Mode
Local and remote
Feature ID
Default (0)
General
Protection Settings
Advanced
AppOnChip
Enable AppOnChip
Select None
Refresh list
Protect
Function name
Estimated upload time
Protectable code
bouncewin.obj::_FormLoad
2 ms
medium
bouncewin.obj::_HandleCollision@16
6 ms
medium
crt0dat.obj::__cinit
2 ms
medium
winxfltr.obj::__XcptFilter
2 ms
small
ioinit.obj::__ioinit
2 ms
medium
Search list
Protect
Index
Date & Time
Log Message
0001 2013-10-21 14:14:49 Sentinel LDK Envelope started successfully
0002 2013-10-21 14:14:49 Note: The current setting will use the Demo Vendor Code!
0003 2013-10-21 14:14:59 Win32_Bounce.exe accepted by Windows Envelope Engine
0004 2013-10-21 14:15:04 AppOnChip requires use of HL keys

Sentinel LDK API

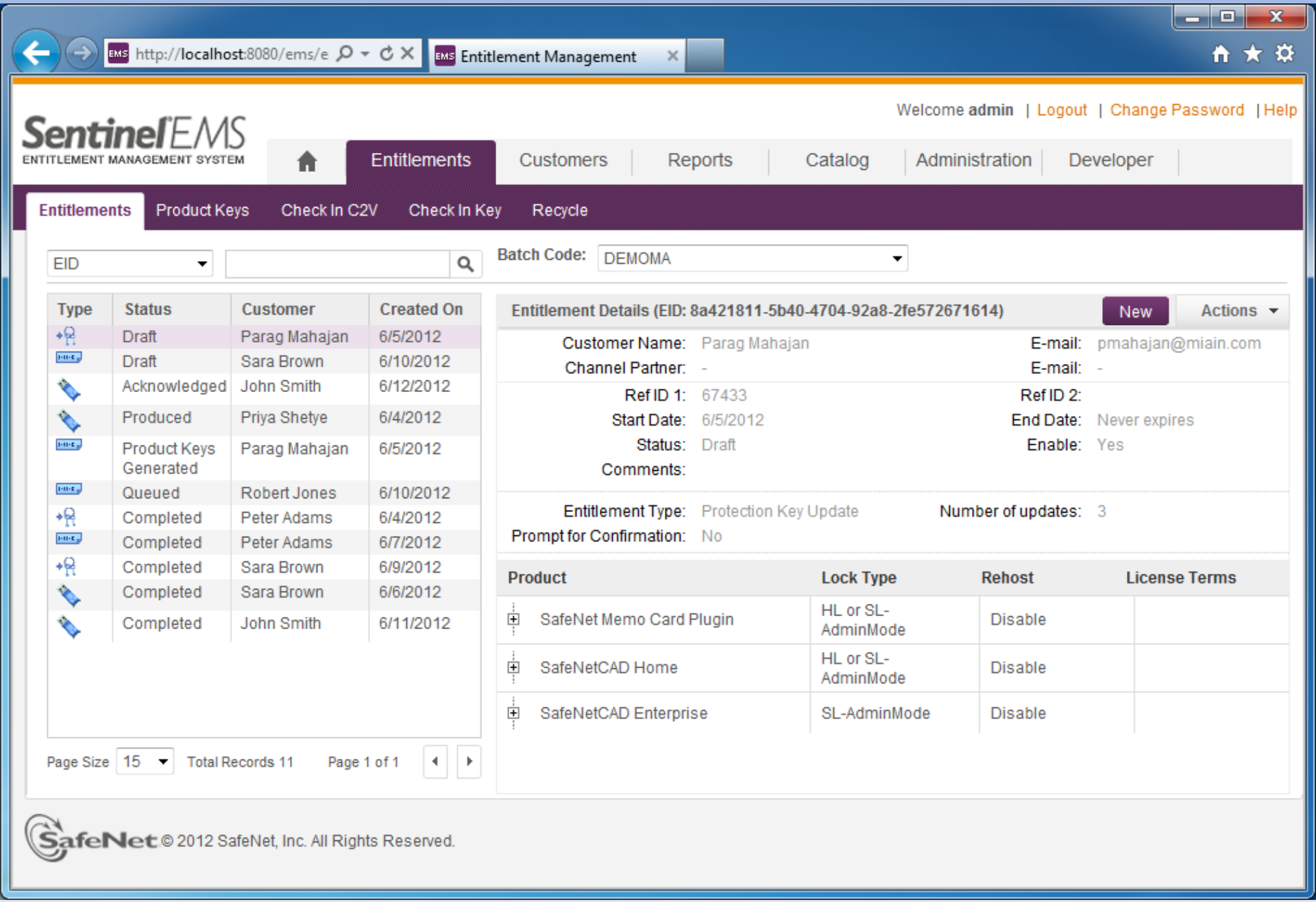
- Функции для ручного построения защиты ПО.
- Всего тринадцать API-функций, которые позволяют построить гибкую защиту программы.
- Удобный генератор исходных кодов защиты ПО.
- Библиотеки для всех популярных языков программирования.


Sentinel LDK ToolBox
File Tools Database Help
Licensing API
Function Pane
Session
hasp_login
hasp_login_scope
hasp_logout
Encrypt/Decrypt
hasp_encrypt
hasp_decrypt
Memory
hasp_get_size
hasp_read
hasp_write
Time
hasp_get_rtc
Handles
Handle #1 (3acc)
Admin API
License Generation API
hasp_encrypt
Size 16
Handle 0x00003acc
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0x000 f3 18 87 1e f6 7f 15 bb 31 76 a8 a3 ec 8a 42 17
0x010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Load...
Save As...
Reset Buffer
Applies AES encryption to a data buffer
Execute
Status 0
HASP_STATUS_OK
Operation completed successfully
Generated Code
hasp_size_t len = 16;
unsigned char data[] = {
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
status = hasp_encrypt(handle, data, len);
/* check if operation was successful */
if (status != HASP_STATUS_OK)
Online - [Server: EMS:8080 User: admin]

Sentinel EMS

Система управления лицензиями, с помощью которой разработчик может управлять продажами своего программного обеспечения.

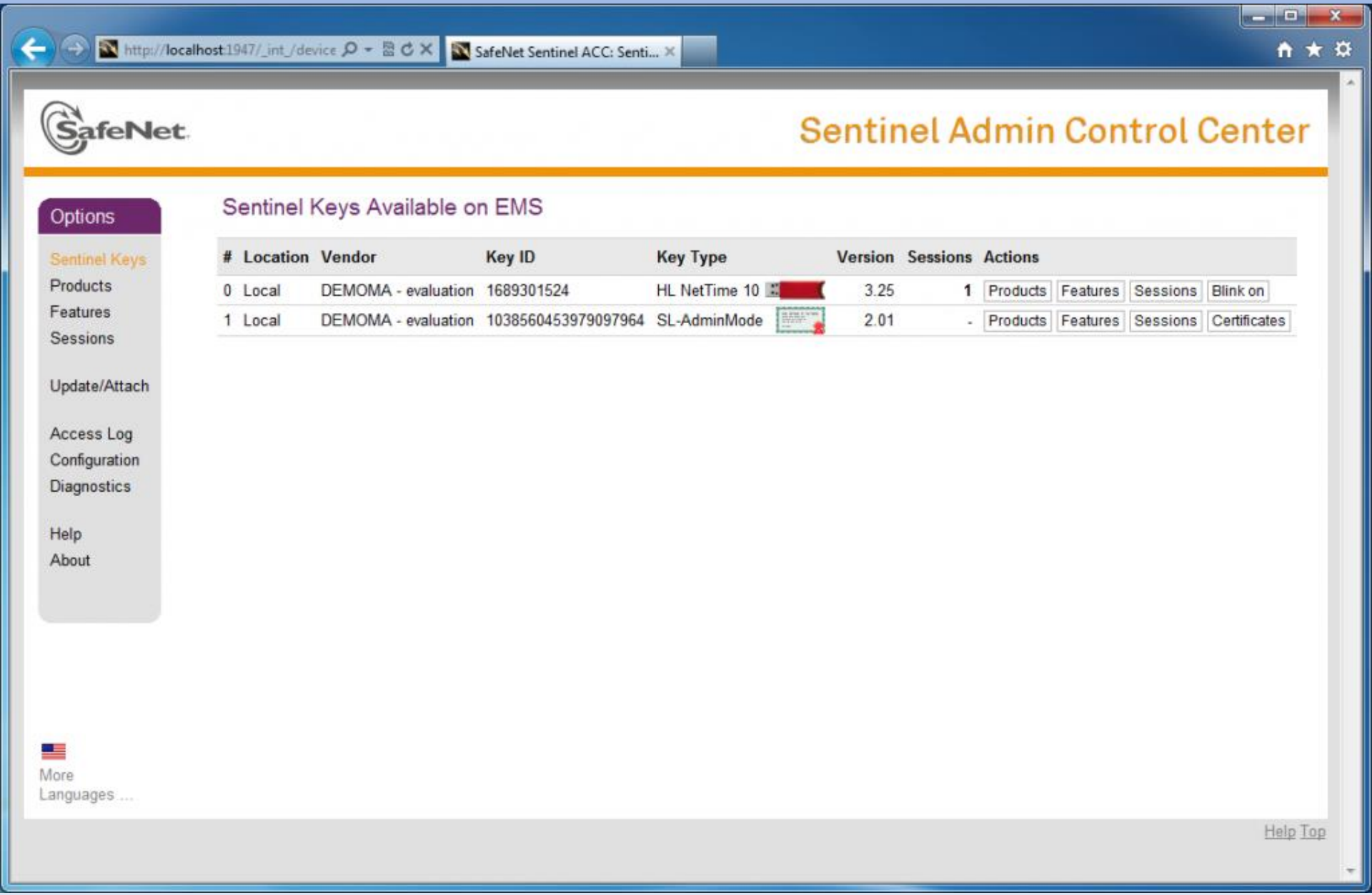
- Создание и удаленное обновление лицензий и демоверсий.
- Онлайн и оффлайн активация защищенного ПО.
- Регистрация пользователей на портале активаций.


http://localhost:8080/ems/e
Entitlement Management
Welcome admin | Logout | Change Password | Help
SentinelEMS
ENTITLEMENT MANAGEMENT SYSTEM
Entitlements
Customers
Reports
Catalog
Administration
Developer
Entitlements
Product Keys
Check In C2V
Check In Key
Recycle
EID
Batch Code:
DEMOMA
Type
Status
Customer
Created On
Draft
Parag Mahajan
6/5/2012
Draft
Sara Brown
6/10/2012
Acknowledged
John Smith
6/12/2012
Produced
Priya Shetye
6/4/2012
Product Keys Generated
Parag Mahajan
6/5/2012
Queued
Robert Jones
6/10/2012
Completed
Peter Adams
6/4/2012
Completed
Peter Adams
6/7/2012
Completed
Sara Brown
6/9/2012
Completed
Sara Brown
6/6/2012
Completed
John Smith
6/11/2012
Page Size 15
Total Records 11
Page 1 of 1
Entitlement Details (EID: 8a421811-5b40-4704-92a8-2fe572671614)
New
Actions
Customer Name: Parag Mahajan
E-mail: pmahajan@miain.com
Channel Partner: -
E-mail: -
Ref ID 1: 67433
Ref ID 2:
Start Date: 6/5/2012
End Date: Never expires
Status: Draft
Enable: Yes
Comments:
Entitlement Type: Protection Key Update
Number of updates: 3
Prompt for Confirmation: No
Product
Lock Type
Rehost
License Terms
SafeNet Memo Card Plugin
HL or SL-AdminMode
Disable
SafeNetCAD Home
HL or SL-AdminMode
Disable
SafeNetCAD Enterprise
SL-AdminMode
Disable
SafeNet © 2012 SafeNet, Inc. All Rights Reserved.

Admin Control Center

Средство администрирования лицензий на стороне конечного пользователя.

- Мониторинг лицензий.
- Управление использованием лицензий.
- Аудит использования лицензий.


http://localhost:1947/_int_/device
SafeNet Sentinel ACC: Senti...
SafeNet
Sentinel Admin Control Center
Options
Sentinel Keys
Products
Features
Sessions
Update/Attach
Access Log
Configuration
Diagnostics
Help
About
More Languages ...
Sentinel Keys Available on EMS
Location Vendor Key ID Key Type Version Sessions Actions
0 Local DEMOMA - evaluation 1689301524 HL NetTime 10 3.25 1 Products Features Sessions Blink on
1 Local DEMOMA - evaluation 1038560453979097964 SL-AdminMode 2.01 - Products Features Sessions Certificates
Help Top

Guardant

- Guardant Sign
- Высокопроизводительный кроссплатформенный USB-ключ с асимметричной криптографией, аппаратной реализацией AES и возможностью работы без драйверов. Позволяет автоматически защищать приложения, использующие WinAPI или платформу .NET. Имеет инструменты для защиты Java-приложений.

Guardant Sign

- Аппаратная платформа: 32-х разрядный микроконтроллер Cortex-M3, 4096 байт защищенной EEPROM.
- Криптографические алгоритмы: Электронная подпись на эллиптических кривых ECC160, Симметричное шифрование AES-128 и GSII64, Хэширование на базе SHA-256 или GSII64.
- Общие характеристики: Интерфейс USB 1.1 и выше, Размеры 58x16x8 мм, Масса 6,3г.

Guardant Sign

Функциональные возможности:

- Туннельное шифрование трафика протокола обмена.
- Работа в HID-режиме без дополнительных драйверов.
- Поддержка Windows, Windows CE, Linux.
- Автоматическая защита Win32 и .NET приложений.
- Инструменты для защиты Java-приложений.
- Псевдокод и обфускация компонентов защиты.
- Часы реального времени (в модификации Time).
- Flash-память 8 ГБ, 16 ГБ или 32 ГБ (в модификации Flash).

GUARDANT CODE

- Высокопроизводительный электронный ключ со встроенными криптографическими алгоритмами и 128 КБ памяти для загружаемого кода.
- Представляет собой доверенную аппаратную платформу, позволяющую выполнять произвольный пользовательский код внутри электронного ключа.
- Разработан для защиты, лицензирования и распространения кроссплатформенного программного обеспечения на Windows, Linux и OS X.

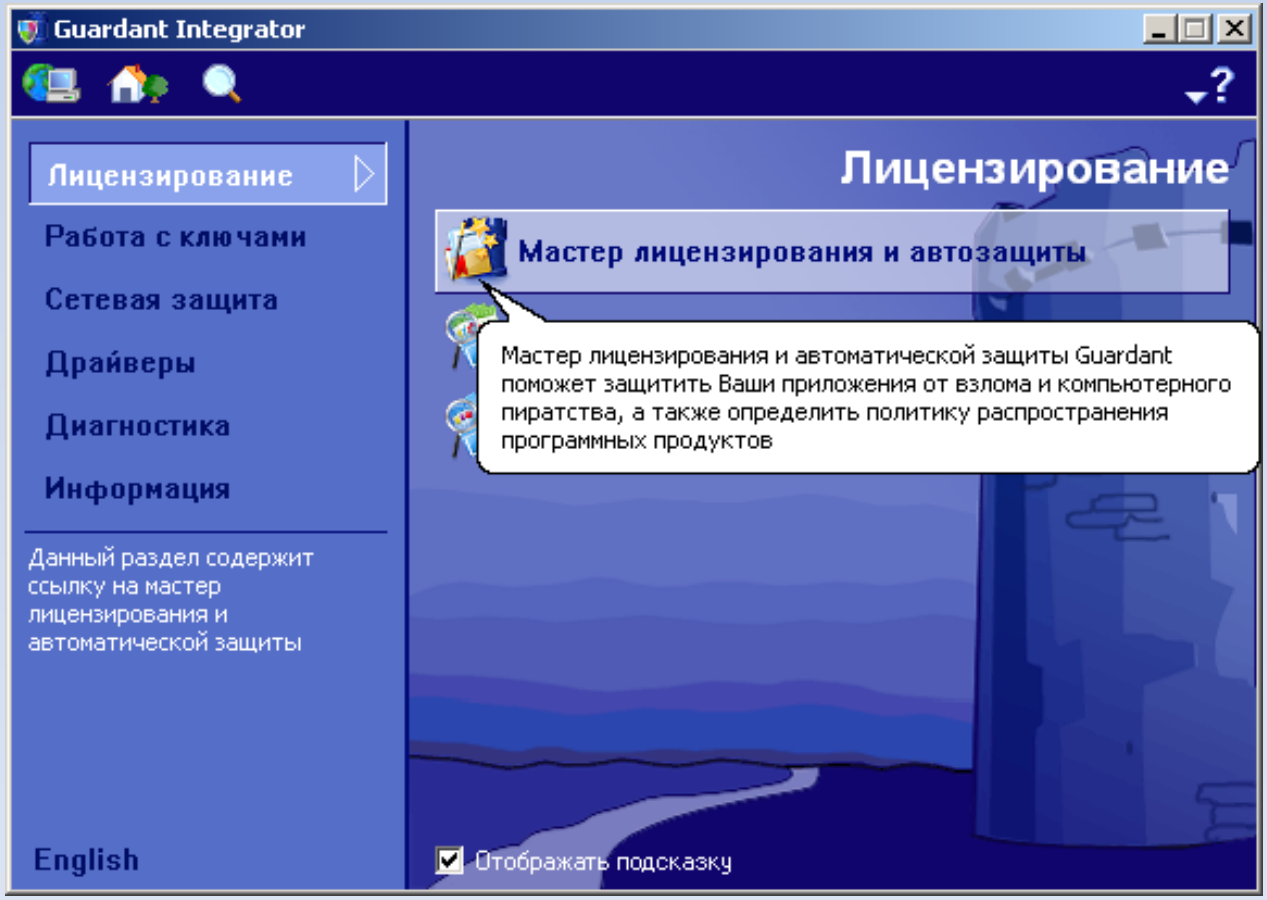
GUARDANT CODE

- Поддерживает мобильные платформы Android и Windows RT.
- Ключ выполнен на базе высокопроизводительно 32-разрядного ARM-микроконтроллера производительностью 120 MIPS.
- Позволяет работать без использования дополнительных драйверов.
- Загрузка и исполнение до 20 000 строк кода на С.

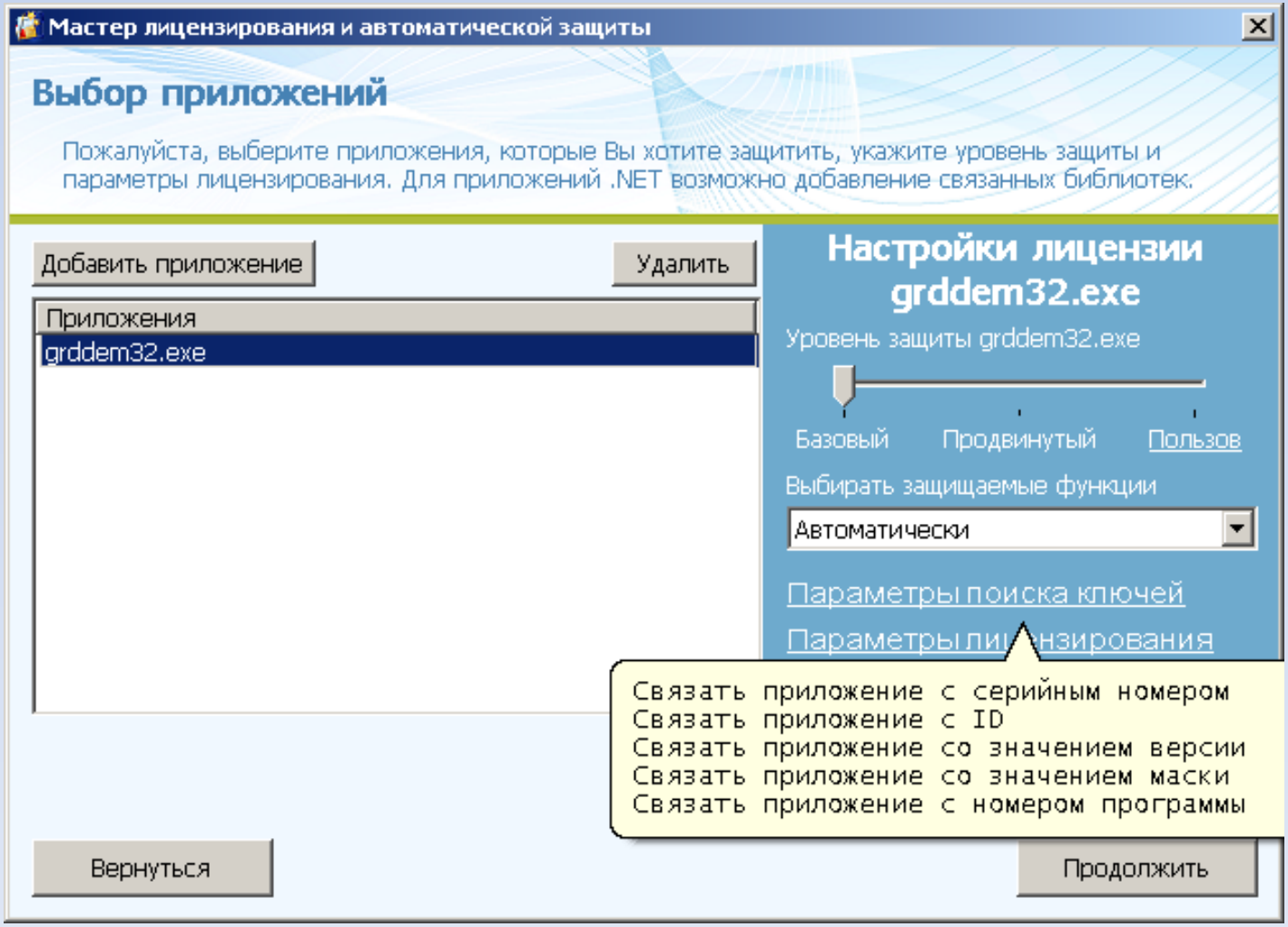
Guardant

- GUARDANT STEALTH II
- GUARDANT SD (Java апплеты)
- Guardant Sign Net (сетевые)
- Guardant SP (программный ключ, активируется через интернет)

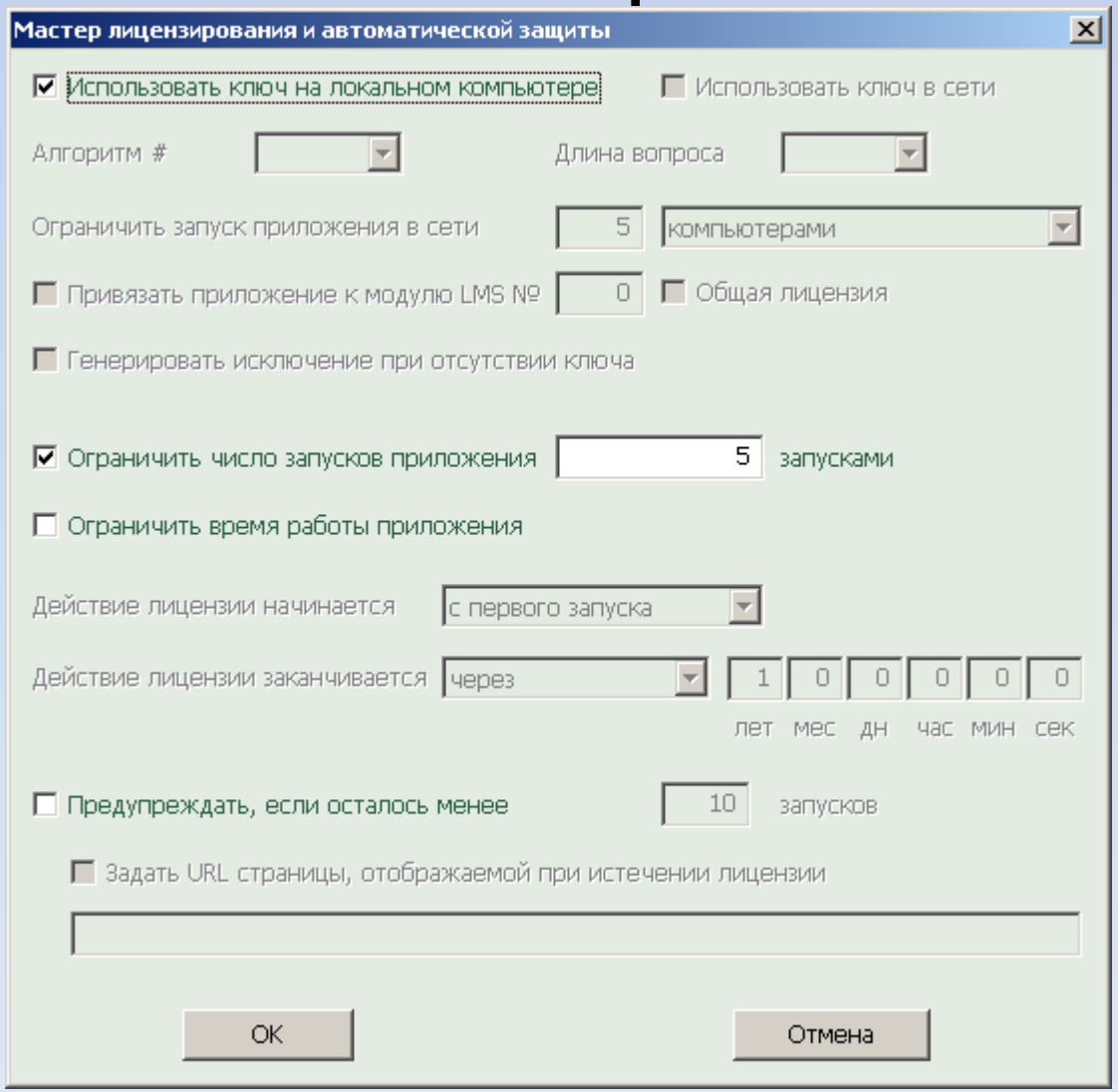
Guardant: Защита с помощью мастера



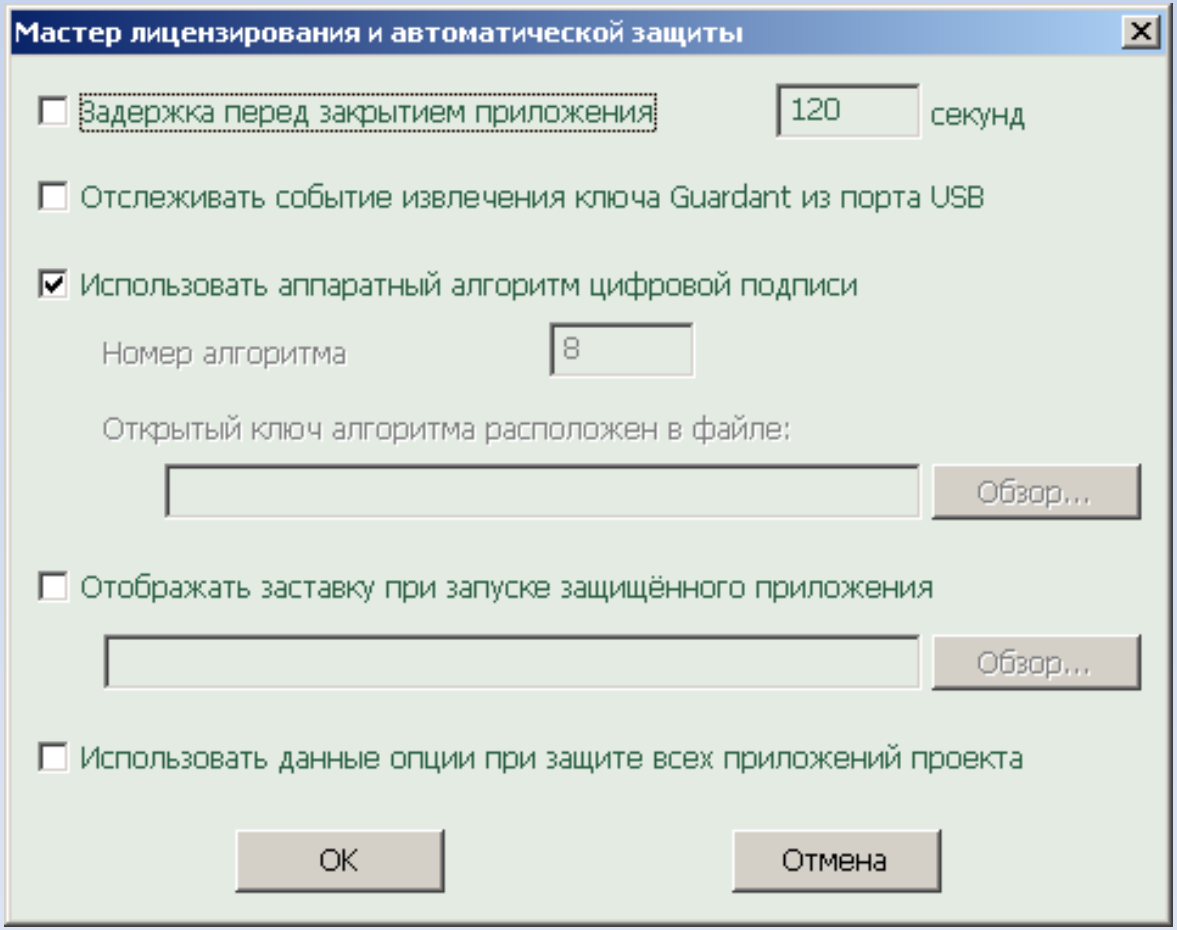
Guardant: Защита с помощью мастера



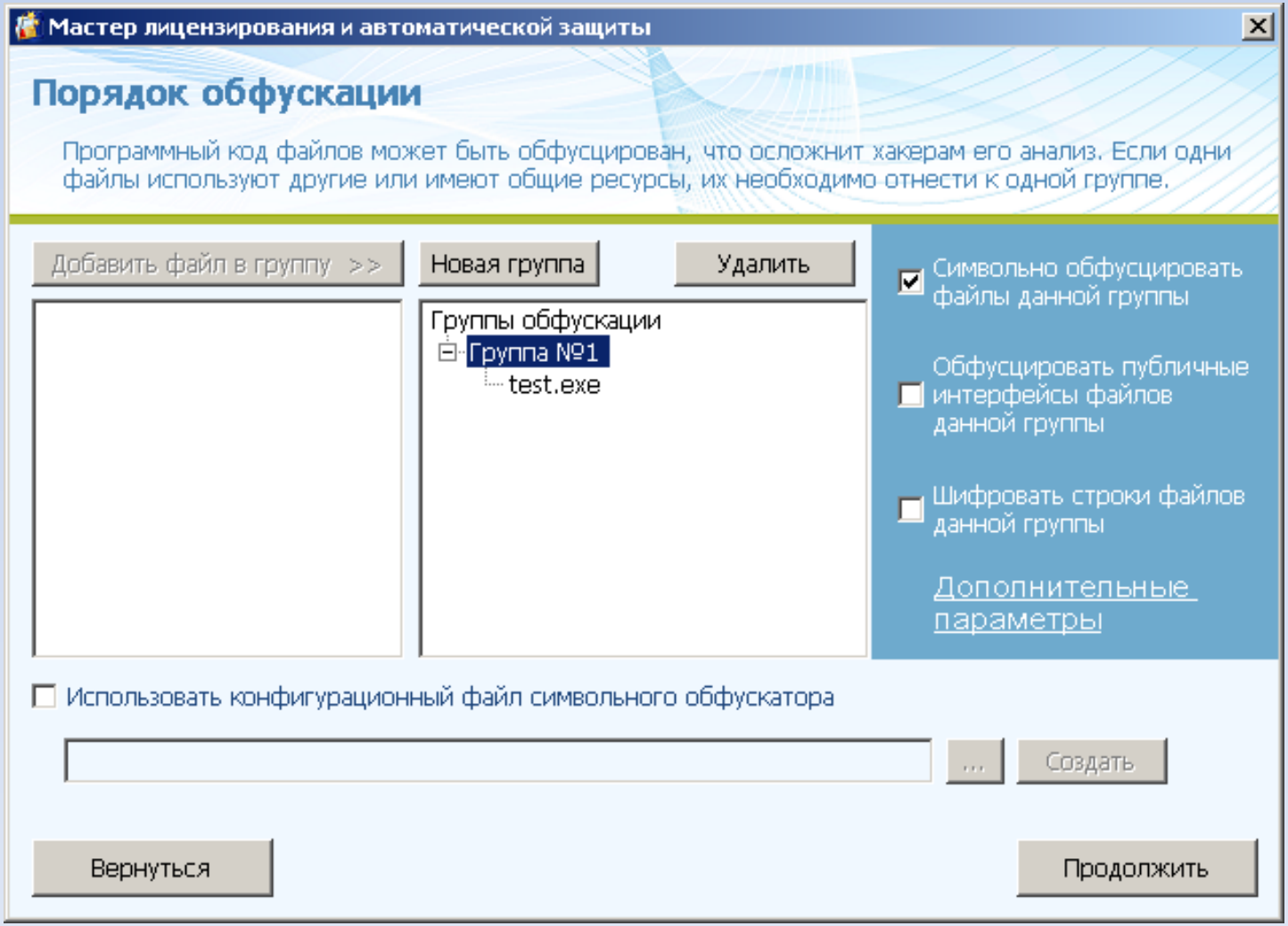
Guardant: Защита с помощью мастера



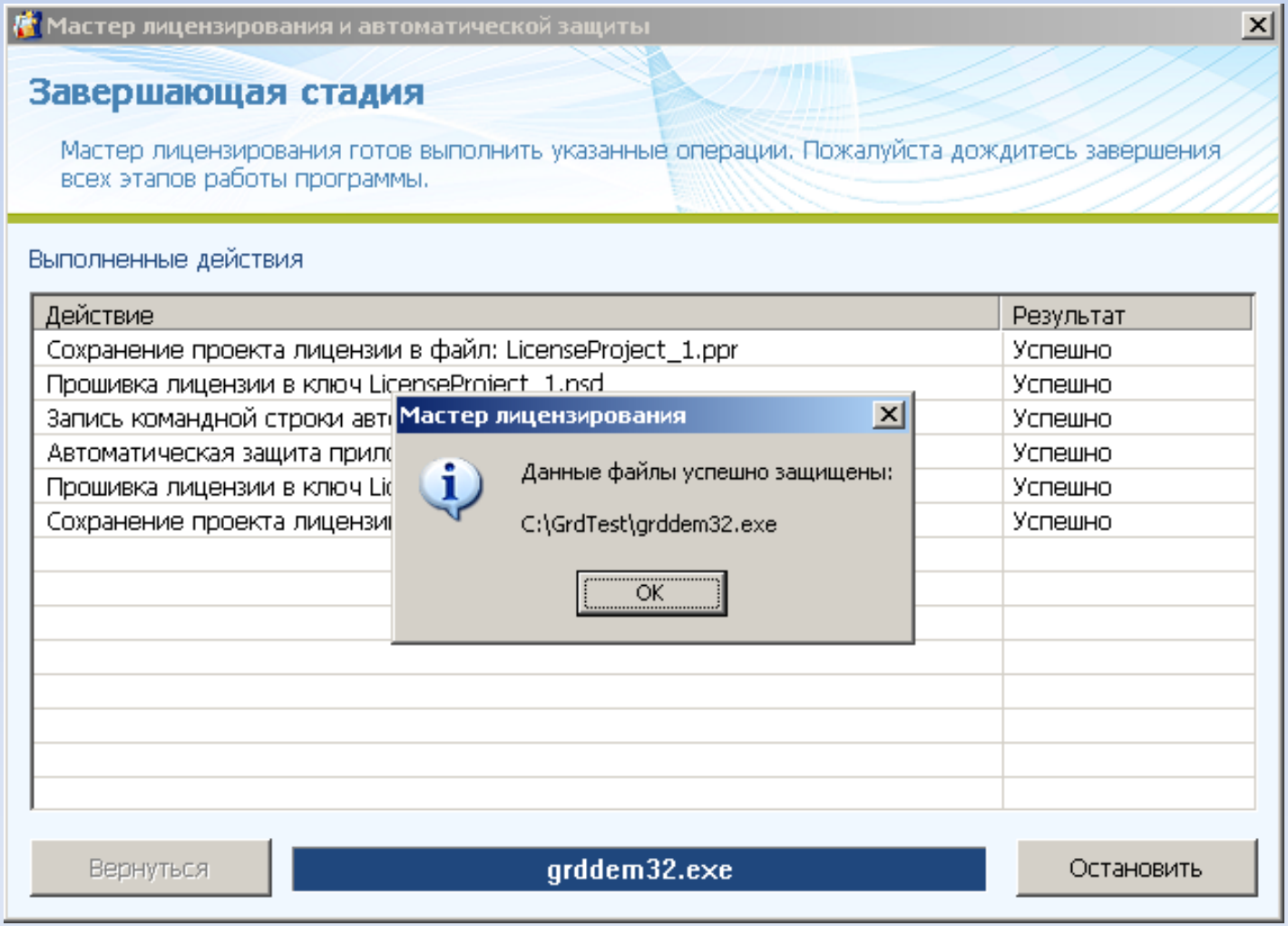
Guardant: Защита с помощью мастера



Guardant: Защита с помощью мастера



Guardant: Защита с помощью мастера



SenseLock

- В основе ключа находится высокозащищенный чип смарт-карты производства компании NXP (Philips), позволяющий исполнять произвольный код и хранить любые данные внутри. Для хранения кода и данных электронный ключ имеет энергонезависимую память объемом в 32 или 64 килобайта.

SenseLock: характеристики

- 16-битный высокозащищенный чип компании NXP (Philips), сертифицированный по стандарту EAL5+ и имеющий надежную защиту от декапсуляции и послойного сканирования.
- Модели с 32 и 64 килобайтами памяти для кода и данных. ~10,000 строк на языке Си.

SenseLock: характеристики

- Язык Си для разработки собственных функций, хранимых в ключе.
- Аппаратно реализованные алгоритмы RSA-1024, DES, Triple DES, алгоритм хэширования SHA-1 и генератор случайных чисел.

SenseLock: характеристики

- Ключ имеет уникальный 16-байтный серийный номер. Разработчик может установить свой собственный 8-байтный идентификатор для отдельного ключа или для целой серии.

SenseLock: модели

- SenseLock EL-Genii (USB, малый размер)
- SenseLock EL-STD (USB)
- SenseLock EL-RTC (часы – на батарейке)
- SenseLock EL-RTCC (часы + подзарядка)
- SenseLock EL-NET RTC (сетевая версия)
- SenseLock EL-STD-SSOP (микросхема)

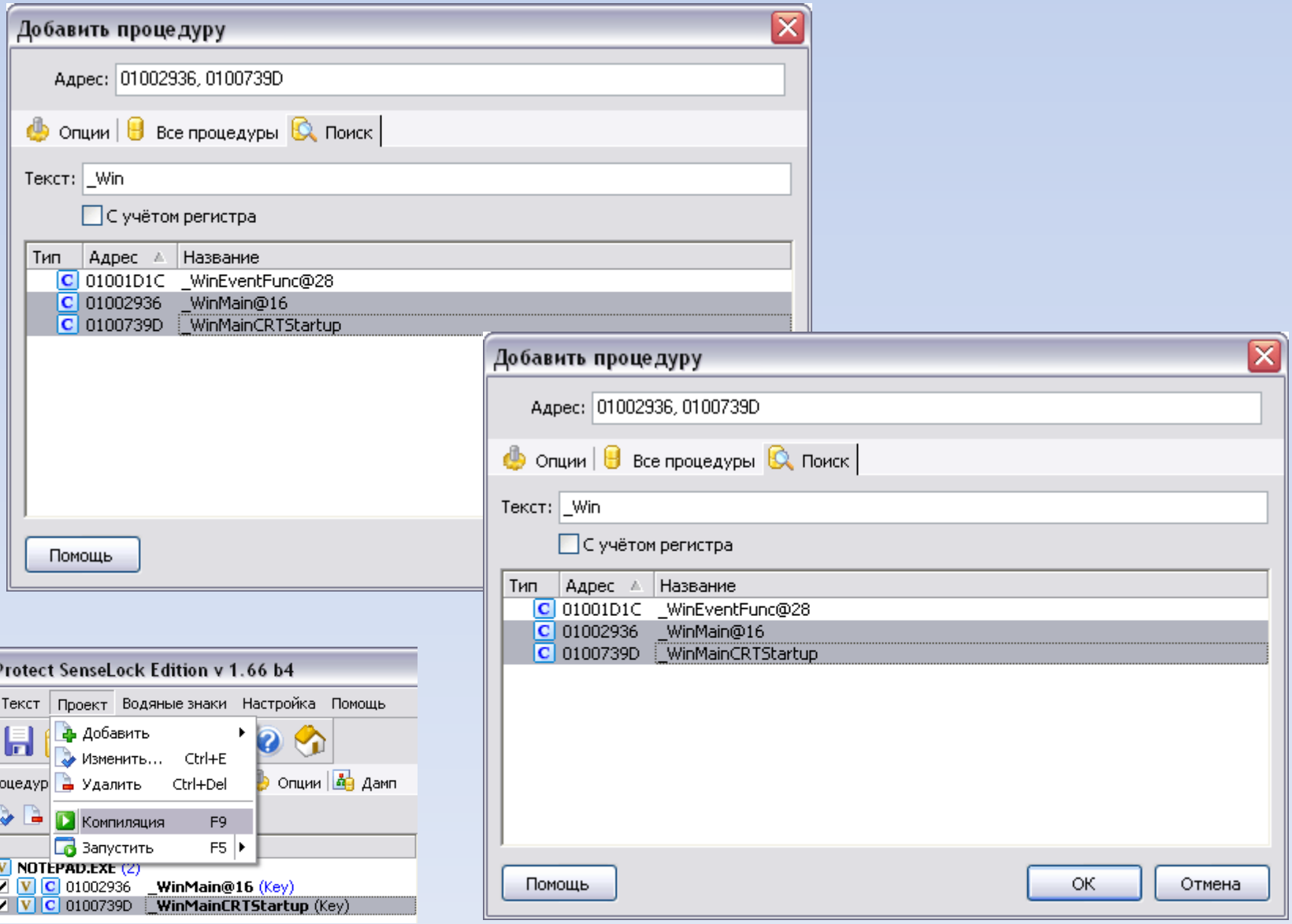





VMProtect SenseLock Edition

- виртуализация исполняемого кода;
- упаковка и шифрование защищаемого файла;
- выполнение кода защиты в ядре электронного ключа SenseLock (модель NET-RTC не поддерживается);
- возможность создавать демонстрационные лицензии, ограничивающие количество запусков, устанавливать ограничения по времени работы программы, лицензировать разные участки кода с привязкой к различным лицензиям;
- протокол обмена данными с электронным ключом на основе асимметричного алгоритма RSA-1024, исключающий появление эмуляторов.

VMProtect SenseLock Edition



Применяются

- Продукты 1С
- Autodesk (Autocad)
- 3D Studio Max
- и многие др.

Ключи защиты: выводы

- Аппаратные ключи не могут остановить пиратство.
- Часто используются автоматизированные средства защиты из SDK, которые могут быть нейтрализованы.
- Большая часть защитных механизмов, применяемых в ключах основана на предположении, что злоумышленник не может обеспечить эмуляцию ключа.

Идентификаторы и ключи защиты

Сходства:

- Имеют защищенную память.
- Могут выполнять криптографические функции

Рассмотренные вопросы

- Персональные идентификаторы
- Электронные ключи