

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

«Санкт-Петербургский государственный технологический институт  
(технический университет)»  
(СПбГТИ(ТУ))

---

Кафедра системного анализа

И.В. Ананченко

ИСПОЛЬЗОВАНИЕ КЛЮЧЕЙ СЕРИИ HASP HL ДЛЯ ЗАЩИТЫ  
ИНФОРМАЦИИ. ЗАЩИТА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Методические указания

Санкт-Петербург  
2012

УДК 681.3.657.1

Ананченко И.В. Использование ключей серии HASP HL для защиты информации. Защита программного обеспечения. - СПб.: изд. СПбГТИ (ТУ), 2012.-70 с.

В методических указаниях к трем лабораторным работам рассматриваются методы защиты программного обеспечения с использованием программно-аппаратных ключей HASP. Работы ориентированы на приобретение студентами навыков установки и эксплуатации защиты программных средств с использованием ключей серии HASP SRM. Методические указания соответствует содержанию дисциплин «Информационная безопасность и защита информации», «Программно-аппаратные средства защиты информации», «Защита информации» государственных образовательных стандартов. Позволяют формировать общепрофессиональные компетенции (ПК-2,4-7) по направлению подготовки 220100: способность применять перспективные методы системного анализа и принятия решений для исследования функциональных задач на основе мировых тенденций развития системного анализа, управления и информационных технологий.

Методические указания предназначены для бакалавров, магистров, аспирантов высших учебных заведений и могут быть использованы в системах непрерывного профессионального образования по компьютерным технологиям.

Ии.49 , библиогр. 8 назв. ПК-2,4-7.

Рецензент: И.А. Смирнов заместитель декана факультета Информационных технологий и управления СПбГТИ(ТУ), к.т.н., доцент кафедры систем автоматизированного проектирования и управления (САПРиУ)

Утверждено на заседании учебно-методической комиссии факультета информационных технологий и управления\_\_\_.2012

Рекомендовано к изданию РИСо СПбГТИ (ТУ)

## ВВЕДЕНИЕ

Программно-аппаратный комплекс HASP SRM предназначен для защиты от несанкционированного использования программного обеспечения. Sentinel HASP – решение, обеспечивающее аппаратную (USB-ключи HASP HL) и программную (HASP SL) защиту программного обеспечения. Более 75% разработчиков в мире [1] выбрали защиту программ Sentinel HASP благодаря простоте встраивания и надежности.

Преимущество HASP SRM перед другими решениями обеспечивается за счет повышенной надежности и гибкости. Два важных принципа лежат в основе HASP SRM:

- принцип разделения между бизнес-процессами и разработкой: Protect Once – Deliver Many;
- независимость от аппаратного обеспечения (Cross-Locking) – защищенное приложение может работать как с аппаратным ключом HASP HL, так и с программным аналогом HASP SL. Оба этих принципа являются взаимодополняющими.

Решения коммерческого характера, формирование программного пакета и определение условий лицензии являются прерогативой маркетингового отдела; эти процессы осуществляются независимо от защиты программного обеспечения (ПО).

Предлагаемая схема отличается повышенной гибкостью, предоставляя большую свободу в плане разработки новых стратегий продаж и моделей лицензирования. В HASP SRM поддерживается семь стандартных моделей лицензирования:

1. тестирование (оплата только после опробования);
2. аренда, прокат, лизинг;
3. оплата за отдельные модули;
4. оплата за отдельные функции;
5. сетевая лицензия;

6. оплата за фиксированный период с первого момента использования;
7. оплата по количеству запусков.

В комплект разработчика HASP SRM входит демонстрационная версия (HASP SRM Demo), которая может быть использована для выполнения лабораторных работ.

Предлагаемые практические работы могут выполняться учащимися под руководством преподавателя при изучении таких дисциплин как, например, «Информационная безопасность», «Защита информации», «Программно-аппаратные средства обеспечения информационной безопасности» или самостоятельно, желающими освоить технологию защиты программного обеспечения с использованием программных ключей HASP SL или аппаратных HASP HL.

Каждая практическая работа завершается списком контрольных вопросов и перечнем пунктов, которые должны быть отражены учащимися в отчетах по проделанной практической работе. Структура отчета уточняется преподавателем и зависит от названия читаемого курса, отводимого на изучение объема часов, требований учебного заведения и др.

При выполнении работ, учащиеся могут работать со стартовым комплектом с ключами той или иной серии, соответствующей конкретному разработчику, например, CRAXZ, OKZAI и пр. Альтернативный вариант проведения работ – использование комплекта разработчика DK Sentinel HASP и демонстрационного ключа Demo HASP HL Max, Time, Net10 или NetTime10.

Комплект разработчика Sentinel HASP (входит в стартовый комплект) включает в себя:

- все программное обеспечение Sentinel HASP;
- полную документацию Sentinel HASP на русском языке;
- примеры встраивания защиты;
- бесплатную техническую поддержку;
- Бесплатное обновление ПО.

Комплект разработчика Sentinel HASP доступен для бесплатной загрузки с сайта производителя средств защиты. Там же можно купить демонстрационные и полнофункциональные ключи HASP HL, а так же Стартовые комплекты [2]. Покупка средств защиты HASP SRM возможна у вендоров, в том числе оформление заказов в интернет магазинах [3].

Для выполнения работ потребуется программное обеспечение комплект разработчика HASP, которое при необходимости можно бесплатно загрузить с сайта [6]. Ключи серии HASP предназначены, в первую очередь, для защиты программного обеспечения. Дополнительно могут решаться и другие задачи, например, использование защищенной памяти ключа HASP HL (весь модельный ряд, кроме HASP HL Basic) для хранения конфиденциальной информации, используемой для процедуры аутентификации [6-8] и пр.

## Ключи HASP SRM

### Распределение ролей для пользователей HASP SRM

Все функции и задачи в HASP SRM условно распределяются по ролям, назначаемым администратором HASP SRM. Рассмотрим их подробнее:

- пользователь с ролью Product Management (обычно эта роль сотрудника отдела продуктов) может определить компоненты и продукты;
- пользователи, обладающие ролью Development (разработка), имеют права по защите и лицензированию программного обеспечения, доступен просмотр и исполнение заказов;
- пользователь с назначенной ролью Customer Services может просматривать и редактировать сведения о клиентах, вручную активировать продукт;
- роль Order Management (отдел продаж) позволяет ее владельцу редактировать, обрабатывать и передавать заказы на исполнение;

- пользователи, владельцы роли Production (производство), имеют права на просмотр и исполнение заказов.

В пособии рассмотрены основные этапы установки программного обеспечения (ПО) HASP SRM Vendor Suite для работы с аппаратными ключами HASP HL и программными HASP SL; последовательность типовых действий установки и менеджмента защищаемого программного обеспечения с использованием технологии HASP SPM:

- *Этап 1 – Подготовительный этап.* Содержит инструкции по установке системы HASP SRM и запуску HASP SRM Vendor Suite.
- *Этап 2 – Определение компонентов ПО.* Перед установкой защиты определяем приложения, как отдельные компоненты.
- *Этап 3 – Установка защиты.* Выполняем установку защиты на приложения и компоненты, определенные на предыдущем этапе.
- *Этап 4 – Формирование программного пакета.* Создание готового программного пакета на базе имеющихся компонентов и приложений.
- *Этап 5 – Создание условно-бесплатной версии.* Создание в HASP SRM Business Studio условно-бесплатной версии для одного из компонентов программного продукта.
- *Этап 6 – Оформление и обработка заказов.* Оформление заказов на созданные продукты и внесение данных о поставщиках. Создаем обновления лицензий для установленных продуктов.
- *Этап 7 – Работа с защищенным приложением на стороне пользователя.* Изучение работы с приложениями, защищенными с помощью HASP SRM, на стороне пользователя.

Чтобы в полной мере оценить все преимущества технологии Protect Once – Deliver Many, необходимо последовательно изучить и выполнить на практике все этапы, начиная с первого.

## **ЗАЩИТА ИСПОЛНЯЕМЫХ EXE ФАЙЛОВ С ПОМОЩЬЮ НАВЕСНОЙ ЗАЩИТЫ HASP**

### **Цель работы**

Приобретение практических навыков работы с HASP ключами и программным обеспечением HASP SRM Vendor Suite. Защита исполняемых exe файлов

#### Этап 1. Подготовительный

Задача	Настроить компьютер и систему HASP SRM для выполнения практических работ
Используемые приложения	<ul style="list-style-type: none"><li>• Установочный диск HASP SRM</li><li>• HASP SRM Vendor Suite</li></ul>

#### Содержание этапа:

- установка программного пакета HASP SRM, в том числе HASP SRM Business Studio Server;
- запуск HASP SRM Vendor Suite.

Для установки программного пакета HASP SRM необходимо выполнить действия:

1. вставить диск с дистрибутивом HASP SRM в CD-привод;
2. выбрать в меню пункт Start the CD Setup (Установить с диска);
3. выполнить установку, руководствуясь инструкциями программы-мастера.

Запуск HASP SRM Vendor Suite. После установки программного пакета HASP SRM необходимо запустить HASP SRM Vendor Suite:

в меню Start (Пуск) выбрать Programs (Программы) → Aladdin → HASP SRM → Vendor Suite. На экране появится приветственное окно HASP SRM Vendor Suite, которое открывает доступ почти ко всем приложениям из состава Vendor Suite.

Важно: нельзя подсоединять к USB порту компьютера ключ HASP SRM, до тех пор, пока полностью не установлен программный пакет HASP SRM.

## Этап 2. Создание компонентов

Задача	Освоить процессы, связанные с созданием программных компонентов
Роль	Отдел продуктов
Используемые приложения	HASP SRM Business Studio

Компоненты выделяются по их функциям и сами по себе могут являться предметом лицензирования. В HASP SRM под компонентом понимается отдельное приложение, его составные компоненты или отдельные функции, например, «Сохранить в файл», «Печать» и т.д.

На данном этапе человек, выполняющий работу, действует в роли менеджера по продуктам, его задача состоит в том, чтобы с помощью HASP SRM Business Studio определить два компонента, которые понадобятся для выполнения следующих этапов.

Для запуска HASP SRM Business Studio следует выполнить действия:

1. в меню Start (Пуск) выбрать Programs (Программы) → Aladdin → HASP SRM → Vendor Suite;
2. в списке приложений выбрать Business Studio. На экране появится окно авторизации;
3. в поле User Name (Имя пользователя) и в поле Password (Пароль) ввести HASP;
4. в поле Server Address ввести имя компьютера;
5. в поле Port (Порт) ввести 443;
6. Нажать ОК. На экране появится главное окно HASP SRM Business Studio.

Главное окно HASP SRM Business Studio (рис. 1) условно делится на три части:

- панель функций. Необходимо выбрать те функции, которые нужно выполнить;



- основное окно. Представлены основные данные по выбранному разделу;
- панель задач. После того как выбрали тот или иной объект в основной части окна, можно производить над ним определенные действия.

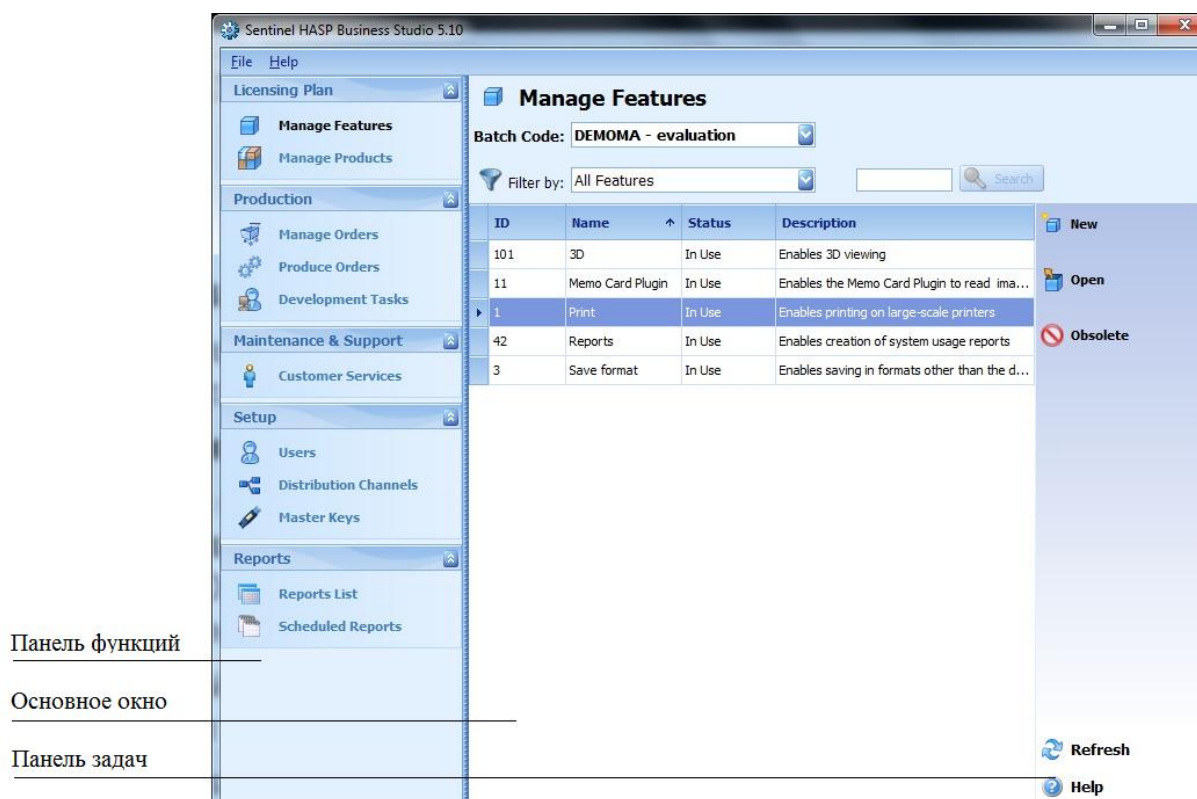


Рисунок 1 – главное окно HASP SRM Business Studio

Рассмотрим выделение программы как отдельного компонента, выполняющей расчет по формуле  $A+B=C$  (написанной на языке Delphi).

Для определения компонента  $A+B=C$  выполним следующие действия:

1. На панели функций в окне Business Studio открываем раздел Licensing Plan (Лицензионный план) и выбираем в нем пункт Manage Features (Работа с компонентами). Основная часть окна отводится под перечень компонентов, справа представлены доступные действия;
2. В основной части окна выбрать поле Batch Code (Код серии), далее, из списка код серии, определенный преподавателем, если таких кодов несколько (или код вашей серии разработчика, например, CRAXZ). Далее для конкретизации везде используется код серии разработчика CRAXZ. При выполнении работы следует, выполняя предлагаемые действия, использовать код серии, с которой

работаете. Для того чтобы появилась возможность выбрать Vendor Key с именем CRAZXZ (рис. 2), необходимо на панели функций в окне Business Studio открыть раздел Setup (Установки) и выбрать в нем пункт Master Keys (Мастер ключи);



Рисунок 2 – раздел Setup

затем в панели задач выбрать Introduce Master (Ввести Мастер ключ), появится окно (рис. 3):

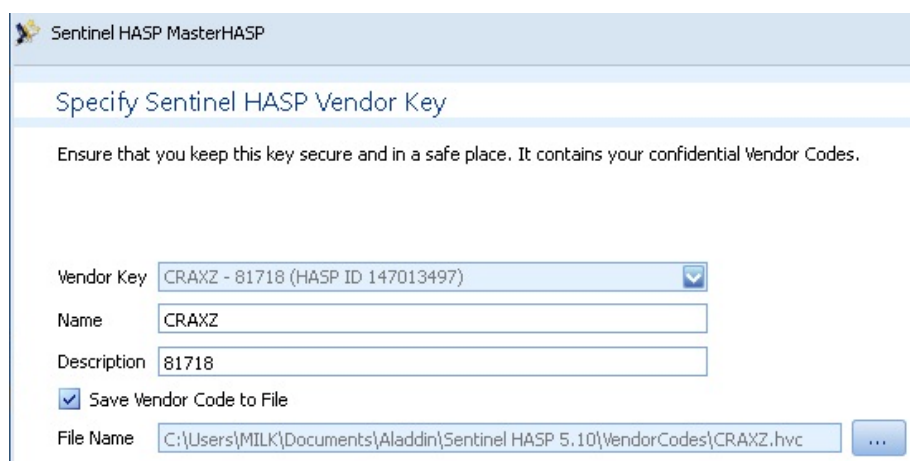


Рисунок 3 – Sentinel HASP Master HASP

Определить имя ключа (например, CRAZXZ) и указать путь, куда будет сохранена информация из мастер ключа (рис. 3). После обработки информации о ключе, информация будет отображена в поле Batch Code.

3. В панели задач следует щелкнуть на New Feature (Создать компонент). Появится новое окно New Feature (Создание компонента);
4. в поле Feature Name (Название компонента) ввести название  $A+B=C$ ;
5. Для дополнительных настроек, следует нажать кнопку Advanced (Дополнительно);

## Specify API Settings

Select the libraries for which you want to generate APIs. By default, the wizard generates a separate API for each Batch Code.

Close any running Vendor Suite application before generating APIs.

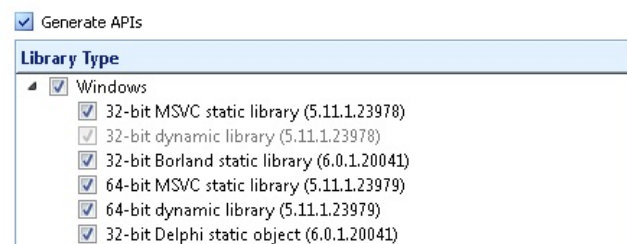


Рисунок 4 – API Settings



Рисунок 5 – окно New Feature (A+B=C)

6. Необходимо установить флажок в пункте Specify Feature ID (Номер компонента) и ввести число 50;
7. Затем нажать ОК, после этого в главном окне можно увидеть в списке компонентов новый компонент с названием A+B=C.

## Выделение приложения mspaint как отдельного компонента

Рассмотрим последовательность действий по созданию компонента mspaint. Для этого необходимо выполнить действия:

1. На панели функций в Business Studio открыть раздел Licensing Plan (Лицензионный план) и выбрать пункт Manage Features (Управление компонентами). Основная часть окна отводится для списка компонентов;
2. В основной части окна выбрать поле Batch Code (Код серии), выбрать из списка код серии CRAXZ;
3. В панели задач щелкнуть на New Feature (Создать компонент). Появится новое окно New Feature (Создание компонента);
4. В поле Feature Name ввести mspaint;
5. Установить флажок в пункте Specify Feature ID (Номер компонента) и ввести 10;
6. Нажать ОК. В главном окне можно увидеть в списке компонентов новый компонент с названием mspaint (рис. 6);



Рисунок 6 – новый компонент с названием mspaint

## 7. Закрыть Business Studio.

Если пункт Specify Feature ID отсутствует в окне, необходимо нажать кнопку Advanced (Дополнительно); если на экране появляется окно с сообщением о том, что компонент с указанным номером (Feature ID) отсутствует, не нужно указывать другой номер. Следует закрыть окно New Feature (Создание компонента) и в окне Manage Features (Управление компонентами) найти компонент с номером 10 и посмотреть, какое название имеет этот компонент. Его необходимо будет указывать при выполнении следующих пунктов (вместо mspaint). Можно проверить, занят ли указанный номер (Feature ID), нажав на кнопку (ID Available), и если указанный номер (Feature ID) не занят, то можно будет увидеть следующее сообщение;

### Этап 3. Установка защиты ПО.

Задача	Освоить защиту приложений с использованием HASP SRM в автоматическом режиме
Роль	Отдел разработки
Используемые приложения	HASP SRM Envelope

Рассмотрим работу с приложением HASP SRM Envelope. Необходимо настроить защиту двух приложений, определенных на предыдущем этапе.

В HASP SRM Envelope используется новая технология защиты программных продуктов от незаконного или несанкционированного использования. При каждом запуске защищенное приложение обращается к определенному ключу HASP SRM и запрашивает с него необходимые данные. Если в этот момент ключ HASP SRM не подсоединен к компьютеру или запрашиваемые данные некорректны, приложение не запускается или перестает работать.

#### **Установка защиты на приложение $A+B=C$**

Необходимо запустить приложение HASP SRM Envelope, подсоединить ключ HASP, отредактировать сообщение, которое будет появляться при отсоединении ключа HASP HL и установить защиту на приложение  $A+B=C$ . Интерфейс HASP SRM Envelope:

- дерево функций: представлены все доступные функции;
- основная область: в этой части окна отображается информация, связанная с разделом, выбранным слева в дереве функций;
- журнал событий: здесь отображается информация о событиях HASP SRM Envelope.

Перед установкой защиты необходимо:

1. подсоединить ключ CRAXZ;
2. открыть меню Start (Пуск) и в нем выбрать пункт Programs (Программы) → Aladdin → HASP SRM → Vendor Suite;

3. в списке приложений выбрать Envelope. Появится окно авторизации;
4. нажать кнопку Work Offline (Работать автономно). Откроется главное окно HASP SRM Envelope;

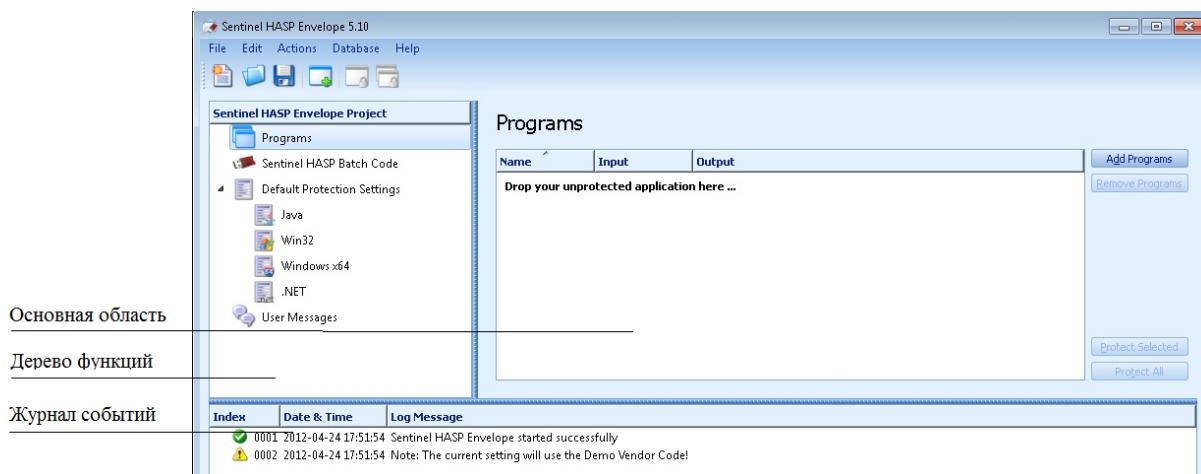


Рисунок 7 – интерфейс HASP SRM Envelope

5. в дереве функций выбрать раздел Sentinel HASP Batch Code (Профиль HASP SRM);
6. в правой части окна выбрать пункт Use Vendor Code from file (Использовать файл с Кодом разработчика);

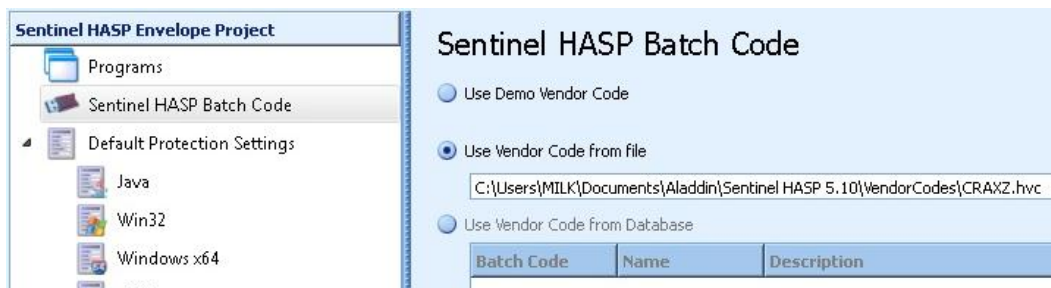


Рисунок 8 – раздел Sentinel HASP Batch Code

7. открыть папку, где хранится файл с кодом разработчика. По умолчанию он находится в: ... \Documents and Settings\[имя текущего пользователя]\My Documents\Aladdin\HASP SRM [версия]\VendorCodes
8. затем необходимо выбрать файл CRA\XZ.hvc и нажать кнопку Open (Открыть). Путь к файлу отобразится в поле Use Vendor Code from file (Использовать файл с Кодом разработчика).

Для установки защиты на приложение  $A+B=C$  необходимо:

1. В дереве функций выбрать раздел Programs (Программы);
2. Справа вверху нажать на кнопку Add Programs (Добавить программы). Появится окно Add Programs (Добавить программы), в котором будет представлено содержимое папки VendorSuite\samples;
3. Выбрать файл A+B=C.exe (файл может находиться в другой папке, в этом случае выяснить место нахождения файла у преподавателя) и нажать Open (Открыть). Теперь это приложение появится в списке приложений в ветви Programs (Программы);

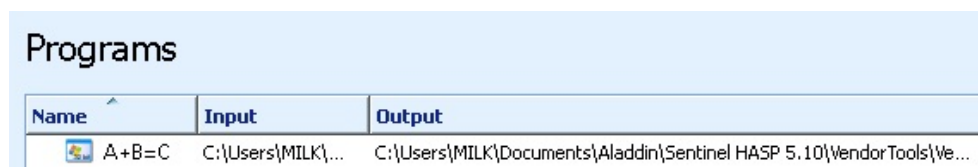


Рисунок 9 – приложение A+B=C в ветви Programs (Программы).

4. В дереве функций выбрать раздел Programs (Программы) и затем A+B=C. В основной части окна отображаются параметры защиты;

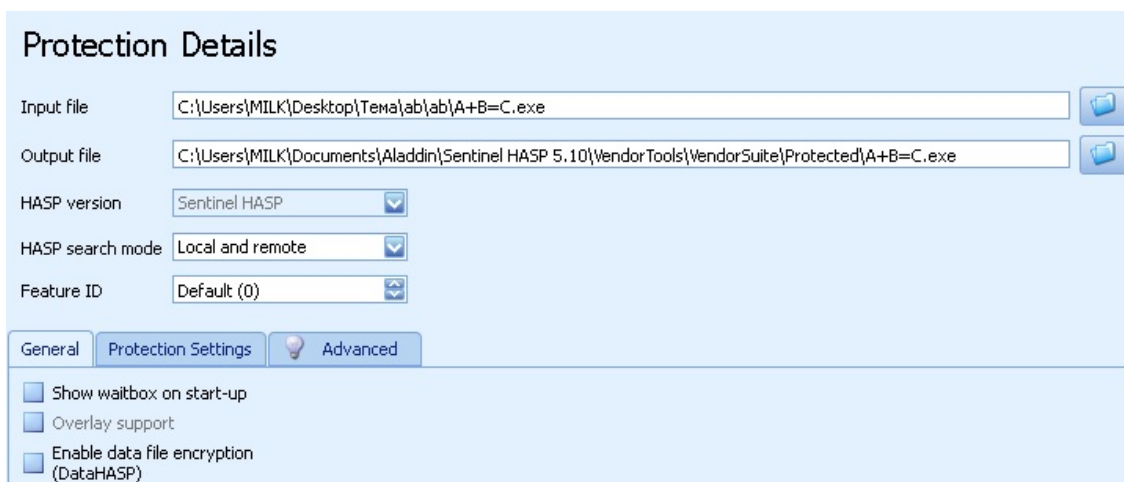


Рисунок 10 – параметры защиты

5. В поле Feature ID (Номер компонента) необходимо указать значение 50 (значение было определено ранее на этапе 2);
  6. В дереве функций выбрать раздел User Messages (Сообщения для пользователя);
- В открывшемся окне можно ввести текст сообщений, которые будут отображаться на стороне пользователя.



8. В списке Message Code выбрать HASP\_HASP\_NOT\_FOUND. Это сообщение будет отображаться, если необходимый ключ HASP SRM отсутствует;

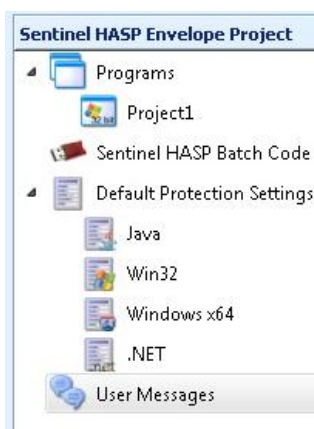


Рисунок 11 – раздел User Messages

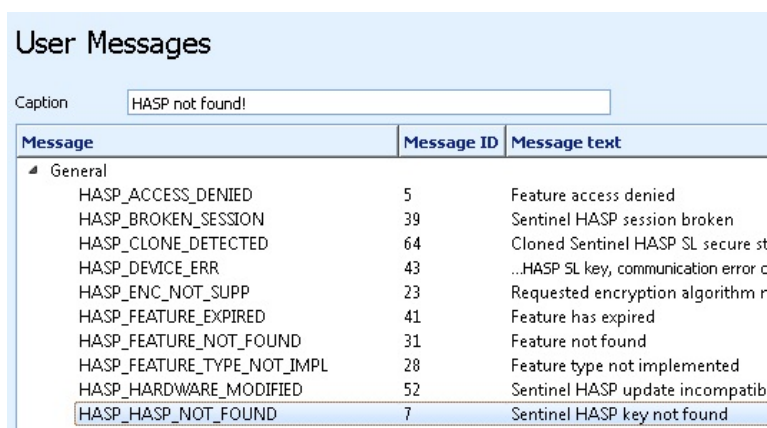


Рисунок 12 – сообщение HASP\_HASP\_NOT\_FOUND

8. в поле Message (Сообщение) вместо HASP not found необходимо ввести текст сообщения об ошибке (например: Для запуска приложения  $A+B=C$  требуется ключ HASP SRM);
9. в дереве функций выбрать раздел Default Protection Settings (Настройки шаблона защиты) и в основной части окна выбрать Win32. После этого в основной части окна появятся параметры защиты для приложений Win32;.
10. параметр HASP key Periodic background checks определяет интервал времени между проверками на наличие ключа HASP SRM. Требуется установить значение поля «5». Защищенное приложение будет каждые пять секунд проверять наличие ключа HASP HL;





Рисунок 13 – раздел Default Protection Settings

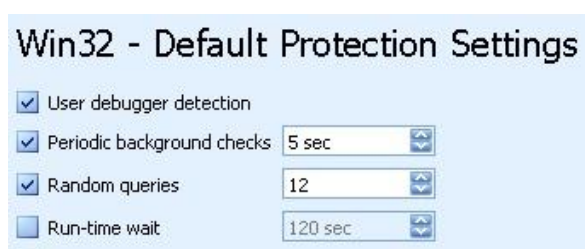


Рисунок 14 – параметр HASP key Periodic background checks

11. в дереве функций выбрать раздел Programs (Программы) и выбрать A+B=C. Затем справа нажать кнопку Protect (Установить защиту). Перед началом установки защиты, появится сообщение об этом;

12. по завершению процесса появится сообщение о том, что на выбранное приложение установлена защита. Нажать ОК. Путь к защищенному приложению таков: ... \Documents and Settings\[имя текущего пользователя]\My Documents\Aladdin\HASP SRM [версия]\VendorCodes\ VendorSuite\Protected.

### Установка защиты на приложение mspaint

Необходимо создать копию приложения mspaint (Paint) из стандартного пакета программ Windows, установить параметры защиты для ранее созданного компонента mspaint и установить защиту.

Для защиты приложения mspaint необходимо:

1. открыть папку с приложением mspaint (как правило, оно находится в ...\\Windows);
2. создать копию приложения mspaint;
3. необходимо убедиться в том, что ключ с кодом CRAXZ подсоединен к компьютеру;
4. запустить приложение HASP SRM Envelope, если оно было закрыто. В окне авторизации нажать кнопку Work Offline (Работать автономно). Появится главное окно HASP SRM Envelope;
5. в дереве функций выбрать раздел Programs (Программы);
6. справа вверху нажать на кнопку Add Programs (Добавить программы);
7. в открывшемся окне указать папку, где находится файл mspaint.exe. Выбрать файл и нажать кнопку Open (Открыть). В ветви Programs (Программы) появится приложение mspaint;
8. в дереве функций раскрыть ветвь Programs (Программы) и выбрать mspaint. В основной части окна указаны параметры защиты приложения;
9. в поле Feature ID (Идентификатор компонента) установить значение 10 (это значение было указано ранее на этапе 2);

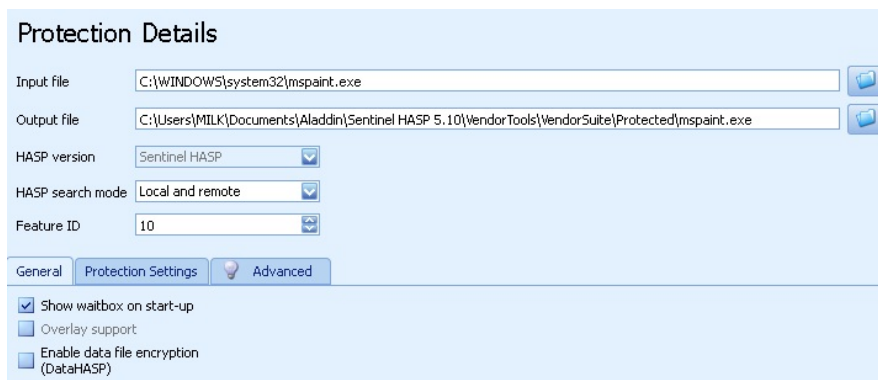


Рисунок 15 – protection Details

10. в дереве функций выбрать раздел Default Protection Settings (Настройки шаблона защиты) и в основной части окна выбрать Win32. В основной части окна можно увидеть параметры защиты для приложений Win32;
11. в окне с настройками шаблона защиты переместить бегунок Encryption level (Уровень шифрования) в крайнее правое положение. Чем выше уровень

шифрования, тем выше уровень защиты приложения, однако при этом скорость загрузки приложения может несколько увеличиться, так как для запуска потребуется больше времени на то, чтобы расшифровать файл;

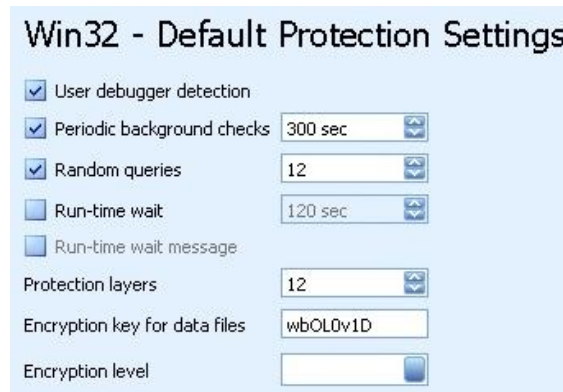


Рисунок 16 – Default Protection Settings

12. исходя из этого, следует предусмотреть сообщение, которое будет сопровождать процесс загрузки приложения. Для этого в настройках защиты (Protection Details) необходимо поставить флажок в пункте Show waitbox on startup (Показывать сообщение о загрузке);

13. нажать кнопку Protect (Установить защиту);

14. по завершению процесса появится сообщение о том, что на выбранное приложение установлена защита. Нажать ОК. Путь к защищенному приложению: ...\\Documents and Settings\\[имя текущего пользователя]\\My Documents\\Aladdin\\HASP SRM 2.10\\ VendorCodes\\VendorSuite\\Protected;

15. закрыть приложение HASP SRM Envelope. Перед выходом будет предложено сохранить проект;

16. Нажать кнопку Save (Сохранить).

#### Этап 4. Формирование программного пакета

Задача	Рассмотреть, что необходимо выполнить, чтобы создать пакет для последующего лицензирования
Роль	Отдел продуктов
Используемые приложения	HASP SRM Business Studio

Под продуктом в данном контексте имеется в виду программный пакет, который включает в себя данные или приложения, предназначенные для

платного распространения. Лицензионные ограничения устанавливаются для каждого компонента отдельно.

На этом этапе рассмотрим процесс создания продуктов. В них будут входить компоненты  $A+B=C$  и `mspaint`, которые были созданы на этапе 2, и каждый из этих компонентов должен иметь свои лицензионные ограничения. Далее описан процесс, как обрабатывать заказы на созданные продукты.

В этом этапе рассмотрено создание трех разных продуктов:

- “`A+B=C_Lite`” – продукт для использования на условиях аренды, действие лицензии ограничивается тремя запусками приложения;
- “`A+B=C_Full`” – полнофункциональный продукт, который по условиям лицензии не имеет никаких ограничений на использование составляющих компонентов. В качестве основы для данного продукта будем использовать “`A+B=C_Lite`”;
- “`mspaint_Full`” – полнофункциональный продукт, который распространяется без ограничений на использование составляющих компонентов.

Примечание: установка и запуск лицензионных продуктов рассмотрены на этапе 7.

### **Создание продукта с ограничением по количеству запусков**

Приводятся инструкции, позволяющие создать продукт из компонента  $A+B=C$  (номер: 50). Лицензия на продукт будет действовать в течение трех запусков. Если пользователь захочет продолжить работу с этим продуктом после трех запусков, ему будет предложено обновить лицензию.

Чтобы создать новый продукт, необходимо:

1. запустить HASP SRM Business Studio;

2. в панели функций выбрать раздел Licensing Plan (Лицензионный план) и выбрать в нем пункт Manage Products (Управление продуктами). В основной части окна представлена информация о доступных продуктах;

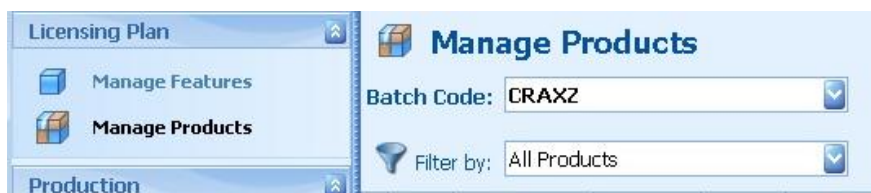


Рисунок 17 – Manage Products

3. в поле Batch Code (Код серии) выбрать код серии CRAXZ;

4. в панели задач нажать New Product (Создать продукт);

5. в основной части окна в поле Product Name (Название продукта) ввести A+B=C\_Lite;

6. в свойствах продукта (Attributes) открыть список Locking type (Тип используемого ключа) и выбрать HASP HL or HASP SL (HASP HL или HASP SL). Этот параметр используется при оформлении заказов для того, чтобы установить, с каким ключом будет использоваться данный продукт.

7. в настройках Features & Memory (Компоненты и память) в списке Available Features (Доступные компоненты) выбрать компонент с номером 50 и щелкнуть на стрелку “вправо”. Выбранный компонент появится в списке компонентов входящих в состав данного продукта (Features In Product);

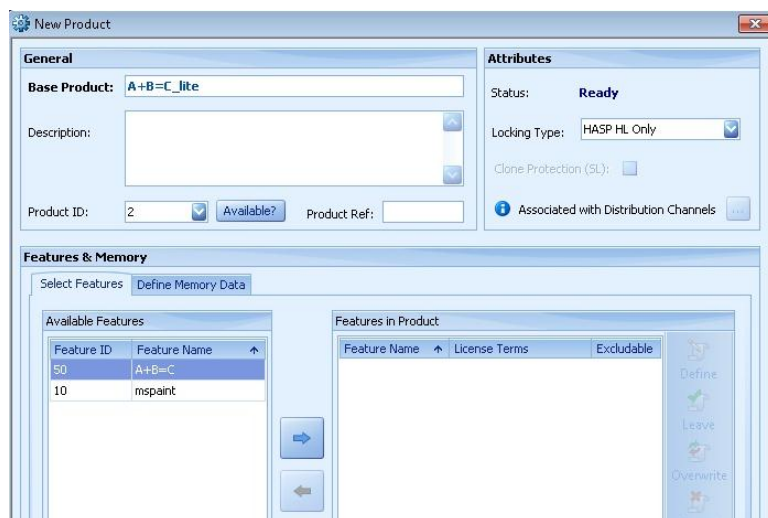


Рисунок 18 – New Product



Рисунок 19 – Features & Memory

8. в списке Features In Product выбрать A+B=C и нажать кнопку Define License Terms (Определите условия лицензии). В открывшемся окне необходимо установить лицензионные ограничения для данного компонента;
9. в списке License Type (Тип лицензии) выбрать Executions (По количеству запусков);
10. в поле Number of Executions (Количество запусков) установить значение 3;
11. закрыть окно Define License Terms (Определите условия лицензии), нажав кнопку ОК;
12. нажать кнопку ОК. В списке доступных продуктов в главном окне появится A+B=C\_Lite. Его статус должен быть Ready (Готов).

### Создание полнофункциональных лицензионных продуктов

В этой части этапа создаем копию имеющихся продуктов и создание на их основе еще два новых:

- первый будет включать в себя компонент A+B=C (ID 50) с неограниченной по сроку лицензией. Этот продукт будет использован для обновления версии A+B=C\_Lite с ограничением по количеству запусков;
- второй продукт будет включать в себя компонент mspaint (ID 10) с неограниченной по сроку лицензией. Далее рассмотрено, как создать

тестовую версию mspaint. По итогам тестирования ее можно будет обновить до полной версии, которая будет называться mspaint\_Full.

Для создания полной версии продукта  $A+B=C$  необходимо:

1. в Business Studio открыть в панели функций раздел Licensing Plan (Лицензионный план) и выбрать в нем Manage Products (Управление продуктами). В основной части окна будет представлена информация о доступных продуктах;
2. проверить, чтобы в Batch Code (Код серии) был указан код серии CRAXZ;
3. выбрать продукт  $A+B=C\_Lite$  и на панели задач нажать кнопку Duplicate (Дублировать);
4. в открывшемся окне ввести в поле Product Name (Название продукта)  $A+B=C\_Full$ ;
5. нажать Define License Terms - откроется окно, где будет предложено определить условия лицензии;
6. в поле License Type (Тип лицензии) выбрать Perpetual (Без ограничений);
7. закрыть окно Define License Terms (Определите условия лицензии), нажав кнопку ОК;
8. нажать кнопку ОК. В главном окне будет два продукта:  $A+B=C\_Lite$  и  $A+B=C\_Full$ . У обоих должен быть статус Ready (Готов);

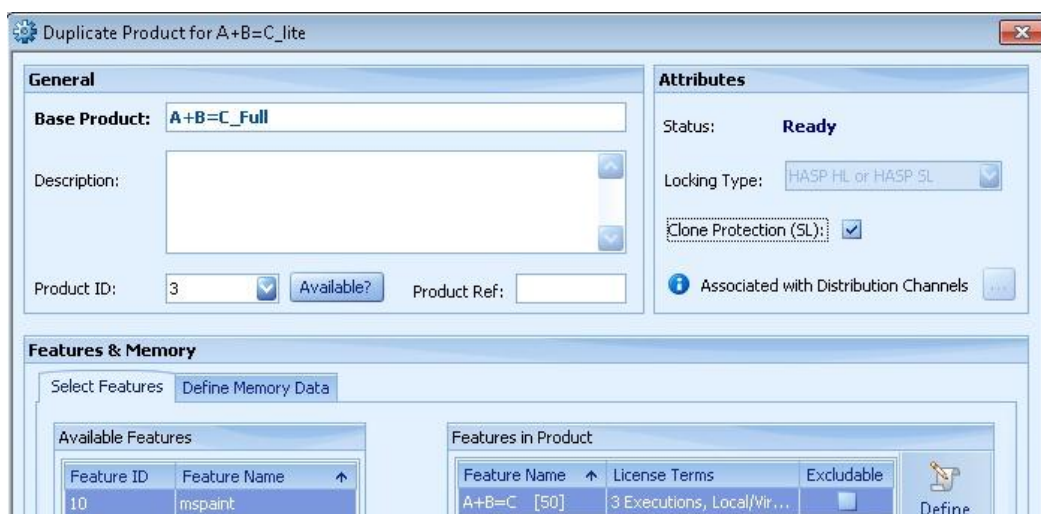


Рисунок 20 – Duplicate Product for  $A+B=C\_lite$





Рисунок 21 – Manage Products (2 продукта)

Создание полнофункционального продукта mspaint:

1. в Business Studio открыть в панели функций раздел Licensing Plan (Лицензионный план) и выбрать пункт Manage Products (Управление продуктами). В основной части окна будет информация о доступных продуктах;
2. проверить, чтобы в Batch Code (Код серии) был указан код серии CRAXZ;
3. выбрать продукт A+B=C\_Full и в панели задач нажать Duplicate (Дублировать);
4. в основной области окна в поле Product Name (Название продукта) ввести mspaint\_Full;
5. для внесения изменений в настройки компонента, в области Features and Memory (Компоненты и память) нужно открыть вкладку Select Features (Выбор компонентов) и выполнить следующие действия:

А. в перечне Features In Product (Составляющие компоненты) выбрать A+B=C и нажать на стрелку “влево”, тогда компонент будет удален из списка компонентов, включенных в состав продукта;

Б. в списке Available Features (Доступные компоненты) в поле ID выбрать 10 (mspaint) и нажать на стрелку “вправо”. Компонент будет снова добавлен в состав продукта;

6. для выделения области памяти для нашего компонента, в окне Features and Memory (Компоненты и память) надо открыть вкладку Define Memory Data (Настройки памяти) и выполнить следующие действия:



- А. нажать New Memory Segment (Новый сегмент памяти);
- Б. в открывшемся окне в поле Offset (Отступ) ввести 0;
- В. в поле Length (Длина) ввести 50;

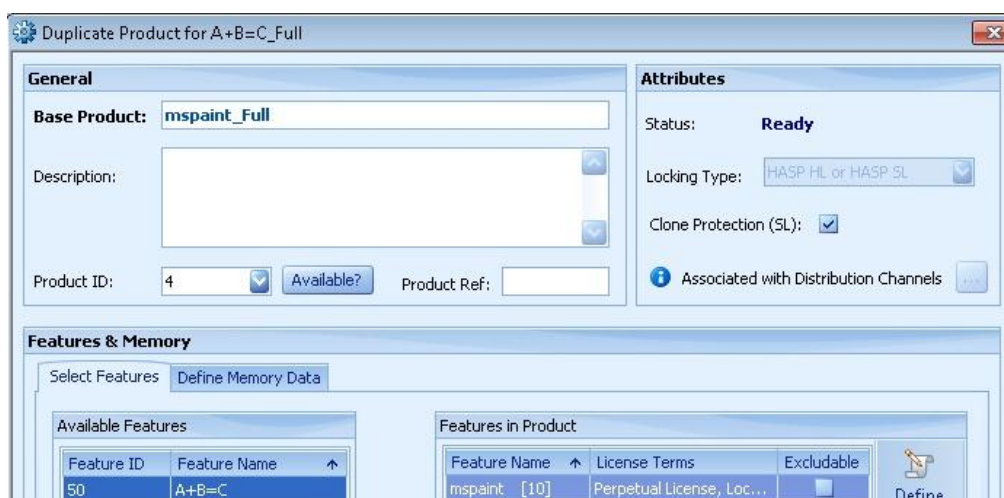


Рисунок 22 – Duplicate Product for A+B=C\_Full

Г. закрыть окно, нажав ОК. На вкладке Define Memory Data будет выделена нижняя часть;

Д. Установить курсор на первую позицию в правой части вкладки и ввести Hello World;

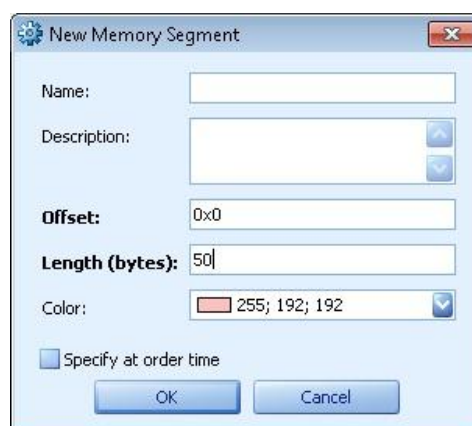


Рисунок 23 – New Memory Segment

7. закрыть окно, нажав кнопку ОК. Теперь список продуктов содержит три пункта: A+B=C\_Lite, A+B=C\_Full и mspaint\_Full.

Этап 5. Создание условно-бесплатной версии

Задача	Создание условно бесплатной версии продукта для тестового пользования
Роль	Отдел продуктов, отдел разработки
Используемые приложения	HASP SRM Business Studio

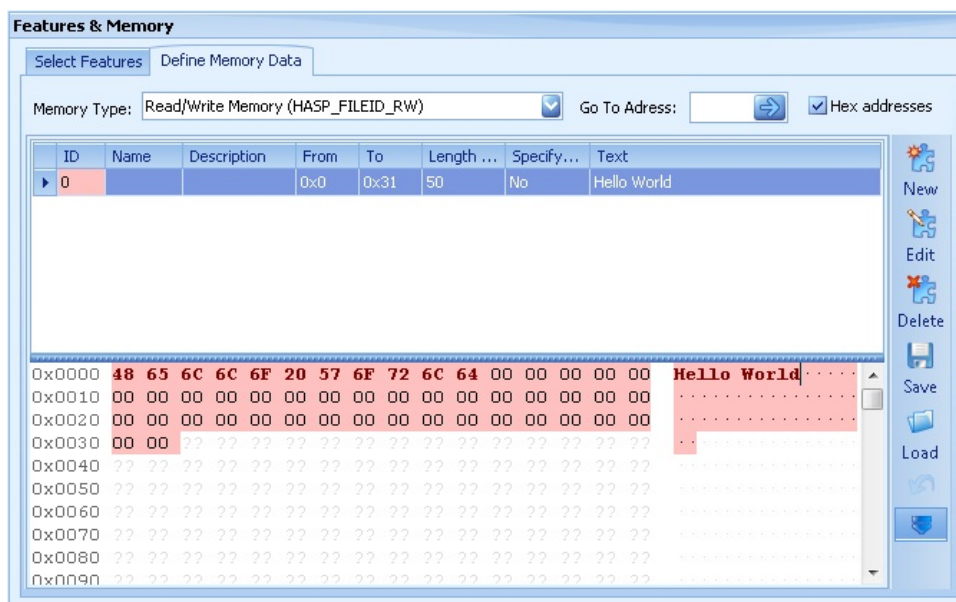


Рисунок 24 – Define Memory Data (Hello World)



Рисунок 25 – Manage Products (3 продукта)

На этом этапе исполнитель снова выступает в роли менеджера по продуктам, и его задачей является создание тестовой версии продукта mspaint. Затем для формирования программного пакета необходимо взять на себя роль разработчика. Предварительно необходимо отредактировать файл установки, который помимо собственно приложения будет устанавливать рабочую среду HASP SRM. Как правило, условно-бесплатное ПО распространяется в целях тестирования или супердистрибуции. Такой программный продукт может использоваться на любом компьютере без ключа HASP SRM, но только в течение ограниченного срока. По истечению установленного срока для запуска приложения требуется ключ HASP SRM.

## Создание условно-бесплатного продукта

Этот этап начинается с создания продукта `mspaint_Evaluation` – условно-бесплатной версии с лицензией, действующей в течение 30 дней с момента первого запуска.

Для создания условно-бесплатной версии приложения необходимо:

1. в панели функций открыть раздел Licensing Plan (Лицензионный план) и выбрать пункт Manage Products (Управление продуктами);
2. на панели задач нажать кнопку New Provisional (Создать продукт). Появится окно со свойствами продукта;
3. в основной области окна в поле Product Name (Название продукта) ввести `mspaint_Evaluation`;
4. Необходимо указать, что компонент `mspaint` (его создали на этапе 2) входит в состав продукта. Для этого в области Features & Memory (Компоненты и память) нужно выбрать компонент с идентификатором 10 и нажать на стрелку “вправо”;

Выбранный компонент появится в списке Features In Product. В поле License Terms (Условия лицензии) должно быть указано 30 days (срок действия лицензии – 30 дней).

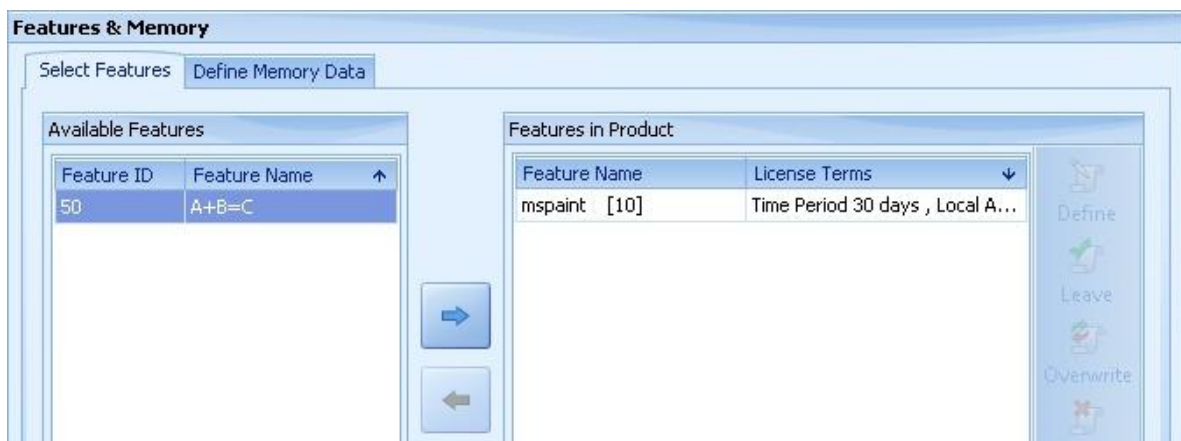


Рисунок 26 – Features & Memory

Примечание: не рассчитано на использование ключей HASP HL Basic, так как этот тип ключа не содержит раздела памяти для записи в нее информации (ключ без памяти).

5. закрыть окно с настройками, нажав кнопку ОК. В списке готовых продуктов появится mspaint\_Evaluation, в поле его статуса должно быть указано «Ready»;



The screenshot shows a window titled 'Manage Products'. It has a 'Batch Code' field with the value 'CRAXZ' and a 'Filter by' dropdown set to 'All Products'. Below these is a table with 7 columns: Type, Name, Product ID, Status, Description, and Locking Type. There are 4 rows of product data.

Type	Name	Product ...	Status	Description	Locking Type
	A+B=C_Full		Ready		HASP HL or HASP SL
	A+B=C_lite		Ready		HASP HL or HASP SL
	mspaint_Full		Ready		HASP HL or HASP SL
	mspaint_Full [...]		Ready	Provisional Product fo...	HASP SL Only

Рисунок 27 – Manage Products (4 продукта)

### Создание программного пакета

В этой части этапа пользователь выступает в роли разработчика. Необходимо подготовить к распространению условно-бесплатное приложение mspaint и создать программу установки, которая будет помимо собственно приложения устанавливать рабочую среду HASP SRM.

Для создания программного пакета из условно-бесплатного необходимо:

1. в панели функций открыть раздел Production и выбрать пункт Development Tasks;
2. В панели задач выбрать Bundle Provisional Products;



Рисунок 28 – Development Tasks

3. проверить, чтобы в поле Batch Code указан код серии CRAXZ;
4. нажать кнопку Add (Добавить). Появится окно Products (Продукты);
5. выбрать mspaint\_Evaluation и нажать кнопку Add (Добавить); mspaint\_Evaluation будет добавлен в список продуктов.
6. нажать кнопку Produce now. Появится окно Produce Bundle of Provisional Products (Создание пакета условно-бесплатных продуктов);
7. в открывшемся окне в поле File Type (Тип файлов) выбрать V2C;
8. напротив поля File Location (Путь к файлу) нажать кнопку Browse (Обзор). По умолчанию файл V2C находится в папке ...\\Documents and Settings\\имя текущего пользователя\\Application Data\\Aladdin\\HASP SRM [версия]\\VendorTools\\VendorSuite\\v2c;
9. нажать ОК. В поле File Location появится путь к выбранному файлу;

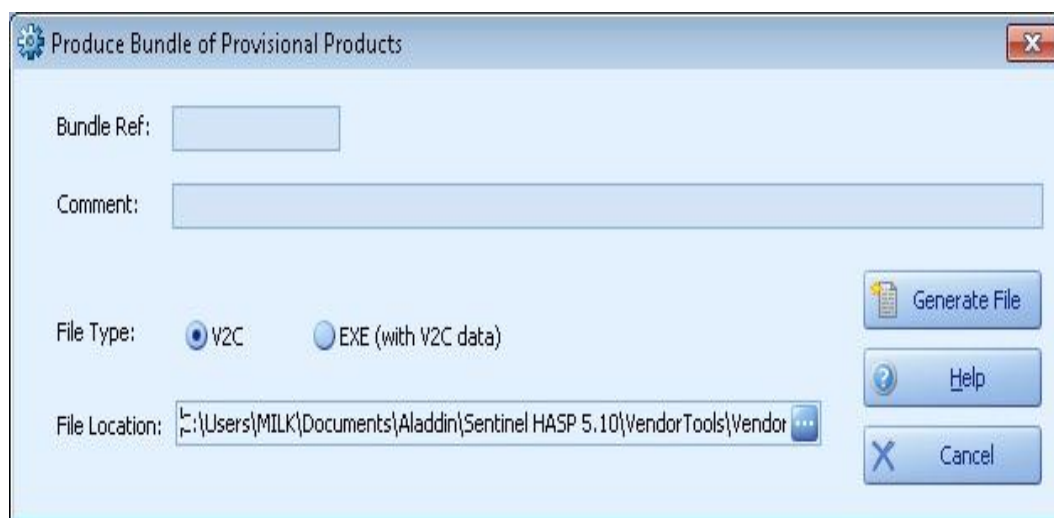
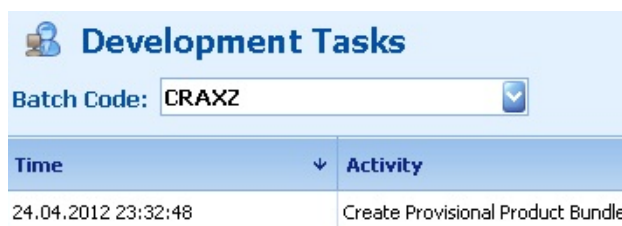


Рисунок 29 – Produce Bundle of Provisional Products

10. нажать кнопку Generate File (Сгенерировать файл). После того как будет сгенерирован файл с лицензионными данными появится соответствующее сообщение;

11. нажать кнопку ОК. В списке задач появится информация о том, что заказ сформирован.



Development Tasks	
Batch Code: CRAXZ	
Time	Activity
24.04.2012 23:32:48	Create Provisional Product Bundle

Рисунок 30 – Сформированный заказ

### Создание файла установки рабочей среды HASP SRM

В завершение этого этапа необходимо создать файл установки, который помимо самого приложения будет также устанавливать рабочую среду HASP SRM. Файл, запускающий программу установки, будет обращаться к созданному ранее файлу V2C. Рабочая среда должна быть обязательно установлена на компьютере конечного пользователя. Только в этом случае приложение, защищенное и лицензированное с помощью HASP SRM, будет работать корректно.

Примечание: в рассматриваемом примере условно-бесплатный программный пакет состоит из одного-единственного приложения – mspaint\_Evaluation.

Сначала необходимо создать программу установки рабочей среды HASP SRM, а затем добавить эту программу в процесс установки защищенного приложения. Ранее было указано в настройках mspaint, что лицензия на этот продукт действует 30 дней.

Для создания файла установки рабочей среды HASP SRM необходимо выполнить следующие действия:



1. в панели функций открыть раздел Production и выбрать пункт Development Tasks;
2. на панели задач нажать RTE Installer (Создать файл установки рабочей среды);
3. в поле V2C File Location (Путь к файлу v2c) указать путь к файлу или выбрать его, нажав кнопку Browse (Обзор); ...\\Documents and Settings\\текущий пользователь\\Application Data\\Aladdin\\HASP SRM [версия]\\Vendor Tools\\Vendor Suite\\v2c
4. двойным щелчком выбрать файл с расширением v2c. Путь к файлу появится в поле File Location (Путь к файлу);
5. в поле File Type (Тип файла) должно быть указано Application (EXE);
6. в поле Output Dir указать путь к сгенерированному файлу программы установки, нажав кнопку Browse (Обзор) и создав на рабочем столе папку EvaluationProducts. Нажать ОК. Путь к выбранной папке появится в поле Output Dir;

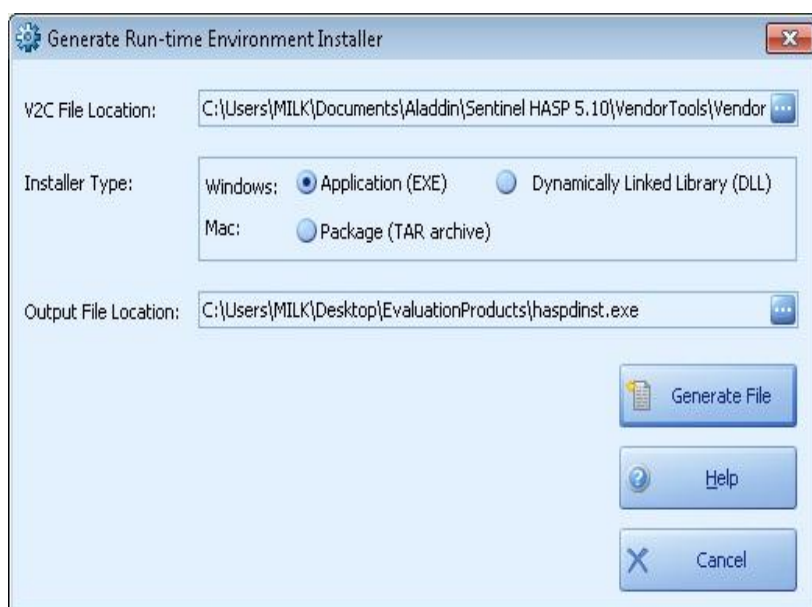


Рисунок 31 – Generate Run-time Environment Installer

7. нажать Generate File (Сгенерировать файл). После того, как файл установки рабочей среды HASP SRM будет сгенерирован, появится соответствующее

сообщение. Нажать кнопку ОК. Внимание: не надо запускать сгенерированный файл, процесс установки описан далее на этапе 7.

#### Этап 6. Оформление и обработка заказов

Задача	Рассмотрен процесс создания учетной записи клиентов, процесс оформления и обработки их заказа
Роль	Отдел продаж, производство
Используемые приложения	<ul style="list-style-type: none"><li>• HASP SRM Business Studio</li><li>• Система удаленного обновления лицензий HASP SRM RUS</li></ul>

На этом этапе пользователь выступает в роли сотрудника отдела продаж, оформляющего три заказа. Следует создать заказ, создать для него учетную запись заказчика и затем подготовить заказ к отгрузке. Для составления заказа использованы те продукты, которые были созданы на этапе 6. Таким образом, будут сформированы три заказа:

- заказ на продукт A+B=C\_Lite. В параметрах лицензии будет настроено использование защищенного приложения только с ключом HASP HL;
- заказ на полнофункциональную версию продукта – A+B=C\_Full. Этот заказ будет использоваться для обновления лицензии на ключе HASP HL, который поставляется в комплекте с продуктом A+B=C\_Lite;
- заказ на mspaint\_Full. По условиям лицензии этот продукт будет использоваться с ключом HASP SL, который будет передан пользователю для обновления тестовой версии mspaint до полнофункциональной.

Примечание: на этапе 7 исполняете роль пользователя. Рассмотрено, как активировать или обновить версию продукта в зависимости от условий лицензии. Во всех перечисленных заказах условия лицензии заданы ранее, однако их можно установить и на этапе оформления заказа.

#### Оформление заказа на аренду ПО с привязкой к ключу HASP HL



Оформление заказа на продукт A+B=C\_Lite и настройка его на использование только с ключом HASP HL.

Для оформления заказа на приложение, которое будет использоваться только с ключом HASP HL необходимо:

1. на панели функций в Business Studio открыть раздел Production и выбрать пункт Manage Orders. В основной части окна представлена информация о заказах;
2. на панели задач нажать кнопку New (Оформление нового заказа);
3. в разделе Customer (Заказчик) щелкнуть на New Customer (Новый заказчик);
4. в открывшемся окне в поле Name (Имя) указать заказчика (Иван Петров);
5. нажать ОК. В поле Customer Name (Имя заказчика) будет указано Иван Петров;
6. нажать кнопку Add (Добавить). Появится окно Products (Продукты);
7. добавить в заказ продукт A+B=C\_Lite, выбрав его и нажав кнопку Add (Добавить).



Рисунок 32 – Products

Чтобы установить для продукта привязку к ключу HASP HL, необходимо:

1. в Order Details (Сведения о заказе) выбрать параметр HASP HL Keys;
2. в поле Number of keys (Количество ключей) ввести 1;
3. нажать кнопку Produce Now (В обработку);
4. подсоединить ключ серии HASP HL, который будет поставлен покупателю;

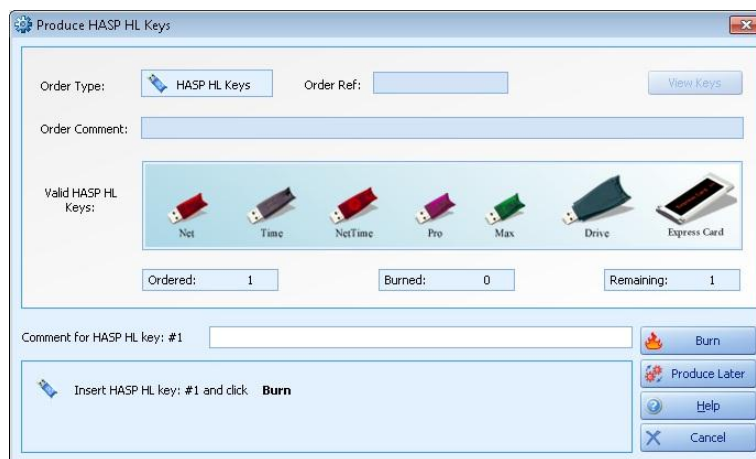


Рисунок 33 – Produce HASP HL Keys

5. нажать кнопку Burn Now. Появится индикатор исполнения процесса;

После окончания процесса записи лицензии в ключ HASP HL, в поле Burned в окне Produce будет указано соответствующее значение.

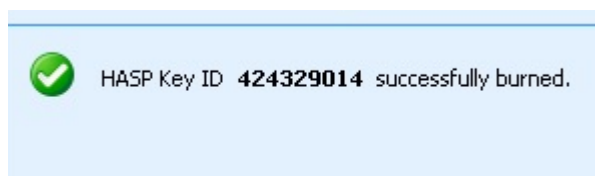


Рисунок 34 – Значение лицензии

6. для завершения процесса надо закрыть окно, нажав кнопку Done (Готово).

Примечание: процедура может осуществляться со стороны пользователя, имеющего роль Production (Производство). Однако его права в этой роли должны быть ограничены выполнением в HASP SRM Business Studio исключительно данной процедуры.

### Создание обновления для лицензии в удаленном режиме

Необходимо создать обновление для лицензионного продукта, используемого только с ключом HASP HL. К процессу обновления продукта вернемся на этапе 7, когда будем действовать от лица конечного пользователя. Дистанционное обновление лицензии позволяет пользователю получить новые

возможности и компоненты и/или продлить срок лицензии на уже установленный продукт. Система удаленного обновления лицензий позволяет значительно оптимизировать процесс, так как пользователю не нужно передавать свой ключ (будь то HASP HL или HASL SL) разработчику. Перед началом работы с системой HASP SRM RUS необходимо ввести код серии.

Чтобы ввести код серии в системе HASP SRM RUS необходимо:

1. на панели функций в Business Studio выбрать раздел Production и в нем пункт Development Tasks;
2. проверить, чтобы в поле Batch Code был указан код CRAXZ;
3. на панели задач щелкнуть на RUS Branding. Откроется новое окно редактора HASP SRM HTML;

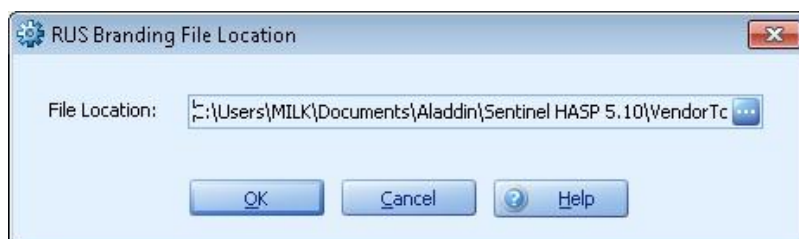


Рисунок 35 – RUS Branding

4. не изменяя никаких параметров, нажать кнопку OK. Появится сообщение о том, что файл обновления лицензии создан. Нажать OK.

#### Обработка заказа на обновление лицензии

На практике, чтобы обновить лицензию на ключе, пользователь должен сообщить продавцу идентификатор ключа HASP. Чтобы воспроизвести этот сценарий, необходимо выполнить следующие действия:

1. запустить приложение Admin Control Center. Для этого нужно открыть меню Пуск (Start), выбрать Programs Программы (Programs) → Aladdin → HASP SRM → Admin Control Center;
2. в панели Administration Options (Опции администрирования) выбрать HASP Keys;

3. проверить номер ключа HASP HL в поле HASP Key ID;

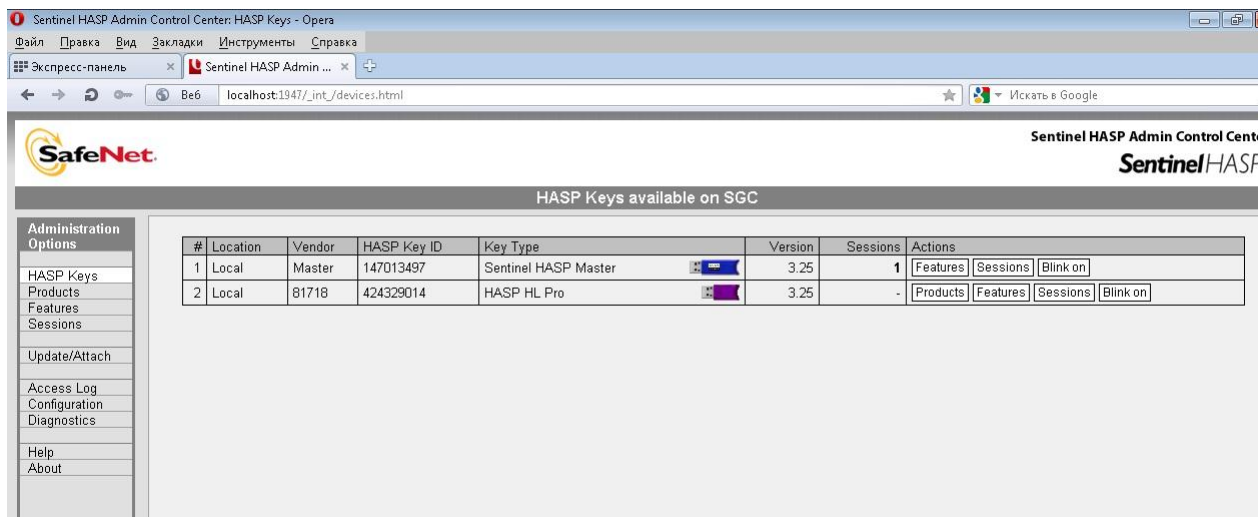


Рисунок 36 – Admin Control Center

4. в Business Studio открыть на панели функций раздел Production и выбрать в нем пункт Manage Orders;

5. на панели задач нажать кнопку New Production Order (Оформление нового заказа);

6. в открывшемся окне выбрать New Customer (Новый заказчик);

7. в поле Name (Имя заказчика) ввести имя заказчика (например, Иван Петров);

8. нажать кнопку ОК. В окне Production Order в поле Customer Name будет указано «Иван Петров»;

9. нажать кнопку Add (Добавить). Появится окно Products (Продукты);

10. добавить в заказ продукт A+B=C\_Full, выбрав его и нажав кнопку Add (Добавить);

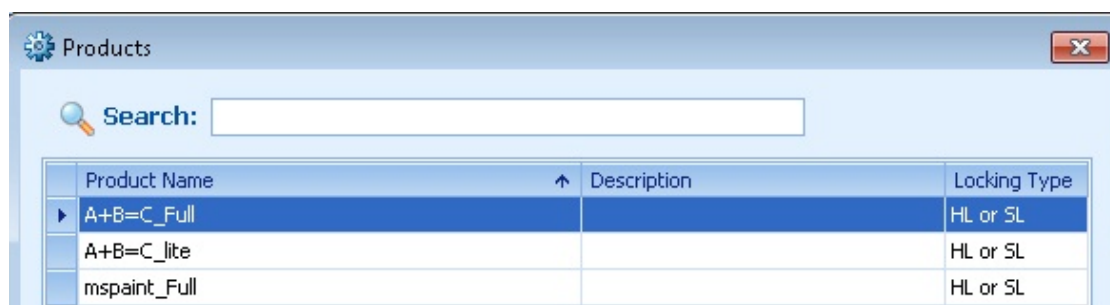


Рисунок 37 – Products

11. в области Order Details выбрать пункт HASP Update;
12. чтобы выбрать ключ, для которого требуется обновление лицензии, необходимо нажать Locate;
13. в открывшемся окне выбрать HASP ID;
14. в поле HASP ID ввести номер ключа, который был получен от пользователя (см. пункт 3). Нажать кнопку Locate. Ключ с указанным HASP ID должен появиться в поле Locate HASP keys by. Нажать кнопку Add to List
15. закрыть окно, нажав кнопку OK;
16. нажать кнопку Produce Now (В обработку);
17. в открывшемся окне в поле File Type выбрать V2C;
18. в поле V2C File Location (Путь к файлу V2C) нажать кнопку Browse (Обзор);
19. нажать кнопку Generate File (Сгенерировать файл). После того как файл будет сгенерирован, и появится соответствующее сообщение;
20. закрыть окно, нажав кнопку OK.

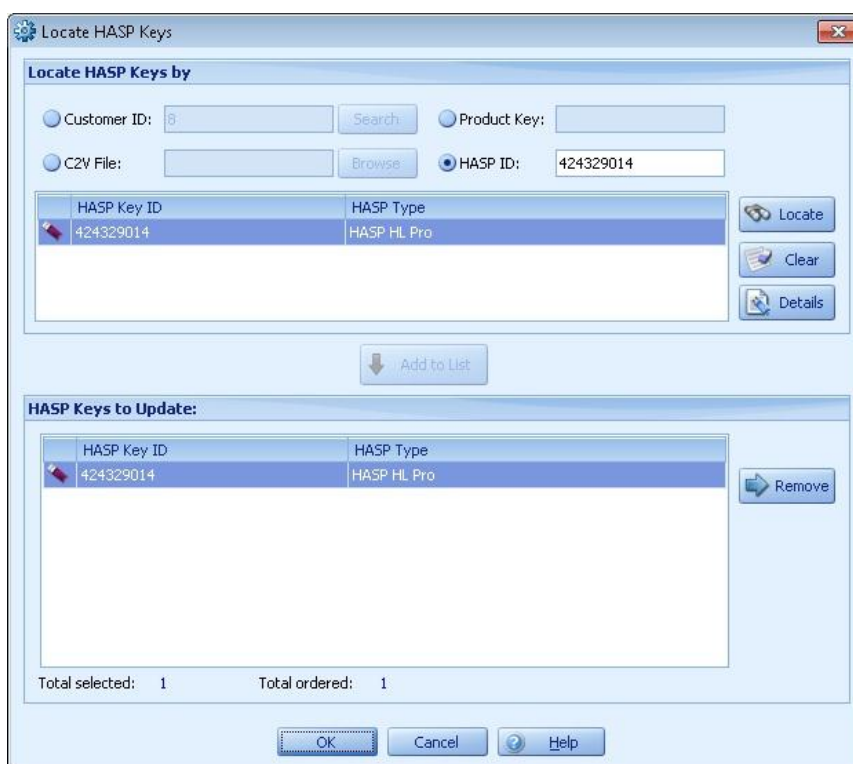


Рисунок 38 – Locate HASP Keys

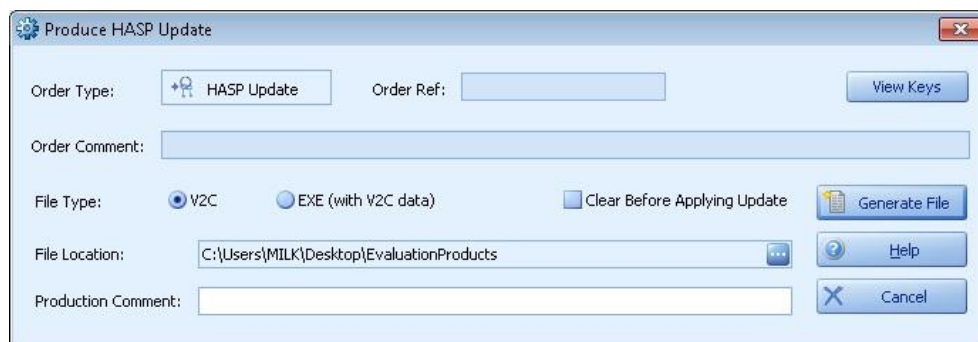


Рисунок 39 – Produce HASP Update



Рисунок 40 – Manage Orders

### Создание неограниченной лицензии с привязкой к программному ключу HASP SL

Необходимо создать заказ на продукт mspaint\_Full, который будет иметь привязку к ключу HASP SL. Затем у пользователя появляется возможность обновить тестовую версию mspaint до полнофункциональной версии. Чтобы создать файл с неограниченной лицензией, необходимо:

1. в Business Studio открыть на панели функций раздел Production и выбрать пункт Manage Orders (Управление заказами);
2. на панели задач нажать кнопку New Production Order (Оформление нового заказа);
3. в открывшемся окне выбрать New Customer (Новый заказчик);
4. в поле Name (Имя заказчика) ввести имя заказчика (например, Иван Петров);
5. нажать кнопку ОК. В окне Production Order в поле Customer Name будет указано «Иван Петров»;

6. нажать кнопку Add (Добавить). Появится окно Products (Продукты);
7. добавить в заказ продукт mspaint\_Full, выбрав его и нажав кнопку Add (Добавить);



Рисунок 41 – Products

8. в области Order Details (Сведения о заказе) выбрать пункт Product Keybased;
9. нажать кнопку Produce Now;
10. в открывшемся окне в поле File Location (Путь к файлу) нажать кнопку Browse (Обзор) и создать папку с названием ProductKeys, где будет храниться Ключ продукта; В поле File Name указано название продукта по умолчанию.

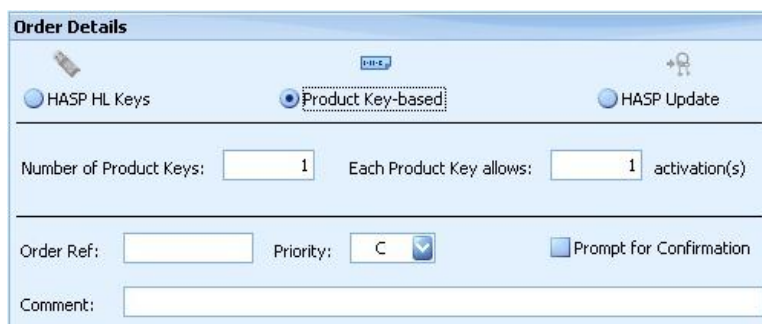


Рисунок 42 – Order Details



Рисунок 43 – Produce Product Keys

11. нажать Generate File. После того как файл будет сгенерирован, появится соответствующее сообщение;

12. в конце закрыть окно, нажав кнопку ОК. Новый заказ появится в списке Manage Orders;

#### Этап 7. Работа с защищенным приложением на стороне пользователя

Задача	Рассмотреть несколько типовых сценариев работы с приложениями, защищенными HASP SRM
Роль	Конечный пользователь
Используемые приложения	<ul style="list-style-type: none"><li>• Admin Control Center</li><li>• Система удаленного обновления лицензий HASP SRM RUS</li></ul>

Рассмотрим примеры работы на стороне пользователя:

- запуск приложения с ключом HASP HL;
- обновление лицензии на ключе HASP HL;
- работа в HASP SRM Admin Control Center;
- установка тестового продукта;
- переход от тестовой к полнофункциональной версии.

Пользователь, получив защищенное приложение, должен проверить, что ключ HASP SRM доступен для использования с этим приложением. Рассмотрим процедуру тестирования различных типов ключей HASP HL.

#### Запуск приложения с ключом HASP HL

В этой части этапа исполняем роль клиента, то есть Ивана Петрова, покупателя ПО. Рассмотрим работу с HASP SRM Admin Control Center – приложением, позволяющим системному администратору через веб-интерфейс управлять лицензионным программным обеспечением в организации с разделением по каждому пользователю.



Чтобы активировать продукт, защищенный ключом HASP HL, необходимо выполнить следующие действия:

1. подсоединить ключ HASP HL к компьютеру. Лицензия хранится на ключе в зашифрованном виде. Приложение можно будет запустить только тогда, когда лицензия будет действительной;
2. в меню Пуск (Start) открыть пункт Программы (Programs) → Aladdin → HASP SRM → Vendor Suite → Admin Control Center. Откроется окно браузера с главной страницей HASP SRM Admin Control Center. Доступны для просмотра данные об используемых в данной компании лицензиях;
3. на панели Administration Options выбрать HASP Keys. В HASP License Manager отображаются все доступные ключи HASP SRM;
4. найти необходимый ключ HASP HL и нажать справа кнопку Features (Компоненты). Будут представлены те компоненты, которые указаны в лицензии ключа;

Внимание: в поле Restrictions для компонента с номером 50 указано Activations 3 left, то есть по условиям лицензии для данного компонента осталось три попытки авторизации.

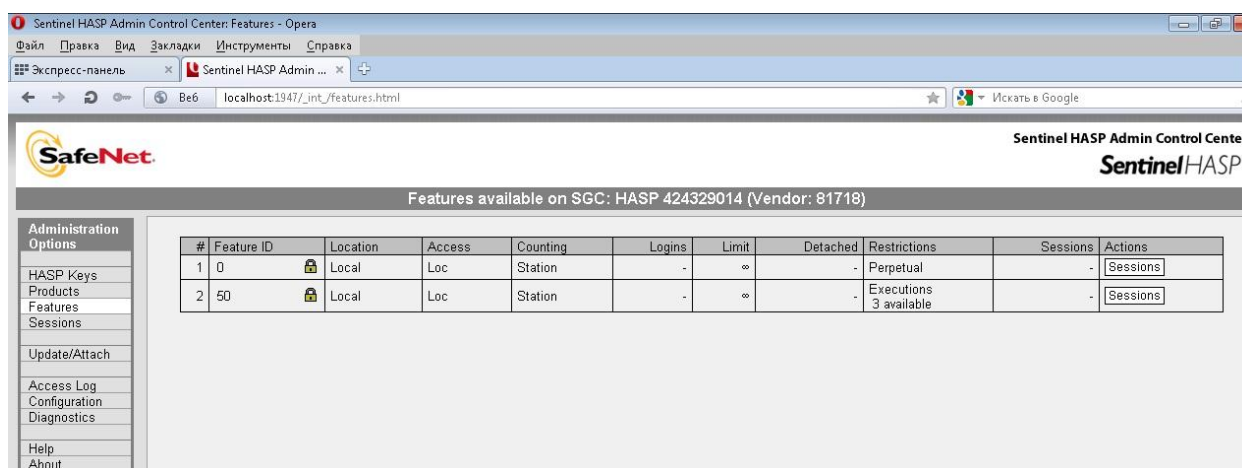


Рисунок 44 – Admin Control Center

5. открыть папку ... \Documents and Settings\[имя текущего пользователя] \My Documents \ Aladdin \HASP SRM [версия] \VendorTools \VendorSuite \Protected и

двойным щелчком запустить файл A+B=C.exe. При запуске можно увидеть сообщение о том, что приложение защищено с помощью HASP SRM;

6. нажать кнопку ОК;

7. закрыть приложение A+B=C;

8. выполнить действия, описанные в пунктах с 5 по 7, еще два раза;

9. При очередной попытке запуска приложения появится сообщение о том, что срок лицензии истек, и приложение не может быть запущено;

10. открыть HASP SRM Admin Control Center и на панели Administration Option (Административные параметры) выбрать пункт HASP Keys. В HASP License Manager отображаются все доступные ключи HASP SRM;

11. выбрать демо-ключ HASP, нажать справа кнопку Features (Компоненты);

12. в поле Restrictions (Ограничения) для компонента с номером 50 будет указано Expired, то есть достигнуто максимальное количество запусков, указанное в лицензии.

### Активация лицензии в удаленном режиме

Выполним операции от имени Ивана Петрова, который купил полнофункциональную версию продукта и собирается установить ее вместо тестовой версии, ограниченной по количеству запусков. Обновление лицензии происходит через систему HASP SRM RUS, и для этого понадобится использовать полученный от разработчика файл HaspUpdate.v2c.

Для активации лицензии необходимо выполнить следующее:

1. подсоединить ключ HASP HL к компьютеру;

2. открыть папку ...\\Documents and Settings\\[имя текущего пользователя]\\My Documents\\Aladdin\\HASP SRM [версия]\\Vendor Tools\\VendorSuite\\Protected) и запустить файл hasprus.exe;

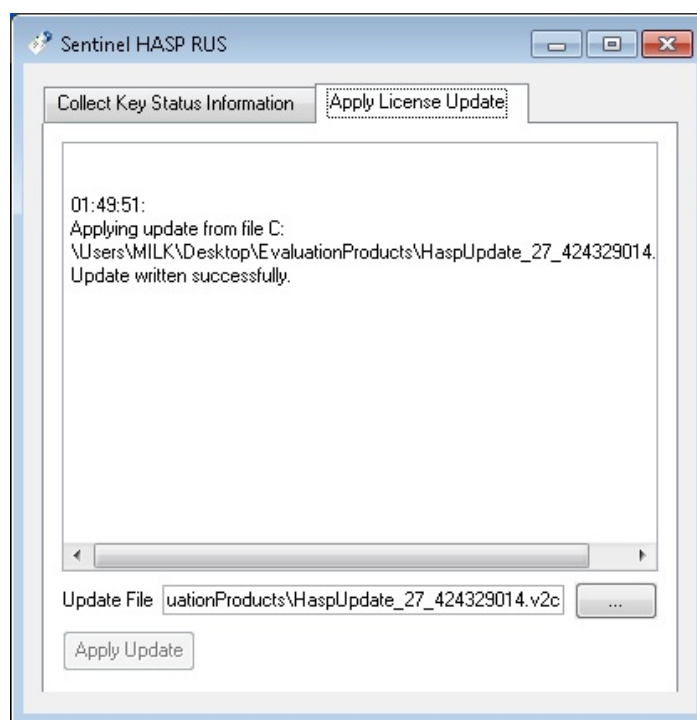


Рисунок 45 – Sentinel HASP RUS

3. в окне программы открыть вкладку Apply License Tab, нажать кнопку Browse (Обзор) и открыть папку ...\\Documents and Settings\\[имя текущего пользователя];

Примечание: если нет возможности точно определить необходимый ключ, следует нажать кнопку Blink on, и на одном из ключей индикатор начнет мигать.

В папке Application Data\\Aladdin\\HASP SRM [версия]\\VendorTools\\VendorSuite\\v2с. хранится файл HaspUpdate.v2с, который был создан на этапе 6.

4. нажать кнопку Open (Открыть), указать путь к файлу в окне HASP SRM RUS;

5. для обновления лицензии нажать кнопку Apply update (Обновить лицензию);

6. запустить HASP SRM Admin Control Center, открыв меню Пуск (Start) и выбрав в нем Программы (Programs)→Aladdin → HASP SRM → Vendor Suite → Admin Control Center;

7. выбрать из списка необходимый ключ HASP HL и нажать справа кнопку Features (Компоненты). В поле лицензии для компонента с номером 50 должно быть указано Perpetual (Без ограничений);

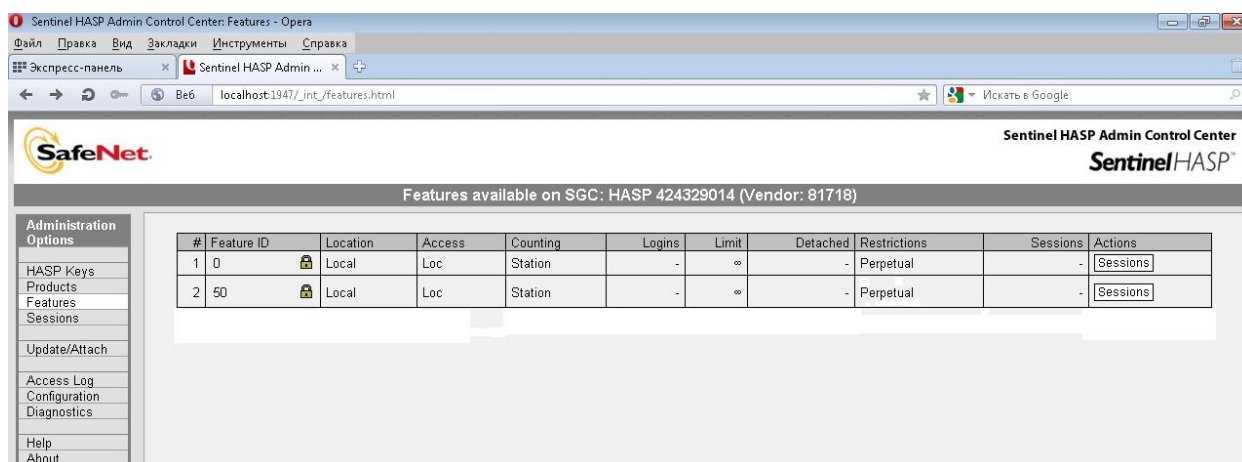


Рисунок 46 – Admin Control Center

8. открыть папку ...\\Documents and Settings\\[имя текущего пользователя]\\MyDocuments\\ Aladdin\\HASP SRM [версия]\\Vendor Tools\\VendorSuite\\Protected, где хранится файл A+B=C.exe. При запуске появится сообщение, что приложение защищено с помощью HASP SRM;
9. чтобы продолжить загрузку приложения, нажать ОК;
10. закрыть приложение и отсоединить ключ HASP HL.

### Установка тестовой версии

Рассмотрим пример установки тестовой версии mspaint на стороне пользователя. По практическим соображениям рекомендуется включать в дистрибутив установку рабочей среды. Однако рассмотрим эти две процедуры в отдельности.

Чтобы установить рабочую среду HASP SRM, необходимо:

1. открыть меню Пуск (Start) и выбрать пункт Выполнить (Run);
2. нажать кнопку Browse (Обзор) и выбрать ранее созданную папку EvaluationProducts. Выбрать файл haspdinst.exe и нажать кнопку Open (Открыть);

В окне появится путь к папке EvaluationProducts.

3. в поле Open после пути надо указать через пробел параметр **-i** ;

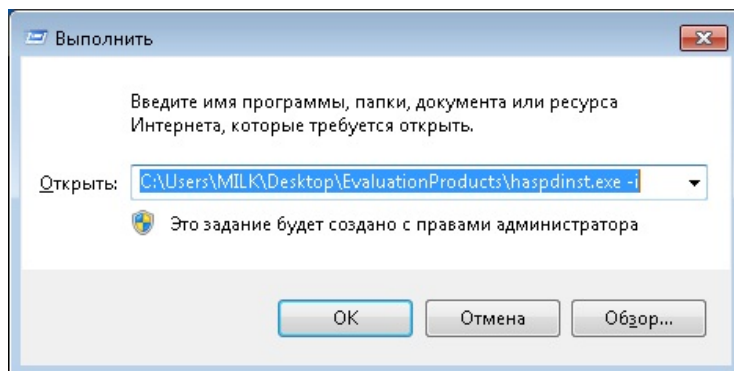


Рисунок 47 – окно Выполнить

4. в конце нажать ОК. После установки рабочей среды HASP SRM RTE появится соответствующее сообщение.

Чтобы просмотреть данные о лицензии для тестовой версии продукта, необходимо выполнить следующие действия:

1. запустить файл mspaint.exe (файл находится в папке ...\\Documents and Settings\\[имя текущего пользователя]\\My Documents\\Aladdin\\HASP SRM [версия]\\VendorTools\\VendorSuite \\Protected). На экране появится сообщение о том, что приложение защищено с использованием демонстрационной версии ключа HASP SRM.

2. нажать ОК;

3. не закрывая приложения, открыть меню Пуск (Start) и затем открыть Программы (Programs) → Aladdin → HASP SRM → Vendor Suite → Admin Control Center;

4. на панели Administration Options выбрать HASP Keys. В окне HASP License Manager будут представлены все ключи HASP SRM;

5. выбрать номер ключа HASP (идентификатор ключа – числовое значение, представляющее собой еще не активированный ключ HASP SL). Нажать справа кнопку Features (Компоненты). Появится информация о лицензионных компонентах. Следует обратить внимание на то, что поле Logins указывает

только на один оставшийся сеанс авторизации, в то время как в поле Restrictions (Ограничения) указан срок действия лицензии 30 дней.

### Обновление тестовой лицензии с использованием ключа продукта

В заключительной части этапа рассмотрим процесс приобретения и получение ключа продукта от разработчика. Ключ продукта затем используется для обновления тестовой версии приложения mspaint до полнофункционального приложения.

1. ранее было создано приложение, которое выполняет активацию продукта через API-интерфейс HASP SRM. Необходимо запустить файл HASP Activation Sample.exe;

2. путь к исполняемому файлу зависит от установленной ОС:

- ...\\Documents and Settings\\[имя текущего пользователя]\\My Documents\\Aladdin\\HASP SRM [версия]\\Samples\\Activation\\cpp\\msvc71\\bin, если работа проводится на любой платформе, кроме Windows Vista;
- если установлена ОС Windows Vista, путь к исполняемому файлу будет следующим: ...\\Documents\\[имя текущего пользователя]\\Documents\\Aladdin\\HASP SRM [версия]\\Samples\\Activation\\cpp\\msvc71\\bin;

3. ввести в поле Product Key ключ продукта, который хранится в папке ProductKey;

4. нажать кнопку Activate (Активировать). Появится сообщение о том, что пример активации продукта выполнен;

5. открыть Admin Control Center и проверить, чтобы в лицензии для компонента с номером 10 было указано значение 10.

### Контрольные вопросы

1. Построение защиты на основе использования Hasp Envelope. Достоинства и недостатки?

2. Как действия выполняются на подготовительном этапе и этапе создания компонентов?
3. Какие действия выполняются на этапе формирования программного пакета и этапе создания условно-бесплатной версии?
4. Какие действия необходимо выполнить на этапе оформления и обработки заказов?
5. Типовые действия по работе с защищенным приложением, выполняемые на стороне пользователя после получения им защищенной программы?

### **Требования к содержанию отчета**

Отчет выполняется на листах формата А4, которые должны быть скреплены перед сдачей отчета преподавателю. Первый лист отчета титульный – содержит информацию об учебном заведении, кафедре, названии дисциплины, теме выполненной работы. Приводится информация об учащемся, выполнившим работу. Приводится информация о преподавателе, проверяющем работу. В нижней части листа информация о месте (городе), где находится учебное заведение и год выполнения работы.

В содержательной части отчета о проделанной работе необходимо указать номер работы и тему работы. Затем описать ход выполнения работы. Желательно проиллюстрировать ход выполнения работы скриншотами (копиями экрана), полученными в процессе выполнения работы.

Если преподавателем были заданы вопросы по выполнению работы, на которые предложено ответить письменно, то приводятся вопросы и ответы на них.

Завершается отчет выводами по проделанной работе. Отмечаются достоинства и недостатки изученной технологии, область использования разработки и т.д.

Желательно привести в отчете список использованной литературы и интернет ресурсов (сайтов сети Интернет).

## **ОРГАНИЗАЦИЯ ВСТРОЕННОЙ ЗАЩИТЫ ПРОГРАММ НА DELPHI С ПОМОЩЬЮ SENTINEL HASP API**

### **Цель работы**

Получить практические навыки работы с Sentinel HASP API, установить встроенную защиту в программу, созданную средствами языка Delphi. Выполнить чтение и запись данных на аппаратный ключ.

### **Краткие теоретические сведения**

API-интерфейс системы HASP SRM представляет собой мощное средство для защиты программного обеспечения. Уровень защиты всецело зависит от объема используемых функций. Соответственно, чем более сложной будет структура защиты, тем надежнее она будет. Перед тем как приступить к практике, рекомендуется внимательно ознакомиться с функционалом HASP SRM Run-time API. Для защиты приложения с помощью HASP SRM API в исходный код защищаемого приложения следует вставить вызовы ключа HASP SRM. Можно вставить проверки наличия ключа в любое место приложения, а также определить его реакцию в зависимости от полученного ответа. Если требуемый ключ не обнаружен, приложение может прекратить работу или перейти в демонстрационный режим.

HASP SRM API позволяет контролировать наличие ключа и определенные данные в его памяти, а также осуществлять шифрование и дешифрование данных.

Все API-функции HASP SRM условно разделяются на пять категорий:

- сессионные функции;
- функции памяти;



- функции шифрования;
- функции времени;
- функции управления.

### Сессионные функции

Для выполнения большинства API-функций HASP SRM необходимо предварительно выполнить вызов login и получить, таким образом, доступ к лицензии, хранящейся на определенном аппаратном ключе. Каждый вызов login должен сопровождаться по окончании сеанса вызовом logout. Все остальные функции выполняются только в рамках установленного сеанса.

### Функции шифрования

После успешного соединения с ключом HASP SRM становятся доступными функции шифрования данных в буфере данных с помощью встроенного крипто-процессора, использующего симметричное шифрование, которое предусматривает использование одного ключа, как для шифрования, так и для дешифрования.

### Функции памяти

HASP SRM API может применяться для:

- чтение данных из буфера в памяти ключа HASP SRM;
- запись данных в буфер в памяти ключа HASP SRM.

Размер буфера данных ограничивается размером памяти используемой модели ключа. Рекомендуется использовать память аппаратного ключа для хранения информации, которая может использоваться для проверки и идентификации конечного пользователя, а также задействовать память для хранения данных, используемых

приложением во время его работы. Доступ к этой конфиденциальной информации является частью общей схемы защиты.

### Функции времени

Ключи HASP HL Time или HASP HL NetTime имеют встроенные таймеры, которые также управляются посредством API-функций HASP SRM. Например, можно считывать и изменять время. В числе API-функций HASP SRM есть две, которые позволяют преобразовывать значения даты и времени.

### Функции управления

В состав HASP SRM API входят две функции, возвращающие данные о системных компонентах, текущем сеансе, ключе HASP SRM и дополнениях к лицензии. В частном случае эти функции могут использоваться для установки обновлений. Для этого не обязательно иметь открытый сеанс.

Рассмотрим некоторые функции подробнее:

- function hasp\_login( feature\_id : hasp\_feature\_t;  
                          vendor\_code : hasp\_vendor\_code\_t;  
                          var handle : hasp\_handle\_t ) : hasp\_status\_t; stdcall;

Входные параметры:

feature\_id     - уникальный идентификатор feature,

vendor\_code    - указатель на секретный код ключа

handle          - указатель на сеанс

Возвращаемые значения:

HASP_STATUS_OK	запрос выполнен успешно
HASP_FEATURE_NOT_FOUND	запрошенная feature не доступна
HASP_CONTAINER_NOT_FOUND	с данным кодом ключей не найдено

HASP_OLD_DRIVER	устаревшая версия драйверов
HASP_NO_DRIVER	Драйвер не найден
HASP_INV_VCODE	ошибка vendor кода

- `function hasp_logout(handle : hasp_handle_t) : hasp_status_t; stdcall;`

Входные параметры:

handle - указатель на сеанс

Возвращаемые значения:

HASP_STATUS_OK	сеанс был успешно завершен
HASP_INV_HND	сеанс не был открыт

- `function hasp_write ( handle : hasp_handle_t;  
fileid : hasp_fileid_t; offset : hasp_size_t;  
length : hasp_size_t; var buffer) : hasp_status_t; stdcall;`

Входные параметры:

handle - указатель на сеанс

fileid - идентификатор файла для записи

offset - сдвиг в файле

length - длина записи

buffer - записываемые данные

Возвращаемые значения:

HASP_STATUS_OK	запись успешно завершена
HASP_INV_HND	сеанс не был открыт
HASP_INV_FILEID	ошибка в идентификаторе файла
HASP_MEM_RANGE	память вне диапазона
HASP_CONTAINER_NOT_FOUND	память вне диапазона

- `function hasp_read( handle : hasp_handle_t;  
fileid : hasp_fileid_t; offset : hasp_size_t;  
length : hasp_size_t; var buffer) : hasp_status_t; stdcall;`

Входные параметры:

handle - указатель на сеанс  
 fileid - идентификатор файла для записи  
 offset - сдвиг в файле  
 length - длина записи  
 buffer - считываемые данные

Возвращаемые значения:

HASP_STATUS_OK	запись успешно завершена
HASP_INV_HND	сеанс не был открыт
HASP_INV_FILEID	ошибка в идентификаторе файла
HASP_MEM_RANGE	память вне диапазона
HASP_CONTAINER_NOT_FOUND	память вне диапазона

Описание остальных функций HASP API можно найти в каталоге:

C:\Users\<Имя пользователя>\Documents\Aladdin\Sentinel HASP  
 5.10\Samples\Runtime\Delphi в файле hasp\_unit.pas.

1. До написания кода программы, необходимо выполнить следующие подготовительные действия:

- из папки C:\Users\<Имя пользователя>\Documents\Aladdin\Sentinel HASP 5.10\API\Runtime\Delphi скопировать файлы hasp\_unit.pas и hasp\_windows\_delphi\_<серия ключа>.obj, где серия ключа – пятизначное число, например 81718, в папку с будущей программой;
- в файле hasp\_unit.pas найти строку {\$L hasp\_windows\_delphi\_demo.obj} и заменить ее на {\$L hasp\_windows\_delphi\_<серия ключа>.obj};
- из папки C:\Users\<Имя пользователя>\Documents\Aladdin\Sentinel HASP 5.10\Samples\Runtime\Delphi скопировать файл hasp\_vcode.inc в папку с будущей программой. В этом файле содержится секретный код ключа DEMOMA (ключ демонстрационной серии). Необходимо открыть файл и заменить код на код используемого ключа (например, для ключей серии разработчика CRAZX, ключ содержится в файле CRAZX.hvc в каталоге

C:\Users\<Имя пользователя>\Documents\Aladdin\Sentinel HASP  
5.10\VendorCodes).

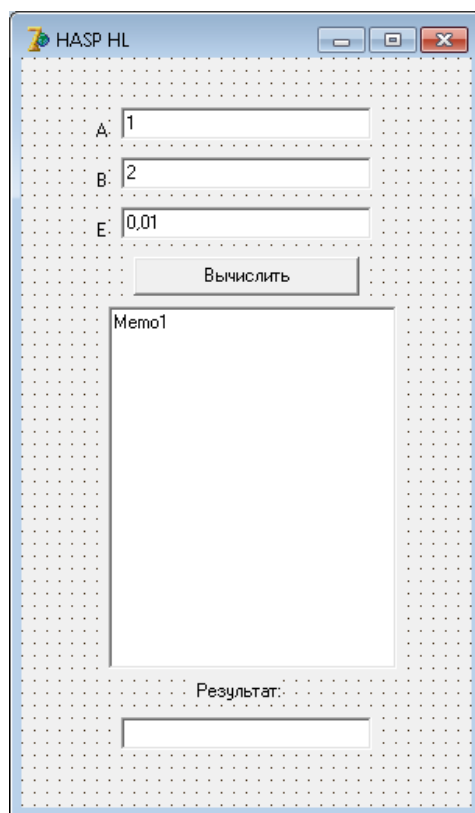


Рисунок 48 – макет пользовательского интерфейса программы

2. После подключения дополнительных файлов можно приступить к разработке программы. Программа выполняет следующие действия:

- проверяет доступность ключа HASP HL;
- при успешной проверке вычисляет определенный интеграл функции  $y = x^2$  на заданном интервале;
- генерирует последовательность чисел Фибоначчи и записывает/считывает их из памяти ключа.

3. На этом шаге создается в папке с программой проект Borland Delphi, размещаются на форме программы необходимые элементы, как показано на рис. 48.

4. После этого создаются процедуры `procedure TForm1.FormCreate(Sender: TObject)` и `procedure TForm1.Button1Click(Sender: TObject)`.

5. Текст программного модуля `unit1.pas`:

```
unit Unit1;

interface

uses

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, hasp_unit;

type
    TForm1 = class(TForm)
        Button1: TButton;
        Edit1: TEdit;
        Edit2: TEdit;
        Memo1: TMemo;
        Edit3: TEdit;
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        Edit4: TEdit;
        Label4: TLabel;
        procedure FormCreate(Sender: TObject);
        procedure Button1Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
const
    AES_MIN_DATA_LEN = 16; {минимальная длина данных для AES
шифрования}
```

MAX\_MEMORY = 128; {Макс. размер памяти, который будет обрабатываться}

CUSTOM\_FEATURE = 42;

DEMO\_MEMBUFFER\_SIZE = 8;

SCOPE\_VIEW\_MAXSIZE = (1024 \* 128);

var

Form1: TForm1;

a,b,s,l,d,e,raz,m:real;

n,i,x:integer;

data : array[0..16] of byte =

(\$74, \$65, \$73, \$74, \$20, \$73, \$74, \$72,  
\$69, \$6e, \$67, \$20, \$31, \$32, \$33, \$34, 0);

membuffer : array [0..DEMO\_MEMBUFFER\_SIZE] of byte;

datalen : hasp\_size\_t;

implementation

{ \$R \*.dfm }

{ \$I hasp\_vcode.inc }

function f(x,m:real):real;

begin

f:=x\*x;

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

Memo1.Clear();

end;

procedure TForm1.Button1Click(Sender: TObject);

const

SCOPE\_VIEW\_MAXSIZE = (1024 \* 128);

var status : hasp\_status\_t;

handle : hasp\_handle\_t;

```

handle2 : hasp_handle_t;
fsize   : hasp_size_t;
time    : hasp_time_t;
info    : pchar;
feature  : integer;
day, month, year, hour, minute, second : cardinal;
scope : Array[0..SCOPE_VIEW_MAXSIZE-1] of char;
view   : Array[0..SCOPE_VIEW_MAXSIZE-1] of char;
begin
  Memo1.Clear();
  feature:= 50;
  status := hasp_login(feature, @vendor_code[1], handle);
  if status = HASP_STATUS_OK then
    begin
      Memo1.Lines.Add('Проверка завершена успешно');
      raz:=100;
      e:=StrToFloat(Edit4.Text);
      n:=100;
      s1:=0;
      s:=200;
      a:=StrToFloat(Edit1.Text);
      b:=StrToFloat(Edit2.Text);
      m:=b-1;
      d:=(b-a)/n;
      while raz>e do
        begin
          for i:=0 to n do
            S1:=S1+d*(F(a + I*d, m) + F(a + (I+1)*d, m))/2;
            raz:=abs(s-s1);
            s:=s1;

```



```

    s1:=0;
    Memo1.Lines.Add(FloatToStr(s));
    n:=n*2;
    d:=(b-a)/n;
    end;
    Edit3.Text:=FloatToStr(s);
end
else if status = HASP_FEATURE_NOT_FOUND then
    Showmessage('Невозможно подключиться к feature')
else if status = HASP_CONTAINER_NOT_FOUND then
    Showmessage('HASP ключ с данным кодом не найден')
else if status = HASP_OLD_DRIVER then
    Showmessage('Устаревшая версия драйвера аппаратного ключа защиты')
else if status = HASP_NO_DRIVER then
    Showmessage('Драйвер ключа не установлен')
else if status = HASP_INV_VCODE then
    Showmessage('Ошибка в коде ключа')
else
    Showmessage('Подключение к feature' + IntToStr(feature) + ' закончилось
неудачей, статус' + IntToStr(status));
    n:=1;
    x:=0;
    for i:=1 to DEMO_MEMBUFFER_SIZE do
    begin
        membuffer[i]:=x;
        x:=x+n;
        n:=membuffer[i];
    end;
    status := hasp_write(handle, HASP_FILEID_RW, 0, fsize, membuffer);
    status := hasp_read(handle, HASP_FILEID_RW, 0, fsize, membuffer);

```

```

case status of
  HASP_STATUS_OK:
    begin
      Memo1.Lines.Add('Чтение памяти выполнено успешно');
      for i:=1 to DEMO_MEMBUFFER_SIZE do
        Memo1.Lines.Add(IntToStr(membuffer[i]));
      end;
    else
      begin
        Memo1.Lines.Add('Чтение памяти не было произведено');
      end;
    end;
status := hasp_logout(handle);
case status of
  HASP_STATUS_OK:
    begin
      Memo1.Lines.Add('Сеанс работы с ключом успешно завершен!');
    end;
  else
    begin
      Memo1.Lines.Add('Сеанс не был открыт');
    end; end; end; end.

```

6. На этом завершающем шаге необходимо скомпилировать программу и проверить ее работоспособность при подключенном и отсутствующем ключе HASP HL. Полученную информацию о работе с программой отразить в отчете о выполненной работе.

## **Контрольные вопросы**

1. Что такое Sentinel HASP API? Какие возможности он предоставляет?
2. Как можно классифицировать функции Sentinel HASP API?
3. Перечислите вспомогательные файлы HASP API, необходимые для функционирования защиты?
4. Возможно ли добавление в код программы функций различных категорий?
5. Где храниться идентификатор ключа HASP?

## **Требования к содержанию отчета**

Отчет выполняется на листах формата А4, которые должны быть скреплены перед сдачей отчета преподавателю. Первый лист отчета титульный – содержит информацию об учебном заведении, кафедре, названии дисциплины, теме выполненной работы. Приводится информация об учащемся, выполнившим работу. Приводится информация о преподавателе, проверяющем работу. В нижней части листа информация о месте (городе), где находится учебное заведение и год выполнения работы.

В содержательной части отчета о проделанной работе необходимо указать номер работы и тему работы. Затем описать ход выполнения работы. Желательно проиллюстрировать ход выполнения работы скриншотами (копиями экрана), полученными в процессе выполнения работы.

Если преподавателем были заданы вопросы по выполнению работы, на которые предложено ответить письменно, то приводятся вопросы и ответы на них.

Завершается отчет выводами по проделанной работе. Отмечаются достоинства и недостатки изученной технологии, область использования разработки и т.д.

Желательно привести в отчете список использованной литературы и интернет ресурсов (сайтов сети Интернет).

## ЗАЩИТА DLL БИБЛИОТЕК С ПОМОЩЬЮ SENTINEL HASP API

### Цель работы

Разработать программу средствами Delphi, использующую динамически подключаемую библиотеку DLL, защищенную с помощью Sentinel HASP.

### Краткие сведения о DLL

DLL (англ. dynamic-link library — «библиотека динамической компоновки», «динамически подключаемая библиотека») — понятие операционных систем Microsoft Windows и IBM OS/2, динамическая библиотека, позволяющая многократное использование различными программными приложениями. К DLL относятся также элементы управления ActiveX и драйверы. Формат файлов DLL придерживается тех же соглашений, что и формат исполняемых файлов, сочетая код, таблицы и ресурсы, отличаясь лишь интерпретацией некоторых полей.

Аналогично защите основного кода программы, Sentinel HASP API позволяет защитить и DLL библиотеку. Это подразумевает под собой внедрение функций HASP API в код динамических библиотек. При этом разработчик программного обеспечения имеет возможность пользоваться всеми доступными функциями (рассмотрены в работе 2).

1. Программа, как и в лабораторной работе 2, должна проверять доступность аппаратного ключа HASP и в случае успешной проверки вычислять определенный интеграл функции  $y = x^2$  на заданном интервале. Различие состоит в том, что теперь функция вычисления интеграла вынесена в отдельную DLL библиотеку и будет импортирована из нее в процессе работы.

Поскольку в эту библиотеку будет устанавливаться встроенная защита, то необходимо добавить в папку с проектом вспомогательные файлы, выполнив пункт 1 задания к лабораторной работе 2.

2. Далее необходимо запустить программу Borland Delphi и создать новый объект DLL Wizard, File->New->DLL Wizard, как показано на рисунке 49.

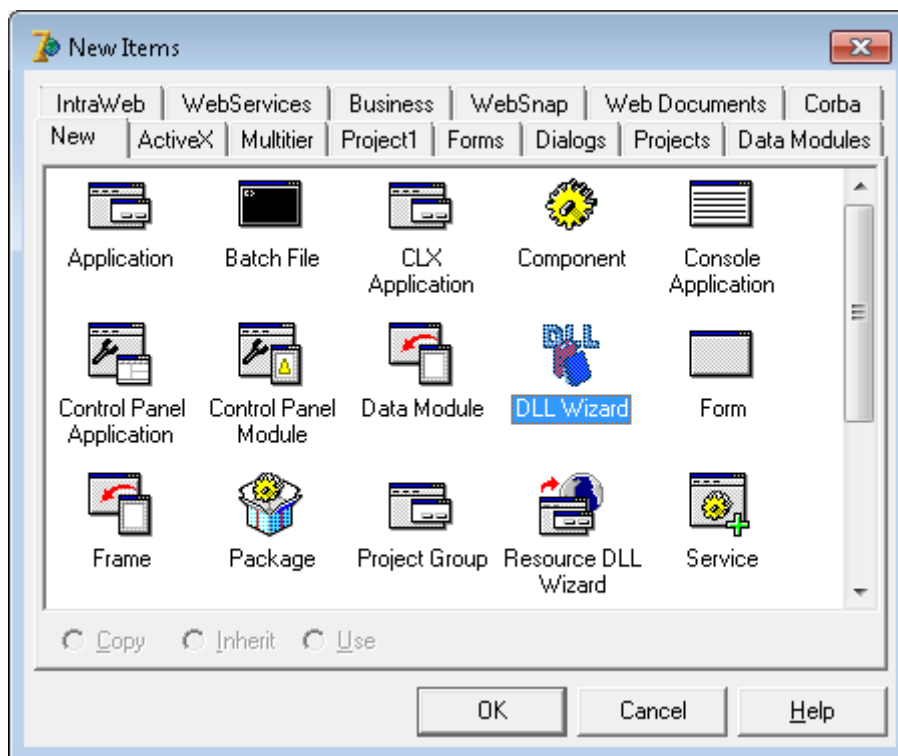


Рисунок 49 – создание DLL библиотеки

3. После сохранения проекта, следует приступить к добавлению в программный модуль необходимых функций. В данном случае это процедура `integral`, которая будет в дальнейшем будет вызываться из DLL библиотеки, и функция `f`. Исходный код программного модуля:

```
library Project1;
```

```
uses
```

```
    SysUtils,
```

```
    Classes,
```

```
    hasp_unit;
```

```

{$R *.res}

{$I hasp_vcode.inc }

function f(x,m:real):real;
begin
f:=x*x;
end;

procedure integral(a,b,e:real; n:integer; var s:real; var status : hasp_status_t);
const
    SCOPE_VIEW_MAXSIZE = (1024 * 128);
var
    handle : hasp_handle_t;
    handle2 : hasp_handle_t;
    fsize : hasp_size_t;
    time : hasp_time_t;
    info : pchar;
    feature,i : integer;
    day, month, year, hour, minute, second : cardinal;
    scope : Array[0..SCOPE_VIEW_MAXSIZE-1] of char;
    view : Array[0..SCOPE_VIEW_MAXSIZE-1] of char;
    s1,d,m,raz:real;
begin
    feature:= 50;
    status := hasp_login(feature, @vendor_code[1], handle);
    if status = HASP_STATUS_OK then
    begin
        s1:=0;
        s:=200;
        raz:=100;
        m:=b-1;
        d:=(b-a)/n;

```

```

while raz>e do
begin
  for i:=0 to n do
    S1:=S1+d*(F(a + I*d, m) + F(a + (I+1)*d, m))/2;
    raz:=abs(s-s1);
    s:=s1;
    s1:=0;
    n:=n*2;
    d:=(b-a)/n;
  end;
end;
end;
exports integral;
begin
end.

```

4. Необходимо обратить внимание, что экспортируется только процедура `integral`. Также можно увидеть в коде DLL библиотеки вызов функции `hasp_login`, что позволит проверить наличие подлинного ключа защиты. Теперь необходимо сформировать DLL-библиотеку, выполнив действия Project->Build All Projects. В папке с проектом появится файл `Project1.dll`, который нужно скопировать в папку основной программы и переименовать в `HASP_INT.dll`.

5. Далее следует приступить к формированию основной программы. Интерфейс пользователя не изменился, он представлен на рисунке 48. Однако код программы теперь выглядит иначе, поскольку были вынесены необходимые математические функции в отдельную DLL-библиотеку. Содержание программного модуля `unit1.pas`:

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

```

```

Dialogs, StdCtrls, hasp_unit;

type
  TForm1 = class(TForm)
    Button1: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Memo1: TMemo;
    Edit3: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Edit4: TEdit;
    Label4: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
const
  AES_MIN_DATA_LEN      = 16; { минимальная длина данных для AES
шифрования }
  MAX_MEMORY             = 128; { Макс. размер памяти, который будет
обрабатываться }
  CUSTOM_FEATURE         = 42;
  DEMO_MEMBUFFER_SIZE   = 8;
  SCOPE_VIEW_MAXSIZE = (1024 * 128);
var
  Form1: TForm1;

```



```

a,b,s,d,e:real;
n:integer;
status : hasp_status_t;
integral: procedure(a,b,e:real; n:integer; var s:real; var status : hasp_status_t);
LibHandle: THandle;
data : array[0..16] of byte =
    ($74, $65, $73, $74, $20, $73, $74, $72,
    $69, $6e, $67, $20, $31, $32, $33, $34, 0);
membuffer : array [0..DEMO_MEMBUFFER_SIZE] of byte;
datalen : hasp_size_t;
implementation
{$R *.dfm}
{$I hasp_vcode.inc }
procedure TForm1.FormCreate(Sender: TObject);
begin
Memo1.Clear();
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
    @integral := nil;
    LibHandle := LoadLibrary('HASP_INT.dll');
    if LibHandle >= 32 then begin
        @integral := GetProcAddress(LibHandle,'integral');
        if @integral <> nil then
            Memo1.Clear();
            n:=100;    a:=StrToFloat(Edit1.Text);    b:=StrToFloat(Edit2.Text);
            e:=StrToFloat(Edit4.Text);    integral(a,b,e,n,s,status);
            if status = HASP_STATUS_OK then    begin
                Memo1.Lines.Add(FloatToStr(s));
            end
        end
    end
end

```

```

else if status = HASP_FEATURE_NOT_FOUND then
    Showmessage('Невозможно подключиться к feature')
else if status = HASP_CONTAINER_NOT_FOUND then
    Showmessage('HASP ключ с данным кодом не найден')
else if status = HASP_OLD_DRIVER then
    Showmessage('Устаревшая версия драйвера аппаратного ключа защиты')
else if status = HASP_NO_DRIVER then
    Showmessage('Драйвер ключа не установлен')
else if status = HASP_INV_VCODE then
    Showmessage('Ошибка в коде ключа')
else
    Showmessage('Подключение к feature закончилось неудачей');
end; FreeLibrary(LibHandle);
end;
end.

```

5. Важно обратить внимание на то, что в коде основной программы нет вызова функций Sentinel HASP API, а также присутствует механизм импорта функций из DLL библиотеки HASP\_INT.dll

6. Теперь необходимо скомпилировать программу и проверить ее работоспособность при подключенном и отсутствующем ключе HASP HL. Результаты описать в выводе к отчету.

### **Контрольные вопросы**

1. Объясните принцип работы DLL библиотек?
2. Какие преимущества дает использование DLL библиотек? Есть ли недостатки и если есть, то какие?
3. Опишите процесс создания динамически подключаемых библиотек в среде Borland Delphi?
4. Возможно ли использование нескольких DLL библиотек одной программой?

## **Требования к содержанию отчета**

Отчет выполняется на листах формата А4, которые должны быть скреплены перед сдачей отчета преподавателю. Первый лист отчета титульный – содержит информацию об учебном заведении, кафедре, названии дисциплины, теме выполненной работы. Приводится информация об учащемся, выполнившим работу; информация о преподавателе, проверяющем работу. В нижней части листа информация о месте (городе), где находится учебное заведение и год выполнения работы.

В содержательной части отчета о проделанной работе необходимо указать номер работы и тему работы. Затем описать ход выполнения работы. Желательно проиллюстрировать ход выполнения работы скриншотами (копиями экрана), полученными в процессе выполнения работы.

Если преподавателем были заданы вопросы по выполнению работы, на которые предложено ответить письменно, то приводятся вопросы и ответы на них.

Завершается отчет выводами по проделанной работе. Отмечаются достоинства и недостатки изученной технологии, область использования разработки и т.д.

Желательно привести в отчете список использованной литературы и интернет ресурсов (сайтов сети Интернет).

## ЛИТЕРАТУРА

1. <http://www.aladdin-rd.ru/catalog/hasp/> Sentinel HASP — защита, лицензирование и распространение программного обеспечения
2. <http://www.aladdin-rd.ru/buy/?pid=4981> Цены и заказ продуктов
3. <http://www.softkey.ru/catalog/program.php?chsph=46998645&ID=46511>  
HASP SRM - защита, лицензирование и распространение программного обеспечения (купить)
4. <http://protect-online.ru/> Сервис лицензирования и защиты ПО (Sentinel HASP SL)
5. <http://www.aladdin-rd.ru/support/downloads/get?ID=27822> Скачать комплект разработчика HASP (образ диска 1.41 Гб) Версия 5.12 рус.
6. <http://www.aladdin-rd.ru/support/training/authentication> Учебное пособие "Аутентификация: от А до Я"
7. [http://www.aladdin-rd.ru/upload/Authentication\\_Book.zip](http://www.aladdin-rd.ru/upload/Authentication_Book.zip) Загрузить главы книги "Аутентификация: теория и практика обеспечения безопасного доступа к информационным ресурсам",
8. Шелупанов А. Аутентификация. Теория и практика обеспечения безопасного доступа к информационным ресурсам. / А. Шелупанов, С. Груздев, Ю. Нахаев, 552 стр., Издательство: Горячая Линия – Телеком, ISBN 978-5-9912-0110-0; 2009 г.

## Содержание

Введение.....	3
1. Лабораторная работа 1. Защита исполняемых exe файлов с помощью навесной защиты HASP .....	7
1.1 Подготовительный этап .....	7
1.2 Создание компонентов .....	8
1.3 Установка защиты ПО.....	13
1.4 Формирование программного пакета .....	19
1.5 Создание условно-бесплатной версии .....	25
1.6 Оформление и обработка заказов .....	32
1.7 Работа с защищенным приложением на стороне пользователя ...	40
2. Лабораторная работа 2. Организация встроенной защиты программы, написанной на Delphi с помощью SENTINEL HASP API .....	48
3. Лабораторная работа 3. Защита DLL библиотек с помощью Sentinel HASP .....	60
Литература .....	68

Кафедра системного анализа

Методические указания

# ИСПОЛЬЗОВАНИЕ КЛЮЧЕЙ СЕРИИ HASP HL ДЛЯ ЗАЩИТЫ ИНФОРМАЦИИ. ЗАЩИТА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Игорь Викторович Ананченко

---

Отпечатано с оригинал-макета. Формат 60×90.<sup>1</sup>/<sub>16</sub>

Печ. л. 4.2 Тираж \_\_\_\_ экз.

---

Санкт-Петербургский государственный технологический институт  
(Технический университет)

---

190013, Санкт-Петербург, Московский пр., 26  
Типография издательства СПбГТИ(ТУ) 49-49-365