HASP®

PROFESSIONAL SOFTWARE PROTECTION

Cross Platform

Руководство разработчика

Aladdin®

SECURING THE GLOBAL VILLAGE

PROFESSIONAL SOFTWARE PROTECTION

Руководство разработчика

Версия 11

АВТОРСКИЕ ПРАВА И ТОРГОВЫЕ МАРКИ

ОГРАНИЧЕННАЯ ГАРАНТИЯ, ОГРАНИЧЕНИЕ ОТВЕТСТВЕННОСТИ И ВОЗМЕЩЕНИЯ

Гарантия Aladdin Knowledge Systems Ltd. (Aladdin) распространяется на программное обеспечение и устройства HASP на период двенадцати (12) месяцев со дня покупки в порядке, указанном в Лицензионном соглашении с Разработчиком.

Гарантийные обязательства и ограничения ответственности Aladdin указаны в Лицензионном соглашении.

ЗА ИСКЛЮЧЕНИЕМ УКАЗАННОГО ВЫШЕ, НЕ СУЩЕСТВУЕТ ИНЫХ ЯВНЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ ГАРАНТИЙ НА ПРОДУКТЫ ALADDIN, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ТАКОВОЙ, ПОДРАЗУМЕВАЕМУЮ ГАРАНТИЮ СПРОСА И ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЁННОЙ ЦЕЛИ.

Были приложены все усилия, чтобы информация в данном документе была полной и точной. Aladdin не несёт ответственности за прямые или косвенные убытки или ущерб в результате неточностей или упущений.

Спецификации в данном документе могут быть изменены без уведомления.

ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ МЕЖДУ РАЗРАБОТЧИКОМ И ALADDIN KNOWLEDGE SYSTEMS LTD.

ВАЖНАЯ ИНФОРМАЦИЯ – ПОЖАЛУЙСТА, ВНИМАТЕЛЬНО ПРОЧТИТЕ ДАННОЕ СОГЛАШЕНИЕ ПРЕЖДЕ, ЧЕМ ОТКРЫТЬ ПАКЕТ И/ ИЛИ ИСПОЛЬЗОВАТЬ ЕГО СОДЕРЖИМОЕ И/ ИЛИ ПРЕЖДЕ, ЧЕМ ЗАГРУЖАТЬ ИЛИ УСТАНАВЛИВАТЬ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.

ВСЕ УКАЗАНИЯ ПО ИСПОЛЬЗОВАНИЮ ПРОДУКТОВ HASP (включая без ограничений Комплект Разработчика, библиотеки, утилиты, дискеты, CD-ROM, устройства HASP® и Руководство Разработчика) (далее «Продукт»), ПРЕДОСТАВЛЯЕМЫЕ КОМПАНИЕЙ ALADDIN KNOWLEDGE SYSTEMS LTD. (или любым дочерним предприятием – каждое из них упоминаемое как «ALADDIN»), ПОДЧИНЯЮТСЯ И БУДУТ ПОДЧИНЯТЬСЯ УСЛОВИЯМ, ОГОВОРЕННЫМ В ДАННОМ СОГЛАШЕНИИ.

ОТКРЫВАЯ ПАКЕТ, СОДЕРЖАЩИЙ ПРОДУКТ, И/ ИЛИ ЗАГРУЖАЯ ДАННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ (как определено далее по тексту) И/ ИЛИ УСТАНАВЛИВАЯ ДАННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ НА ВАШ КОМПЬЮТЕР И/ ИЛИ ИСПОЛЬЗУЯ ДАННЫЙ ПРОДУКТ, ВЫ ПРИНИМАЕТЕ ДАННОЕ СОГЛАШЕНИЕ И СОГЛАШАЕТЕСЬ С ЕГО УСЛОВИЯМИ.

ЕСЛИ ВЫ НЕ СОГЛАСНЫ С ДАННЫМ СОГЛАШЕНИЕМ, НЕ ОТКРЫВАЙТЕ ЭТОТ ПАКЕТ И/ ИЛИ НЕ ЗАГРУЖАЙТЕ И/ ИЛИ НЕ УСТАНАВЛИВАЙТЕ ДАННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И НЕМЕДЛЕННО (по меньшей мере, в течение 7 дней с даты получения этого пакета) ВЕРНИТЕ ЭТОТ ПРОДУКТ В ALADDIN, СОТРИТЕ ДАННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ВСЕ ЕГО ЧАСТИ СО СВОЕГО КОМПЬЮТЕРА И НЕ ИСПОЛЬЗУЙТЕ ЕГО НИКОИМ ОБРАЗОМ. ПРИ ВОЗВРАЩЕНИИ ПРОДУКТА С КОПИЕЙ ПЛАТЕЖНОГО ДОКУМЕНТА ВАМ БУДЕТ ВОЗВРАЩЕНА ЕГО СТОИМОСТЬ.

1. Права и Собственность

ДАННОЕ ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ НЕ ЯВЛЯЕТСЯ СОГЛАШЕНИЕМ О ПРОДАЖЕ. Программное обеспечение пакета Aladdin's HASP Product Development Kit, включая все переработки, исправления, модификации, дополнения,

обновления и/ или усовершенствования к ней (далее по всему тексту и любой его части определяемое как «Программное обеспечение»), и связанная с ним документация НЕ ПРЕДНАЗНАЧЕНЫ ДЛЯ ПРОДАЖИ и являются и останутся безраздельной собственностью Aladdin. Все права на интеллектуальную собственность и на Продукт (включая, без ограничений, авторские права, профессиональные секреты, торговые марки и т.д.), подтвержденные или включенные в приложенные/ взаимосвязанные/ имеющие отношение к данному руководству, данные, содержащиеся в нём, являются и будут являться собственностью исключительно компании Aladdin. Данное Соглашение не передает Вам права на Программное обеспечение, а лишь предоставляет ограниченное право на использование, которое подлежит отмене согласно условиям данного Лицензионного соглашения. Ничто в данном Соглашении не подтверждает отказ Aladdin от прав на интеллектуальную собственность по какому бы то ни было закону.

2. Лицензия

По выплате взносов за права, Aladdin настоящим предоставляет Вам, а Вы принимаете личную, неэксклюзивную и полностью могущую быть отозванной ограниченную Лицензию на использование данного Программного обеспечения только в исполнительной форме, как описано в прилагаемой к Программному обеспечению документации и только в соответствии с условиями данного Соглашения:

(i) Вы можете установить Программное обеспечение и использовать его на компьютерах, расположенных на месте Вашего бизнеса, как описано в соответствующей документации от Aladdin;

(ii) Вы можете путём слияния присоединить Программное обеспечение к программам Вашего компьютера с единственной целью, описанной в данном Руководстве программиста; однако любая часть Программного обеспечения, объединённая с другой компьютерной программой, неизбежно будет считаться его производной и будет по-прежнему соотносится с условиями настоящего Соглашения.

(iii) Вы можете создать определенное количество копий Программного обеспечения – не более трех (3) – исключительно в целях резервного копирования и для нужд разработки. Программное обеспечение не должно использоваться в любых иных целях.

3. Передача лицензии

В случае объединения данного Программного обеспечения с иными компьютерными программами согласно разделу 2 Вы можете передать разрешение в соответствии с условиями данного Соглашения на использование подключенного Программного обеспечения и перепродать аппаратное обеспечение, состоящее из устройств HASP®, которые Вы приобрели у Aladdin, дистрибьюторам и/ или пользователям.

Предваряя подобную продажу и передачу лицензии, Вы обязаны включить особую ссылку в Ваш контракт с таковыми дистрибьюторами и/ или пользователями, а также во всём остальном обеспечить подчинение дистрибьюторов и/ или пользователей условиям гарантии и лицензии, оговоренным в данном Соглашении компанией Aladdin.

4. Запрещённое использование

За исключением указанного в разделах 1, 2 и 3, Вы соглашаетесь:

(i) не использовать, не модифицировать Программное обеспечение, не передавать разрешение на данное Программное обеспечение и любые другие Продукты Aladdin, за исключением явно указанного в данном Соглашении и Руководстве программиста;

(ii) не продавать, не передавать лицензию, не сдавать в аренду, не передавать, не переводить, не закладывать, не разделять Ваши права в рамках данного Разрешения с кем/ кому-либо ещё;

(iii) не модифицировать, не демонтировать, не декомпилировать, не реконструировать, не видоизменять и не расширять данное Программное обеспечение и не пытаться раскрыть исходную систему кодирования данного Программного обеспечения;

(iv) не помещать данное Программное обеспечение на сервер с возможностью доступа к нему через открытую сеть;

(v) не использовать какие бы то ни было резервные или архивные копии данного Программного обеспечения (или позволять кому-либо ещё использовать такие копии) с любой иной целью, кроме замены оригинальной копии, в случае его разрушения или дефектности.

Если Вы являетесь членом Европейского Союза, данное соглашение не затрагивает Ваших прав согласно любому законодательству, осуществляющему Директиву Совета ЕС о Правовой Защите Компьютерных Программ. Если Вы стремитесь получить какую бы то ни было информацию в рамках значения данной Директивы, прежде всего Вам следует обратиться в Aladdin.

5. Ограниченная гарантия

Aladdin гарантирует, что:

(i) данное Программное обеспечение с момента доставки его Вам в течение трёх (3) месяцев будет работать в достаточном соответствии с Руководством разработчика, при условии, что оно будет использовать на компьютерном аппаратном обеспечении и с операционной системой, для которой оно было разработано;

(ii) что устройство HASP® в течение двенадцати (12) месяцев с даты поставки будет в достаточной мере избавлено от значительных дефектов в материалах и конструктивных характеристиках.

6. Отказ от гарантии

ALADDIN НЕ ГАРАНТИРУЕТ, ЧТО ЛЮБОЕ ИЗ ЕГО ИЗДЕЛИЙ БУДЕТ СООТВЕТСТВОВАТЬ ВАШИМ ТРЕБОВАНИЯМ, ИЛИ ЧТО ЕГО РАБОТА БУДЕТ БЕСПЕРЕБОЙНОЙ ИЛИ БЕЗОШИБОЧНОЙ. В РАМКАХ ЗАКОНА ALADDIN ОТКРЫТО ОТКАЗЫВАЕТСЯ ОТ ВСЕХ ГАРАНТИЙ, НЕ ОГОВОРЕННЫХ ЗДЕСЬ И ВСЕХ ПОДРАЗУМЕВАЕМЫХ ГАРАНТИЙ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ТАКОВОЙ, ПОДРАЗУМЕВАЕМУЮ ГАРАНТИЮ СПРОСА И ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЁННОЙ ЦЕЛИ. НИ ОДИН ИЗ ДИЛЕРОВ, ДИСТРИБЬЮТЕРОВ,

ПРОДАВЦОВ, АГЕНТОВ ИЛИ СОТРУДНИКОВ ALADDIN НЕ УПОЛНОМОЧЕН ПРОИЗВОДИТЬ МОДИФИКАЦИИ, РАСШИРЕНИЯ ИЛИ ДОПОЛНЕНИЯ К ДАННОЙ ГАРАНТИИ. Если Вы произвели какие-либо модификации Программного обеспечения или любой из частей данного Продукта во время гарантийного периода; если средства и устройство HASP® подвергаются аварии, неосторожному или неправильному обращению; или если Вы нарушаете любое из условий настоящего Соглашения, то гарантия, упомянутая выше в разделе 5, будет немедленно прекращена. Гарантия не действительна, если Программное обеспечение используется на или в сочетании с аппаратным обеспечением или программой иной, чем немодифицированная версия аппаратного обеспечения или программы, в сочетании с которыми по своей конструкции должно было использоваться данное Программное обеспечение, как описано в данном Руководстве разработчика.

7. Ограничение возмещения

В случае нарушения гарантии, оговоренной выше, единственным обязательством Aladdin будет, руководствуясь лишь усмотрением Aladdin:

(i) заменить или бесплатно отремонтировать Продукт или его составляющие, если это не противоречит вышеупомянутому ограничению гарантии;

(i) возместить стоимость, выплаченную Вами за Продукт или его составляющие. Любая замененная или отремонтированная составляющая будет на гарантии или в течение оставшегося времени от начального гарантийного периода, или в течение 30 дней, если срок начального гарантийного периода истекает ранее. Гарантийные требования должны быть выставлены в письменном виде в течение гарантийного периода, но не позднее семи (7) дней с момента обнаружения дефекта, и содержать в себе подтверждения, удовлетворяющие Aladdin. Вся Продукция должна быть возвращена дистрибьютору, через которого была совершена покупка (если покупка состоялась не непосредственно в Aladdin), и отправлена возвращающей стороной с оплаченной стоимостью перевозки и страховки. Продукция или составляющие таковой должны быть отправлены с копией Вашего чека.

8. Исключение косвенных убытков

Стороны признают, что Продукт по сути своей сложный и не может быть полностью лишен ошибок. ALADDIN НЕ НЕСЁТ ОТВЕТСТВЕННОСТИ (КАК ПО ДОГОВОРУ, ДЕЛИКТУ, ВКЛЮЧАЯ ХАЛАТНОСТЬ, ТАК И В ЛЮБОЙ ИНОЙ ФОРМЕ) ПЕРЕД ВАМИ ИЛИ ЛЮБОЙ ТРЕТЬЕЙ СТОРОНОЙ ЗА ЛЮБЫЕ ПОТЕРИ ИЛИ УБЫТКИ (ВКЛЮЧАЯ КОСВЕННЫЕ, ЭКСТРЕННЫЕ ИЛИ ПОБОЧНЫЕ УБЫТКИ), ВКЛЮЧАЯ, БЕЗ ОГРАНИЧЕНИЙ, ЛЮБЫЕ ПОТЕРИ ИЛИ УБЫТКИ ПРИБЫЛЬНОСТИ БИЗНЕСА, ПОТЕРЮ ДОХОДНОСТИ ИЛИ ПРЕСТИЖА, ПОТЕРЯННУЮ ИЛИ ПОВРЕЖДЁННУЮ ИНФОРМАЦИЮ ИЛИ ДОКУМЕНТАЦИЮ ВСЛЕДСТВИЕ КАКОГО-ЛИБО ИСПОЛЬЗОВАНИЯ ДАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И/ИЛИ ЛЮБОЙ СОСТАВЛЯЮЩЕЙ ДАННОЙ ПРОДУКЦИИ, ДАЖЕ ЕСЛИ ALADDIN ОСВЕДОМЛЁН О ВОЗМОЖНОСТИ ПОДОБНЫХ УБЫТКОВ.

9. Ограничение ответственности

В СЛУЧАЕ, ЕСЛИ, НЕСМОТРЯ НА УСЛОВИЯ ДАННОГО СОГЛАШЕНИЯ, ALADDIN ПРИЗНАН ОТВЕТСТВЕННЫМ ЗА УБЫТКИ НА ОСНОВАНИИ КАКИХ-ЛИБО ДЕФЕКТОВ ИЛИ НЕСООТВЕТСТВИЯ ЕГО ПРОДУКЦИИ, ПОЛНАЯ ОТВЕТСТВЕННОСТЬ ЗА КАЖДУЮ ДЕФЕКТНУЮ ЕДИНИЦУ ПРОДУКЦИИ НЕ БУДЕТ ПРЕВЫШАТЬ СУММУ, ВЫПЛАЧЕННУЮ КОМПАНИИ ALADDIN ЗА ЭТУ ДЕФЕКТНУЮ ЕДИНИЦУ ПРОДУКЦИИ.

10. Никаких иных гарантий

За исключением вышесказанного, в соответствии с преамбулой, Aladdin не дает никаких ни явных, ни подразумеваемых гарантий на качество и производительность продукта, а также на его соответствие при использовании тем или иным способом.

11. Прекращение действия

В случае невыполнения Вами условий данного Соглашения действие Вашей Лицензии и настоящего Соглашения будет прекращено. По прекращении действия данного Лицензионного Соглашения, произведённому компанией

Aladdin: (i) Разрешение, предоставленное Вам в данном Соглашении, будет недействительным, и Вы, по прекращении срока действия, не сможете продолжать дальнейшее использование данного Программного обеспечения и других лицензионных Продуктов; (ii) Вы незамедлительно вернёте компании Aladdin всё материальное имущество, представляющее право Aladdin на интеллектуальную собственность и все копии такового и/или сотрёте/удалите любую информацию, содержащуюся в них в электронном виде. Разделы 1, 4, 6-12 будут продолжать действовать даже в случае прекращения действия настоящего Соглашения.

12. Действующий Закон и Законодательство

Данное Соглашение должно быть истолковано и определено в соответствии с законами Израиля (за исключением конфликта применения правовых норм), и только израильский суд уполномочен отправлять правосудие в любых конфликтах и спорах, возникающих из данного Соглашения. Применение Конвенции Объединённых Наций о Договорах Международной Продажи Товаров (the United Nations Convention of Contracts for the International Sale of Goods) однозначно исключается. Неспособность любой из сторон привести в исполнение любое из прав, предоставленных по данному тексту, или принять меры против другой стороны в случае любого нарушения, обозначенного по тексту, не должно рассматриваться как отказ этой стороны от последующего приведения в исполнение прав или совершения последующих действий в случае дальнейших нарушений.

13. Программное обеспечение третьих сторон

Если Продукт содержит в себе любое программное обеспечение, предоставленное какой-либо третьей стороной, такое программное обеспечение от третьей стороны предоставляется «как оно есть» без какой-либо гарантии, и разделы 2, 3, 6, 8, 9-13 настоящего Соглашения применяются ко всем поставщикам программного обеспечения от таких третьих сторон и к программному обеспечению от третьих сторон, как если бы они были Aladdin и Продукт соответственно.

14. Разное

Настоящее Соглашение представляет собой полное соглашение, касающееся Разрешения, и может быть изменено только посредством письменного соглашения, данного обеими сторонами. Если выполнение какого-либо условия настоящего Соглашения представляется невозможным, такое условие будет скорректировано только в пределах, обеспечивающих возможность выполнения данного условия.

Я ПРОЧЁЛ И ПОНЯЛ НАСТОЯЩЕЕ СОГЛАШЕНИЕ О РАЗРЕШЕНИИ И СОГЛАСЕН С ИСПОЛНЕНИЕМ ВСЕХ ЕГО УСЛОВИЙ.

Соответствие CE

Вся продукты HASP соответствуют Директиве CE EMC и соответствующим стандартам*. Продукты HASP маркируется логотипом CE, а CE карта соответствия HASP включается в каждый комплект поставки или предоставляется по запросу.

*EMC директива 89/336/EEC и соответствующие стандарты EN 55022, EN 50082-1.

Соответствие FCC

Уполномоченные органы FCC определили, что HASP не является Периферийным вычислительным устройством класса «B» и, таким образом, не требует соответствия нормам FCC.

Соответствие Y2K

Все продукты линейки HASP соответствуют стандарту Y2K. Соответствие этому стандарту означает, что все устройства HASP будут корректно сохранять, хранить, обрабатывать и представлять даты, следующие за первым января 2000 года с сохранением функциональных возможностей и таким же образом, как и даты до 31-го декабря 1999 года.

Карта соответствия Y2K включается во все комплекты поставки, либо предоставляется по запросу.

Соответствие UL

Все продукты HASP успешно прошли тесты UL 94 на воспламеняемость пластических материалов для частей устройств и приборов. Продукты HASP соответствует нормам безопасности оборудования информационных технологий UL 1950 (UL 1950 Safety of Information Technology Equipment regulations).

Соответствие ISO9002

Все продукты HASP разработаны и выпускаются компанией Aladdin Knowledge Systems, сертифицированной по стандарту ISO 9002.

Система обеспечения качества Aladdin одобрена Международной организацией по стандартизации (ISO), что является подтверждением того, что стандарты выпускаемых продуктов и оказываемых компанией Aladdin услуг постоянно соответствуют спецификациям, предназначенным для обеспечения полного удовлетворения клиентов.

Сертификат соответствия

По запросу Aladdin Knowledge Systems предоставит Сертификат соответствия любому разработчику программного обеспечения, который желает продемонстрировать, что все продукты HASP соответствуют установленным спецификациям. Разработчики программного обеспечения могут предоставлять этот сертификат конечным пользователям вместе со своими программами.

Оглавление

Список таблиц

О руководстве программиста HASP4

Руководство программиста HASP было создано, чтобы помочь разработчикам программного обеспечения защитить их приложения методами, которые полностью удовлетворяют их потребностям.

В первой части («Общие сведения», стр. 1) приводятся общие сведения о системе защиты с помощью HASP, описывается различное аппаратное обеспечение HASP, а также предоставляется детальное описание процесса установки программного обеспечения HASP. Информация данной части необходима для любого разработчика, независимо от используемого метода защиты и конкретной модели ключа HASP.

Вторая часть («Использование инструментария HASP», стр. 41) призвана помочь в освоении утилит HASP для Windows и Mac. С помощью инструментов, описываемых в данной части, вы сможете защитить приложения с минимальными усилиями, запрограммировать ключи HASP до передачи клиентам. Здесь же приводятся некоторые сведения, облегчающие техническую поддержку клиентов. Информация данной части необходима для любого разработчика, независимо от используемого метода защиты и конкретной модели ключа HASP

Третья часть («Использование HASP API», стр. 115) призвана предоставить необходимые сведения о методе и стратегии защиты с помощью HASP API. В этом разделе вы найдете детальное описание всех функций HASP API. Информация данной части необходима для любого разработчика, который собирается использовать вызовы HASP внутри собственного кода.

Четвертая часть («Использование HASP в сети», стр. 219) призвана ознакомить разработчика с системой HASP4 Net и инструментарием этой системы. Сведения, сообщаемые в этой части, необходимы любому разработчику, который собирается использовать ключи HASP4 Net.

Пятая часть («Использование системы дистанционного перепрограммирования», стр. 287) призвана дать исчерпывающую информацию по утилитам и API, которые позволят вам обновлять память ключей HASP ваших клиентов удаленно.

С целью облегчить изучение системы HASP в конце большинства глав приводится раздел «Часто задаваемые вопросы». Также, для вашего удобства, в конце Руководства приводится глоссарий по основным терминам, используемым в данном документе.

В приложениях вы сможете найти информацию по техническим спецификациям, паролям для демонстрационных ключей HASP, информацию по устранению неполадок.

Часть 1
Общие сведения

В данном разделе приводится информация об основных концепциях системы защиты HASP, дающая общее представление о программных и аппаратных решениях, а также о том, как правильно устанавливать программное обеспечение.

В главе «Вступление» (стр. 3) рассказывается о преимуществах HASP, методах защиты информации. Здесь можно найти описание ключей HASP, поддерживаемых платформ и операционных систем.

В главе «Основные концепции» (стр. 13) разъясняются основные концепции системы HASP и описывается содержимое инсталляционных наборов HASP.

Глава «Установка HASP» (стр. 25) описывает программное обеспечение HASP, стадии защиты программного обеспечения, основные этапы установки системы HASP в различных операционных средах.

Вступление

Система HASP компании Aladdin представляет собой систему профессиональной защиты программного обеспечения. В данном разделе вы найдете описание системы защиты HASP, а также содержимого наборов HASP Developer Kit и HASP Starter Kit.

О HASP

HASP – это революционная система защиты программного обеспечения с помощью аппаратных средств защиты. Она позволяет защищать приложения от несанкционированного доступа и использования.

Защищаемое приложение запрашивает HASP в режиме реального времени. В случае получения положительного ответа от HASP, т.е. в случае правильной идентификации используемого алгоритма, приложение исполняется. В противном случае приложение может не загрузиться, загрузиться в демонстрационном режиме или закрыть доступ к некоторым своим возможностям.

Реализация такой защиты крайне проста, в то время как степень защиты – значительна. После реализации защиты вашего приложения оно будет исполняться только в случае подключения соответствующего ключа HASP.

Преимущества системы HASP

Широкие возможности разработчика

HASP предоставляет возможности по защите широкого спектра приложений, продуктов, включая защиту памяти, сетей, систем, работающих в режиме реального времени, на различных аппаратных платформах.

Максимальная простота использования

Все наши продукты максимально просты в изучении и использовании. Набор функций представлен в виде стандартного API. Это позволяет в минимально короткие сроки реализовывать систему защиты приложения.

Поддержка множества систем программирования

HASP имеет интерфейсы для различных компиляторов и языков программирования, что позволяет вести разработку для большинства платформ и операционных систем.

Поддержка различных ОС

На данный момент поддерживается: Windows 3.x, Windows 95, 98, ME, 2000, NT, XP, .Net, Linux, Mac OS 8.6, Mac OS 9.x, Mac OS X.

Кросс–платформенные решения

HASP (в версии USB) позволяет реализовывать независимые от платформы решения. Один и тот же ключ может использоваться для защиты приложений для Windows, MAC и Linux. Таким образом, вы экономите время на разработке интегрированной защиты, а также на поставке своих решений и логистике, а следовательно, и цене конечного продукта.

Высокая степень защиты

Заказной чип ASIC или передовой микроконтроллер

Все ключи HASP (кроме версии USB) построены на чипах ASIC (Application Specific Integrated Circuit). В чипе используется 2800 вентилей, он выполнен по технологии 1.2 микрона. ASIC разработан инженерами Aladdin и защищен от декомпиляции, что позволяет говорить о том, что наше аппаратное обеспечение практически невозможно взломать.

Модели USB уникальны в том, что они содержат микроконтроллер, поддерживающий тот же уровень защиты.

Передовые возможности шифрования HASP

Передовые возможности шифрования/дешифрования информации с помощью аппаратных средств HASP позволяют осуществлять тесную интеграцию аппаратных средств с защищаемым программным обеспечением. Интеллектуальные функции самого ключа позволяют критическим функциям защищаемого приложения быть доступными в зависимости от наличия корректного ключа (в противном случае эти функции будут недостотупны).

Любые данные, используемые приложением, могут быть зашифрованы в любом месте приложения. Шифруемая информация является функцией от данных, посланных на HASP, и уникального кода разработчика. Таким образом, эта система предоставляет куда большие возможности, чем простая периодическая проверка на наличие ключа.

Передовые алгоритмы защиты и алгоритмы защиты от декомпиляции

Приложения HASP используют собственные системы защиты кода и алгоритмов. В дополнение к этому, HASP использует самые передовые технологии защиты от декомпиляции. Специальные антихакерские возможности, применяемые в системе HASP, создают практически непреодолимые препятствия для взломщика.

Кодированные сообщения

Весь обмен данными между ключом и приложением зашифрован. Шифрование идет в режиме реального времени и случайным образом, что позволяет исключить возможность эмуляции присутствия ключа в системе.

Выгода для покупателя

Использование HASP выгодно как вам, так и вашим покупателям. Ниже приведено несколько причин, почему защита приложений выгодна вашим клиентам.

Эффективность с точки зрения соотношения эффективность/затраты

Вследствие того, что защита с помощью HASP поднимает ваши продажи, вы можете выделять больше денег на разработку и поддержку вашей продукции. В свою очередь, покупатели выигрывают, приобретая более совершенное ПО. Как следствие - разработка идет быстрее, а техническая поддержка имеет более высокое качество.

Защита лицензионного соглашения

Защита с помощью HASP позволяет лучше соблюдать лицензионное соглашение. HASP – наиболее незаметное и в то же время эффективное средство для обеспечения соблюдения соглашения. Это значит, что покупатель вашего ПО не будет вынужден сам заботиться о соблюдении лицензионного соглашения (например, отпадает необходимость в контроле за правильным использованием ПО сотрудниками фирмы).

Защита инвестиций добросовестных пользователей

HASP заботится о добросовестных клиентах, защищая их от порочной практики, когда недобросовестные пользователи используют неоплаченное ПО, распыляя, таким образом, ваши технические ресурсы.

Поддержка и сервис по всему миру

Несколько производственных площадок

Система HASP производится на нескольких заводах, расположенных на трех континентах, что позволяет доставлять систему без задержек и иметь возможности для расширения производства в случае необходимости.

Поддержка пользователей в 40 странах

Aladdin имеет офисы в восьми странах мира и более ста дистрибьюторов. Поддержка наших решений может быть оказана в любое время в любой точке земного шара.

Консультационные услуги

Для получения самой полной информации о HASP, а также в случае необходимости проведения тренингов вы можете обратиться к консультантам нашей международной компании. Они могут предложить вам следующие тренинги:

- Интеграция HASP и дистрибуция в вашей компании.
- Анализ наилучшей стратегии защиты ваших приложений
- Помощь в создании алгоритмов и полный цикл работ по созданию защиты ваших приложений с помощью HASP.

Ключи защиты HASP

Существуют ключи HASP в виде различных моделей и для различных портов .

Таблица 1.1 Модели HASP

Модель HASP	Размер памяти (байтов)	Возможности системы FAS	Возможности	Тип ключей
HASP4 std.	Нет	Нет	Шифрование/ дешифрование	Параллельный, USB
HASP4 M1	112	16 приложений	Шифрование/ дешифрование, HASP ID	Параллельный, USB
HASP4 M4	496	112 приложений	Шифрование/ дешифрование, HASP ID	Параллельный, USB, PC-Card
HASP4 Time	496 + 16	8 приложений, дата окончания	Шифрование/ дешифрование, HASP ID, часы реального времени	Параллельный, USB
HASP4 Net 5	496	112 приложений на 5 станциях	Шифрование/ дешифрование, HASP ID, доступ к сети	Параллельный, USB
HASP4 Net 10	496	112 приложений на 10 станциях	Шифрование/ дешифрование, HASP ID, доступ к сети	Параллельный, USB
HASP4 Net 20	496	112 приложений на 10 станциях	Шифрование/ дешифрование, HASP ID, доступ к сети	Параллельный, USB

Модель HASP	Размер памяти (байтов)	Возможности системы FAS	Возможности	Тип ключей
HASP4 Net 50	496	112 приложений на 50 станциях	Шифрование/ дешифрование, HASP ID, доступ к сети	Параллельный, USB
HASP4 Net 100	496	112 приложений на 100 станциях	Шифрование/ дешифрование, HASP ID, доступ к сети	Параллельный, USB
HASP4 Net U	496	112 приложений на любом количестве станций	Шифрование/ дешифрование, HASP ID, доступ к сети	Параллельный, USB

Типы ключей HASP4 для различных портов

Ключи HASP4 существуют в различных версиях для параллельного порта, USB, а также в виде PC-Card. Функциональность ключей идентична.

- Ключи HASP4 для параллельного порта соединяются с параллельным портом компьютера. Такие ключи могут быть использованы только с PC.
- Ключи HASP4 USB – это ключи, которые соединяются с портом USB и могут быть использованы на платформах PC и Mac.
- Ключи HASP4 PC-Card – это карты защиты, которые используются в слотах PCMCIA, присутствующих практически на любом ноутбуке.

Модели ключей HASP4

HASP4 Standard – минимальная цена, максимальная функциональность

HASP4 Std. – это самое недорогое из предлагаемых решений для защиты ПО. Данная модель HASP4 использует все наработки HASP и предлагает очень качественное, но недорогое решение для ваших нужд по защите информации.

HASP4 Std. предлагается в версиях для параллельного порта и USB.

HASP4 M1 и HASP4 M4 – наиболее универсальные и надежные решения

HASP4 M1 и HASP4 M4 соединяют в себе высокий уровень безопасности на основе алгоритмов шифрования и гибкость использования 496 байтов памяти. Каждый ключ имеет свой уникальный ID-номер. Память ключа позволяет вам осуществлять гибкую маркетинговую политику благодаря возможности применения демонстрационных режимов, включения дополнительных опций, режимов выдачи лицензий на работу 112 приложений одновременно с использованием одного и того же ключа.

HASP4 M1 поставляется в версиях для параллельного порта и для USB.

HASP4 M4 поставляется в версиях для параллельного порта, USB и в виде PC-Card.

HASP4 Time – защита приложений с использованием часов реального времени

HASP4 Time содержит в себе часы реального времени, показывающие время и дату. HASP4 Time позволяет вам идентифицировать время использования приложения вашими клиентами. HASP4 Time создан на основе HASP4 M4. HASP4 Time содержит 512 байт памяти и уникальный ID-номер для каждого ключа. Используя HASP4 Time, вы можете контролировать до восьми модулей ПО плюс различные приложения, использовать стратегии предоставления ПО в аренду, а также распространять демо-версии, срок действия которых истекает в определенный момент.

HASP4 Time поставляется в версиях для параллельного порта и для USB.

HASP4 Net – лицензирование сетевого ПО

HASP4 Net – замечательное решение для защиты сетевых приложений. Вы можете подсоединить один ключ HASP4 Net к любой сетевой станции, чтобы защитить приложения и одновременно ограничить количество используемых копий по сети. HASP4 Net поддерживает всю функциональность HASP4 M4.

Комплект разработчика HASP

Комплект разработчика HASP содержит все необходимое для того, чтобы вы смогли оценить степень защиты HASP. Набор включает в себя:

Программное обеспечение

ПО HASP поставляется на отдельном компакт-диске.

Аппаратное обеспечение

Комплект разработчика HASP поставляется с демонстрационным ключом. Демонстрационный ключ может быть HASP4 M1, HASP4 Time или HASP4 Net (в зависимости от того, какой ключ вы выбрали при заказе).

Демонстрационный ключ в этом наборе может быть использован только в целях ознакомления. Если вы заказываете ключи, Aladdin присваивает вам уникальный код разработчика, который отличает ваши ключи от ключей других разработчиков.

Документация

Ваш комплект разработчика поставляется с одной копией Руководства программиста HASP4.

Стартовый комплект HASP

Стартовый комплект HASP похож на комплект разработчика HASP, но он содержит 5 уникальных ключей (либо 2 ключа в случае HASP4 Net). Ваши ключи содержат уникальные пароли, которые знаете только вы.

Ключи, поставляемые в этом комплекте, могут быть использованы не только в целях ознакомления, но и для подготовки дистрибутивов приложений для конечных пользователей вашего продукта.

Со стартовым набором HASP вы подготовлены к выпуску тиража приложения. Просто защитите свое приложение и закажите то количество ключей, которое вам необходимо.

Основные концепции

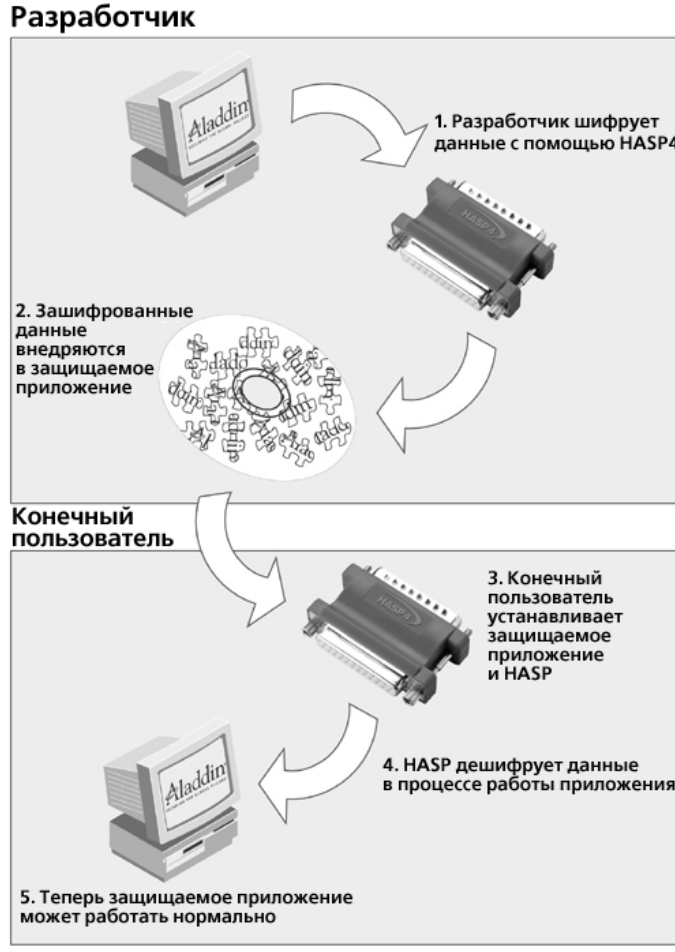
Для наиболее полной и эффективной реализации защитных стратегий HASP следует ознакомиться с базовыми принципами и терминами, описываемыми в этой главе.

В случае, если вы собираетесь использовать HASP4 Net, рекомендуется ознакомиться с разделом «Основные концепции HASP4 Net» (стр. 221), чтобы получить представление о терминах и понятиях, специфичных для HASP4 Net.

Как работает HASP?

Во время работы приложение запрашивает HASP, подключенный к компьютеру. Если HASP возвращает ответ с использованием корректного алгоритма, то приложение исполняется. В противном случае приложение не загружается, либо может переключиться в демонстрационный режим, закрыть свои основные возможности и т.п.

Рисунок 2.1. Механизм защиты HASP



Идентификация ключа HASP

Защита основывается на присутствии корректного ключа HASP в системе.

Ключ HASP содержит информацию, специфичную для вашей компании. Таким образом, всегда видно, принадлежит ли ключ вам или другой сторонней организации.

Присутствие корректного ключа может быть проверено следующими методами:

- С использованием механизма шифрования, основывающемся на присутствии ключа
- Проверка специфичного для ключа ID-номера
- С использованием функций памяти ключа

Использование механизма шифрования

Реализуя механизмы защиты, вы проверяете наличие ключа HASP. Система HASP производит эту проверку методом шифрования/дешифрования данных с использованием самого ключа.

Для того чтобы использовать эту возможность, вам необходимо сделать некоторые подготовительные шаги. Для того чтобы расшифровать данные, вы должны послать уже зашифрованные данные ключу. Данные расшифровываются HASP с использованием сервиса DecodeData. Далее вы сможете проверить, корректны ли расшифрованные данные. Если они корректны, значит, ключ присутствует в системе.

Дешифрованные данные могут быть проверены с использованием данных вашего приложения.

Зашифрованные данные являются функцией уникального кода разработчика и ваших собственных данных. Таким образом, шифрование одной и той же строки ключами разных разработчиков приведет к разным результатам.

Вы можете шифровать данные с использованием HASP Edit и HASP API. Для получения более полной информации обратитесь к разделам «Доступ к ключам с использованием HASP Edit» на стр. 71 или «Защита при помощи HASP API» на стр. 117.

Рис.2.2. Дешифрование данных для проверки ключа HASP

Информация для разработчика

При заказе ключей в компании Aladdin вы получаете ключи со встроенной информацией о вашей компании. Эта информация используется для того, чтобы отличать ваши ключи от ключей других производителей.

Код разработчика

Код разработчика – это уникальный код, который компания Aladdin присваивает каждому разработчику ПО. Этот код прожигается на чипе ASIC, чтобы не допустить декомпиляции. Таким образом, повышается степень защищенности приложений.

При заказе дополнительных партий ключей вы используете тот же код разработчика, что и ранее. Соответствие этого кода может быть проверено сверкой строки из пяти или семи символов, напечатанных на каждом ключе HASP.

Пароли HASP

Пароли HASP – это два целых числа, присвоенные каждому разработчику. Пароль базируется на вашем коде разработчика. Такой пароль позволяет быть уверенным, что только вы сможете получить доступ к функциям HASP.

Позаботьтесь о сохранности ваших паролей, так как только зная пароль, вы сможете получать доступ к HASP, защищать приложения и получать доступ к утилитам HASP!

Проверка ID-номера HASP

Каждый ключ HASP имеет свой уникальный ID-номер. Защищаемое приложение может проверять данный номер.

Ключи HASP со встроенными ID-номерами помогают вам различать пользователей вашего ПО. Проверяя данный номер, вы всегда сможете точно сказать, присутствует ли конкретный ключ в системе или нет.

Вы можете получить этот номер, используя программу HASP Edit. Для получения более полной информации обратитесь к разделу «Доступ к ключам с использованием HASP Edit» стр. 71.

Вы не можете заказать ключи с предопределенными номерами. Они присваиваются ключам случайным образом в момент производства.

Использование опций памяти HASP

Все ключи HASP (кроме HASP Standard) содержат внутреннюю память. Вы можете использовать эту память, чтобы:

- Контролировать доступ к различным модулям ПО или различным приложениям.
- Присваивать уникальный ID-номер каждому пользователю.
- Распространять демонстрационные версии, которые могут быть активированы ограниченное количество раз.
- Сохранять пароли, код программы, переменные или любые другие данные.

Для получения более полной информации о различных моделях ключей и доступной памяти см. Таблицу 1.1 (стр. 8).

Вы можете редактировать память ключа с использованием программы HASP Edit. Для получения более полной информации обратитесь к разделу «Доступ к ключам с использованием HASP Edit» стр. 71

Методы защиты HASP

HASP позволяет применять два различных метода защиты:

- HASP Envelope
- HASP API

HASP Envelope

HASP Envelope – это базовый метод защиты. Данный метод создает «защитный щит» вокруг приложения. HASP Envelope зашифровывает ваши файлы и встраивает проверки наличия HASP, а также методы защиты от декомпиляции. После защиты вашего приложения таким методом оно не сможет быть запущено, если корректный ключ не присутствует в системе.

Использование данного метода не предполагает изменение исходного кода вашего приложения. Именно поэтому это наиболее быстрый метод защиты. В то же время, этот метод защиты очень надежен. Приложение, защищенное подобным образом, практически невозможно взломать.

Для получения более полной информации обратитесь к разделу «Защита с помощью HASP Envelope» стр. 43.

HASP API

Если вы собираетесь защищать разрабатываемое вами приложение, вы можете использовать HASP API. Так как API одновременно зашифрован и защищен, этот метод также очень надежен.

Используйте API для вставки вызовов HASP в вашем приложении. Используя API, вы всегда сможете вставить код, проверяющий присутствие ключа, в любом месте вашего приложения, а также решить, что делать в случае отсутствия ключа в каждом конкретном случае. В дополнение к этому вы сможете использовать память ключа для сохранения там наиболее важных данных.

Для того чтобы использовать API, вам будет необходимо вносить изменение в исходный код программы.

Какой метод использовать?

Вы можете использовать любой из двух методов или комбинировать их.

Используйте метод HASP Envelope для быстрой защиты приложения без модификации исходного кода.

Используйте API в том случае, если вы хотите модернизировать ваши методы защиты.

Оба этих метода обеспечивают высочайшую степень защиты. Для получения еще более высокой степени защищенности мы предлагаем использовать оба этих метода одновременно. Каждый метод имеет свои плюсы, поэтому они будут дополнять друг друга.

Встройте вызовы HASP API в ваше приложение, скомпилируйте его, установите связь объектных файлов HASP (или библиотек dll) с вашим приложением, а затем используйте метод защиты HASP Envelope для ваших исполняемых файлов.

Система полного управления доступом (FAS – Full Authorization System)

Система полного управления доступом позволяет защищать несколько приложений с использованием лишь одного ключа HASP. Эта система позволяет настроить параметры и условия, при которых будет исполняться каждое приложение.

Опции FAS

С помощью FAS вы сможете:

- Установить количество запусков приложения (HASP4 M1, HASP4 M4 и HASP4 Net).

  Эта опция полезна для распространения демо-версий приложений.

- Установить дату окончания действия лицензии (HASP4 Time).

  Эта опция полезна при передачи приложений в аренду.

- Установить количество станций, на которых приложение сможет исполняться (HASP4 Net).

FAS – это опциональная возможность Envelope в случае защиты компьютера локальным ключом. В случае же сетевой защиты FAS обязательна. FAS является интегральной частью защиты HASP4 Net вне зависимости от того, используете ли вы Envelope или API.

Как работает FAS?

Приложение, защищаемое HASP с использованием FAS, делает несколько проверок.

- Приложение проверяет, подсоединен ли корректный ключ к системе.
- Если корректный ключ найден, проверяется память ключа HASP для того, чтобы удостовериться, что приложение имеет полномочия на запуск.

В случае, когда приложение авторизовано, производится еще несколько проверок (в зависимости от используемой модели ключа):

- В случае использования HASP4 M1 и HASP4 M4 производится проверка на общее количество запусков приложения. Каждый раз, когда приложение запускается, счетчик запусков этого приложения уменьшается на единицу. В тот момент, когда счетчик становится равным нулю, приложение не запускается, выдавая соответствующий код ошибки.
- В случае использования HASP4 Time производится проверка даты окончания лицензии (сравнивается текущая дата и дата окончания лицензии). В случае, если дата окончания лицензии прошла, приложение не запускается, выдавая соответствующий код ошибки.
- В случае использования сетевых ключей для получения наиболее полной информации по интеграции сетевой защиты и FAS следует обратиться к разделу «Как работает HASP4 Net?» стр. 222.

Часто задаваемые вопросы

Вопрос: Могу ли я менять пароль HASP?

Ответ: Нет. Это обусловлено требованиями защиты от декомпиляции. Пароль прошит в аппаратный контроллер ключа, зашифрован и не может быть изменен.

Вопрос: Могу ли я предоставлять моим покупателям ключи HASP с уникальными кодами и паролями?

Ответ: Да. Для этого мы рекомендуем использовать ключи HASP с памятью, каждый из которых содержит уникальный идентификационный номер для каждого ключа. В дополнение к этому, вы можете записать в память ключа информацию, уникальную для каждого покупателя.

Вопрос: Может ли возникнуть такая ситуация, что другой разработчик будет иметь те же пароли, что и я?

Ответ: Нет. Каждый разработчик получает уникальный набор паролей.

Вопрос: Что происходит, когда разряжается батарея ключа HASP4 Time?

Ответ: Батарея рассчитана на 3-5 лет непрерывной работы. С точки зрения лицензирования ПО это большой срок. Когда батарея у HASP4 Time разрядится, этот ключ будет вести себя так же, как и HASP4 M4. Приложения, которые используют методы защиты Envelope и FAS, перестанут запускаться. Время жизни батареи увеличивается, если ключ подключен к работающему компьютеру.

Вопрос: Могут ли ключи HASP использоваться в сети?

Ответ: Да. Имеются две возможности использования подобных ключей в сети. Во-первых, можно подключить HASP к каждой станции, на которой запускаются защищаемые приложения. Во-вторых (это наиболее эффективный метод) можно защищать сетевые приложения ключами HASP4 Net – специальным решением для защиты сетевых приложений.

HASP4 Net позволяет вам защищать несколько станций с использованием лишь одного ключа. В случае, если вы собираетесь использовать HASP4 Net, рекомендуется ознакомиться с разделом «Основные концепции HASP4 Net».

Вопрос: Могут ли несколько ключей HASP быть соединены между собой?

Ответ: Да. Возможно каскадное соединение ключей.

Модели USB являются оконечными устройствами, поэтому они не могут быть соединены друг с другом (данная концепция неприменима к USB).

Вопрос: Как насчет ключей, созданных другими производителями?

Ответ: Ключи HASP полностью прозрачны. Вы можете соединять их с ключами других производителей, при этом порядок соединения не имеет значения.

Вопрос: Можно ли защищать ключами HASP приложения или данные, созданные с использованием таких инструментов, как Lotus или AutoLisp?

Ответ: Да. Существует специальная система защиты данных. Она называется DataHASP. Эта система защищает файлы методом шифрования содержимого этих файлов. Только авторизованные приложения смогут получить доступ к таким файлам. DataHASP может использоваться с приложениями DOS или Win32. Она работает с любыми ключами HASP.

Вопрос: В чем преимущества использования ключей HASP с памятью?

Ответ: Есть несколько преимуществ с точки зрения защиты. Память необходима для:

- Распространения демонстрационных версий приложений.
- Защиты нескольких приложений с помощью одного и того же ключа.
- Для сохранения данных (тем самым повышается степень защищенности приложений).

Вопрос: Каким образом я смогу обновлять память ключа HASP после поставки приложения покупателю?

Ответ: Для этого существует система удаленного перепрограммирования HASP (Remote Update System – RUS). Вы можете обновлять память ключа через телефон, электронную почту или по факсу. Таким образом, вы сможете обновить счетчики запусков приложения или информацию, необходимую для запуска старых или новых приложений.

Для получения более полной информации, свяжитесь с представителем Aladdin в вашем регионе.

Установка HASP

Эта глава посвящена описанию программного обеспечения HASP, основных стадий защиты приложений, а также основных процедур установки для различных операционных систем.

Программное обеспечение HASP

Программное обеспечение HASP состоит из следующих основных компонентов:

- Демоны и драйверы для доступа к ключам HASP.
- HASP Edit для подготовки ключей HASP к распространению.
- Aladdin Diagnostic для помощи конечному пользователю.
- HASP API и HASP Envelope для интегрирования защиты приложений.
- Менеджер лицензий HASP License Manager (далее по тексту – HASP LM) и Aladdin Monitor для работы с сетевыми лицензиями.

Демоны и драйверы устройств для доступа к ключам HASP

Демоны и драйверы для доступа к ключам HASP служат мостом между ключом HASP и защищаемым приложением. И вы, и ваши покупатели должны установить драйверы для того, чтобы ваши приложения могли нормально функционировать.

Демоны и драйверы для доступа к ключам HASP разработаны для следующих операционных систем: Windows 95/98/ME, Windows NT/2000/XP, Mac OS 8.6, Mac OS 9.x, Mac OS X, Linux.

Вы можете интегрировать установку драйверов в инсталляционную процедуру вашего приложения.

Доступ к ключам HASP с помощью HASP Edit

HASP Edit – это утилита для доступа к HASP и редактирования памяти ключей.

Aladdin Diagnostic для помощи конечному пользователю

Утилита Aladdin Diagnostic позволяет вашим покупателям собирать информацию об их системах и ключах HASP.

Защита приложений

HASP Envelope

HASP Envelope – это инструмент, который позволяет защищать ваше приложение без модификации исходного кода.

Вы можете использовать HASP Envelope при помощи графического интерфейса или командной строки. Для получения более полной информации обратитесь к разделу «Защита с помощью HASP Envelope» на стр. 43.

HASP API

HASP API позволяет вам защищать ваши приложения путем включения в их исходный код вызовов функций HASP.

Корневой каталог диска HASP имеет подкаталоги в соответствии с поддерживаемыми ОС, программными языками и компиляторами. Каталог для каждого компилятора содержит следующие файлы:

- Библиотеки, которые должны быть связаны с приложением.
- Демонстрационное приложение, показывающее, как использовать API со всеми ключами HASP.

Для получения более полной информации обратитесь к разделу «Защита при помощи HASP API» на стр. 117.

HASP Demo

Это утилиты, которые позволяют вам тестировать функциональность сервисов HASP. Для получения более полной информации обратитесь к разделу «Тестовая утилита HASP» на стр. 129.

Предоставление лицензий в сети

Для получения полной информации рекомендуется ознакомиться с разделом «Основные концепции HASP4 Net» на стр. 221.

HASP LM

Это утилита, которая осуществляет связь между защищаемым приложением и ключом HASP4 Net.

Aladdin Monitor

Утилита, позволяющая контролировать использование защищенных приложений, а также HASP LM по сети.

Защита приложений – краткое описание

Ниже приводится краткое описание шагов по защите приложений с использованием HASP.

Шаг 1: Инсталляция драйвера или сервиса HASP

Позволяет вам получить доступ к ключу. Самые новые драйверы доступны по адресу: www.eAladdin.com/support/hasp/vendor.asp

Информация по инсталляции представлена ниже.

Шаг 2: Использование утилиты HASP Edit

Позволяет вам выбрать методы защиты.

Для получения более полной информации обратитесь к разделу «Доступ к ключам с использованием HASP Edit» на стр. 71.

Шаг 3: Использование HASP API

Позволяет вам встроить вызовы HASP в ваше приложение.

Для получения более полной информации обратитесь к разделу «Защита при помощи HASP API» на стр. 117.

Шаг 4: Использование утилиты HASP Envelope

Позволяет вам зашифровать исполняемые файлы.

Для получения более полной информации обратитесь к разделу «Защита с помощью HASP Envelope» на стр. 43.

Шаг 5. Использование утилиты HASP Edit

Позволяет вам подготовить ваши ключи к продаже совместно с защищаемым продуктом.

Для получения более полной информации обратитесь к разделу «Доступ к ключам с использованием HASP Edit» на стр. 71.

Установка HASP в среде Windows

Установка программного обеспечения HASP

Вставьте компакт-диск с программным обеспечением HASP в дисковод. Мастер установки запустится автоматически. В случае, если мастер не запустился, запустите его вручную. Для этого запустите файл setup.exe из каталога Setup.

Далее следуйте инструкциям на экране.

Установка драйвера устройства HASP

Драйвер устройства будет установлен автоматически в момент общей установки. Мы также предоставляем альтернативные методы установки драйвера:

Таблица 3.1. Инструменты для установки драйвера устройства (Windows)

Приложение	Операционная система
Hinstall.exe	Win32, командная строка
HDD32.EXE	Win32, графический интерфейс
HDD16.EXE	Win16, графический интерфейс

Эти приложения расположены в подкаталоге Drivers вашего компакт-диска HASP. Приложения автоматически распознают версию ОС и инсталлируют соответствующий драйвер в корректное место на вашем жестком диске.

Для получения полной информации обратитесь к файлу справки *hdd.hlp*.

Для установки драйверов в среде Windows NT/2000/XP вы должны обладать полномочиями администратора системы.

В ОС Windows 95/98/ME драйверы загружаются динамически (если ранее не были установлены).

После установки драйверов приложения, защищаемые версией HASP более ранней, чем 6.1, требуют перезагрузки компьютера. После перезагрузки драйверы будут загружены автоматически.

Если вы устанавливаете драйверы более поздней версии, чем те, что были установлены ранее, вам будет необходимо перезагрузить компьютер. После перезагрузки новые драйверы запускаются автоматически.

Настройка установки драйверов HASP

Вы можете загрузить драйверы одним из следующих способов:

- Запустите файл *Hinstall.exe* или *HDD32.EXE/HDD16.EXE*. Вы можете написать командный файл, который выполняет эти действия автоматически для того, чтобы запускать его на всех требуемых станциях.
- Установите драйвер HASP при установке вашего приложения путем создания вашей собственной инсталляционной процедуры.

В каталоге *Drivers/drvapi* вы можете найти интерфейсы к различным компиляторам, включая интерфейс к программе Install Shield. Каждый подкаталог содержит демонстрационную версию программы инсталляции драйверов. Для получения полной информации по утилите *hinstall* и API-инсталляции драйверов обратитесь к файлу справки hdd.hlp.

Утилита Hinstall

Hinstall.exe – Это приложение Win32, которое инсталлирует драйверы HASP в ОС Windows 95/98/ME и Windows NT/2000/XP.

Инсталляция драйверов:

- Введите команду **hinstall –i** в командной строке.

  На экране появится окно, информирующее о том, что драйвер устройства HASP установлен успешно.

Удаление драйвера HASP:

- Введите команду **hinstall –r** в командной строке.

  На экране появится окно, информирующее о том, что драйвер устройства HASP удален успешно.

Обновление драйвера HASP:

- Установите новый драйвер, следуя инструкциям, описанным выше.

Установка HASP в среде Mac

ПО HASP для Mac включает в себя демон и драйвер, HASP API, утилиту HASP Edit и HASP LM. Файлы располагаются в архиве sit в каталоге MAC HASP CD.

Установка демона HASP в среде Mac OS X

Демон HASP (*aksusbd)* позволяет системе Mac OS X и приложениям получать доступ к ключу HASP.

Этот демон (*aksusbd*) необходим и вам, и покупателям ваших приложений для корректной работы ключа HASP. Для установки этого демона вы должны запустить скрипт *dinst*.

Чтобы установить демон:

1. Зарегистрируйтесь на машине как root или как другой пользователь, имеющий полномочия администратора.
2. Откройте сессию терминала под OS X. Чтобы выполнить это, выберите **Applications** из меню **Go,** откройте каталог Utilities и выберите **Terminal**.
3. Перейдите в каталог демонов.
4. Запустите скрипт, напечатав *./dinst*. В случае если вы зашли не как root, запустите скрипт, напечатав *sudo ./dinst*.

   Демон будет запускаться автоматически каждый раз при старте компьютера.

После того, как демон был поставлен и загружен, ваша система будет распознавать ключи HASP, присоединенные к порту USB.

Вы можете сконфигурировать демон, используя следующие ключи командной строки:

Таблица 3.2. Ключи командной строки для *aksusbd* (Mac)

Ключ	Описание
-v	Распечатывает номер версии в формате xx.xx
-u <umask>	Определяет биты разрешения доступа для специального файла сокетов. Значение по умолчанию – 666.
-l <value>	Позволяет выбрать тип сообщений диагностики. Возможные значения: 0 – только ошибки 1 – нормальный (по умолчанию) 2 – подробный 3 – очень подробный
-h	Распечатывает помощь

Чтобы сконфигурировать демон:

1. Отредактируйте файл *Aladdin*, на который ссылается скрипт *dinst.*
2. Введите необходимые ключи в строку *aksusbd*, которая запускает демон.

Установка драйвера HASP для MAC OS 8.6 и OS 9.x

Драйвер HASP позволяет системе MAC OS 8.6 и OS 9.x и приложениям получать доступ к ключу HASP.

Этот драйвер необходим и вам, и покупателям ваших приложений для корректной работы ключа HASP.

Чтобы установить драйвер:

1. Скопируйте файлы *MacHasp4Shim* и *MacHaspUsbDD* в подкаталог **Extensions** каталога **System Folder.**

Если файлы с такими именами уже присутствуют в каталоге, вам будет необходимо сначала переместить их оттуда, поскольку файлы в этом каталоге не могут быть перезаписаны.

2. Перезагрузите систему.

Теперь ваша система будет распознавать ключи HASP, подключенные к порту USB, корректно.

Установка HASP Edit в среде Mac

Чтобы установить HASP Edit:

1. Откройте архив sit на HASP CD.
2. Скопируйте файлы приложения HASP Edit на ваш компьютер.

Для получения более полной информации обратитесь к разделу «Доступ к ключам с использованием HASP Edit» на стр. 71.

Установка утилит и поддержки HASP4 Net

Для получения полной информации рекомендуется ознакомиться с разделом «Основные концепции HASP4 Net» на стр. 221.

Установка HASP в среде Linux

Программное обеспечение HASP под Linux включает в себя следующие компоненты:

- Драйверы Kernel для различных версий ядра и сервисов.
- Утилиты для опроса версии драйвера и показа доступных параллельных портов.
- Приложение HASP Demo и его исходные тексты.
- Библиотеку HASP.

Все вышеперечисленное содержится в архиве *hasplinux101.tgz* в каталоге Linux компакт-диска HASP.

Для получения более полной информации обратитесь к текстовым файлам в архиве.

Открытие архива

Чтобы открыть архив:

1. Создайте новый каталог.
2. Раскройте архив, используя следующую команду:

   Tar -xzf [path/]hasplinux101.tgz

Инсталляция демона и драйвера Kernel

Чтобы получить доступ к ключу HASP, вы должны загрузить драйвер aksparlnx.o и демон aksusbd.

Все операции должны делаться с полномочиями root.

Инсталляция драйвера Kernel (aksparlnx.o)

Чтобы установить драйвер:

- Создайте узел устройства
- Инициализируйте систему parport
- Загрузите драйвер

Создание узла устройства

Чтобы получить доступ к драйверу, создайте узел устройства /dev/Hardlock.

Этот узел должен иметь тот же главный номер, что и номер, используемый для загрузки драйвера (командная строка major=xxx для insmod или 42 по умолчанию).

1. Для создания узла используйте команду:

```
mknod /dev/Hardlock c 42 0
```

1. Откройте доступ к узлу для всех:

```
chmod 666 /dev/Hardlock
```

В случае, если доступ к устройствам Aladdin должен быть ограничен специальной группой пользователей (например, группа Aladdin):

```
chgrp aladdin /dev/Hardlock

chmod 666 /dev/Hardlock
```

Инициализация системы Parport

Драйвер *aksparlnx* использует драйвер Linux parport для доступа к параллельному порту без ограничения доступа к этому порту других пользователей (например, для доступа к принтеру, zip-дисководу).

Чтобы инициализировать систему parport перед запуском драйвера *aksparlnx*, используйте команду:

```
modprobe parport_pc
```

Драйвер parport сообщит системе и в регистрационный файл, что он присутствует в системе. В этом случае драйвер aksparlnx получит возможность доступа к ключам на любом из параллельных портов по умолчанию.

Загрузка драйвера

Чтобы загрузить драйвер *aksparlnx*, используйте команду:

```
insmod <path_to_driver>/aksparlnx.o
```

Сообщение статуса будет сгенерировано для логирования. Вы будете проинформированы о том, что инсталляция прошла успешно (либо нет, в случае ошибки).

Загрузка драйвера (modprobe и insmod) должна быть вставлена в загрузочный скрипт для того, чтобы драйверы загружались всякий раз при загрузке системы.

Опции *aksparlnx.o*

Таблица 3.3. Ключи командной строки драйвера *aksparlnx.o* (Linux)

Ключ	Описание
-major= <number>	Драйвер использует значение 42 по умолчанию. В случае необходимости вы можете изменить этот номер с помощью описываемого ключа. Также подобное изменение возможно с использованием команды: `mknod /dev/Hardlock c <number> 0`
-loglevel= <value>	Выберите тип диагностики. Возможные значения: 0 - только ошибки 1 - нормальный уровень (по умолчанию) 2 - подробный 3 - очень подробный Все сообщения регистрируются в syslog с приоритетом kern.info (и kern.debug). Вы можете просмотреть /etc/syslog.conf для того, чтобы определить место, куда будут помещаться все сообщения. Обычно сообщения помещаются в файл /var/log/messages.
-timeout= <value>	Определяет максимальный промежуток времени, который драйвер проводит в ожидании предоставления доступа к порту драйвером parport. Время измеряется в сотых долях секунды. Значение по умолчанию 100 (1 секунда). После истечения заданного промежутка времени запрос будет отклонен с типом ошибки PORT_BUSY.
-hlportaddress= <address>	Если у вас в системе имеется порт, который не был замечен драйвером parport (и, таким образом, невидимый для *aksparlnx.o*), вы сможете указать адрес этого порта вручную.  Используйте этот ключ с осторожностью. Если вы зададите неправильный адрес порта, система может «зависнуть».

Инсталляция демона Aladdin

Открытие доступа к ключам USB

Для того чтобы открыть доступ к ключам USB, *usbdevfs* должен быть смонтирован на /proc/bus/usb. В более новых дистрибутивах он монтируется автоматически (например, в SuSe 7.0).

Чтобы смонтировать usbdevfs вручную, исполните команду:

```
mount -t usbdevfs none /proc/bus/usb
```

Открытие доступа к ключам для параллельного порта

Для того чтобы получить доступ к ключам для параллельного порта, драйвер kernel *aksparlnx* должен быть проинсталлирован до того, как будет запущен *aksusbd*.

Загрузка демона

Загрузите демон:

```
<path>/aksusbd
```

Демон запустится и переведет себя в фоновый режим. Сообщение статуса будет сгенерировано для регистрации. Вы будете проинформированы о том, что инсталляция прошла успешно (либо нет, в случае ошибки).

Демон сообщает свою версию, версию API, используемую для USB, и версию API драйвера kernel (для ключей для параллельного порта).

В случае, если драйвер kernel оказывается недоступным в момент, когда запускается *aksusbd,* ключи параллельного порта не будут доступны, но ключи USB доступны в любом случае. Регистрационный файл системы отражает статус ключей.

В случае, если */proc/bus/usb* не смонтирован в момент запуска *aksusbd*, ключи USB оказываются недоступными.

Лучше всего запускать демон в момент загрузки системы из скрипта, расположенного по адресу */etc/rc.d/init.d* или */etc/init.d* (в зависимости от версии дистрибутива Linux).

Настройки *aksusbd*

Таблица 3.4 Ключи командной строки для *aksusbd* (Linux)

Ключ	Описание
-v	Распечатывает десятичный номер версии в формате xx.xx
-u <umask>	Определяет биты разрешения доступа для специального файла сокетов. Значение по умолчанию – 666
-l <value>	Позволяет выбрать тип сообщений диагностики. Возможные значения: 0 – только ошибки 1 –нормальный (по умолчанию) 2 –подробный 3 –очень подробный Все сообщения регистрируются в syslog с приоритетом kern.info (и kern.debug). Вы можете просмотреть /etc/syslog.conf для того, чтобы определить место, куда будут помещаться все сообщения. Обычно сообщения помещаются в файл /var/log/messages
-h	Распечатывает помощь

Часть 2
Использование инструментария HASP

В этой части описаны утилиты HASP Envelope, HASP Edit и Aladdin Diagnostic.

Глава «Защита с помощью HASP Envelope» на стр. 43 поясняет метод использования HASP Envelope, который позволяет наиболее быстро защищать ваши приложения и шифровать данные.

Глава «Доступ к ключам с использованием HASP Edit» на стр. 71 поясняет методы использования HASP Edit для Win32 и для Mac. С помощью этой утилиты вы можете получать доступ к ключам HASP.

В главе «Поддержка конечных пользователей» на стр. 109 содержится информация о методах диагностирования проблем конечных пользователей с использованием утилиты Aladdin Diagnostics, которая позволяет пользователям собирать информацию об их системе и ключах HASP.

Защита с помощью HASP Envelope

HASP Envelope представляет собой наиболее простой и эффективный способ защиты приложений. Защита HASP Envelope проста, так как не требует модификации исходного кода приложения.

HASP Envelope может использоваться в форме командной строки (DOS, Win16, Win32), а также в виде графического интерфейса Win32.

Интерфейс HASP Envelope позволяет легко производить следующие операции:

- Защищать приложения
- Защищать файлы данных Win32
- Сохранять параметры защиты FAS в памяти HASP

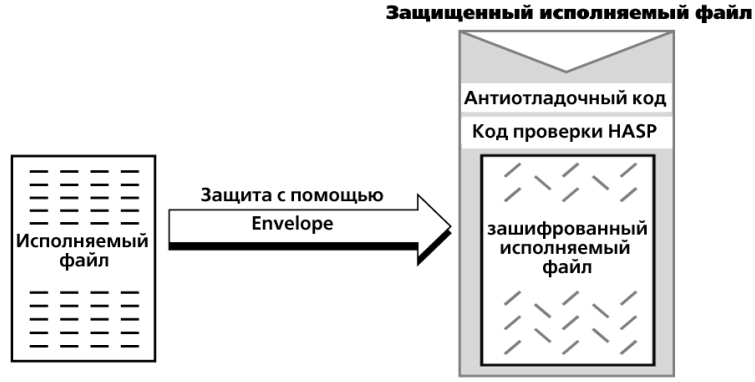
В случае если вы собираетесь использовать HASP Envelope совместно с защитой API, сначала создайте защиту API, и лишь потом – HASP Envelope.

Общие сведения о HASP Envelope

HASP Envelope добавляет защитную оболочку вокруг приложений и библиотек DOS, Windows и Win32.

Подобная защита позволяет шифровать файлы, при этом современные методы защиты от декомпиляции повышают общий уровень защиты.

Рисунок 5.1. Защита HASP Envelope



Перед тем как создать защиту HASP Envelope, выполните следующие операции:

- Сохраните копию исходного файла

  По умолчанию в момент построения защиты старый файл разрушается, а на его месте создается новый.

- Убедитесь, что на вашем диске достаточно свободного места.

  После применения защиты HASP Envelope ваше приложение будет занимать больше места на диске. Требуемое количество свободного места зависит от типа защищаемого приложения.

- Защита для DOS и Win32 остается в памяти все время, пока работает защищаемое приложение. Защита DOS требует 28Кб памяти.

Запуск HASP Envelope

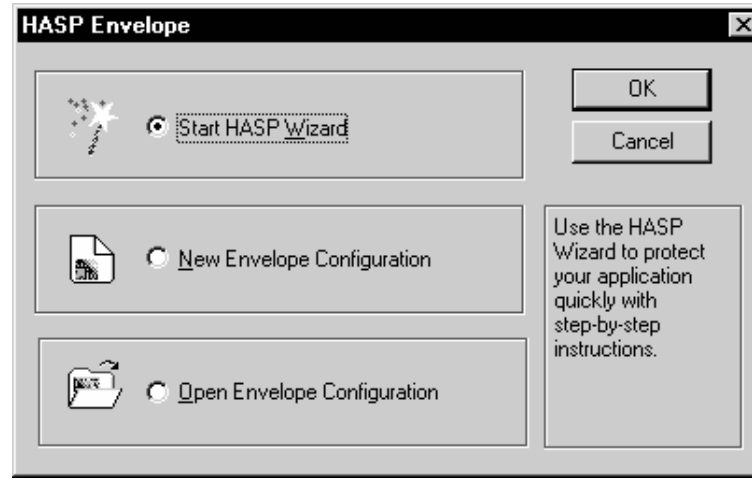
Для защиты вашего приложения вы можете использовать графический интерфейс, либо командную строку.

1.Установите драйвер устройства HASP. См. «Установка HASP в среде Windows».

2.Подключите ключ(и) HASP, с помощью которого (которых) вы хотите защищать ваше приложение, к компьютеру.

3.Запустите утилиту HASP Envelope. Для этого вы можете выбрать **HASP Envelope** на диске **HASP CD**, либо запустить программу **haspenv.exe**.

На экране появится окно HASP Envelope.



Вы можете использовать одну из трех опций:

- Выберите **Start HASP Wizard**, чтобы запустить пошагового мастера, который поможет вам установить защиту.
- Выберите **New Envelope Configuration,** чтобы начать сессию защиты с использованием новых параметров защиты.
- Выберите **Open Envelope Configuration,** чтобы начать сессию защиты с использованием сохраненных параметров защиты.

Защита приложения

Данный раздел описывает процесс защиты с использованием опций конфигурации **New** и **Open**.

1. В главной вкладке Envelope введите путь и имя приложения для защиты в поле **Unprotected File.**
2. Введите пароли HASP в соответствующие поля ввода.
3. В случае если вы разрабатываете защиту с использованием FAS, введите параметры FAS в зависимости от модели ключа HASP, который вы используете:

   Для HASP4 M1 и HASP4 M4 введите номер программы и количество активаций (вкладка Options).

   Для HASP4 Time введите номер программы и дату окончания действия лицензии (вкладка Options).

   Для HASP4 Net введите номер программы во вкладке Main. Вы можете также ввести количество активаций и/или количество лицензий (можно ввести оба параметра).
4. Заполните оставшиеся поля во вкладках Main, Options и Error Messages в зависимости от ваших нужд. Ниже эти параметры будут рассмотрены подробно.
5. Нажмите на пиктограмму Protect или выберите пункт **Protect Application** меню **Tools**.

   На экране появится окно, информирующее вас о том, что приложение было успешно защищено.
6. Нажмите OK.

   Если вы установили параметры защиты FAS, вас спросят, хотите ли вы сохранить эти параметры. Более полная информация приводится в разделе «Система полного управления доступом FAS».

   Теперь защищаемое приложение не сможет быть запущено без корректного ключа HASP.

   Повторяйте шаги 5-6 столько раз, сколько вам необходимо (для различных приложений либо для одного и того же приложения).

Защита файлов данных Win32

С помощью DataHASP вы сможете защищать файлы данных (например, приложения, созданные с помощью генераторов приложений). DataHASP шифрует данные таких файлов и позволяет расшифровывать их только авторизованным приложениям.

Никогда не шифруйте один и тот же файл дважды. В противном случае приложение не сможет получить доступ к такому файлу.

Для защиты вашего приложения вы можете использовать графический интерфейс, либо интерфейс командной строки.

В качестве альтернативы вы можете создать ваше собственное приложение и вызывать сервисы API 60, 61, 88 или 89 для шифрования/дешифрования файла данных, который используется любыми способами вашим приложением.

В случае защиты файлов данных вам будет также необходимо защитить приложение, использующее эти файлы. В этом разделе приводятся инструкции, следуя которым можно защитить одновременно и файлы данных, и приложение, их использующее.

Приложение, которое будет авторизовано для дешифрования файлов данных, не должно поддерживать опции копирования и экспорта файлов данных. В противном случае пользователь сможет экспортировать зашифрованные данные в нешифрованные файлы.

Подготовка к защите

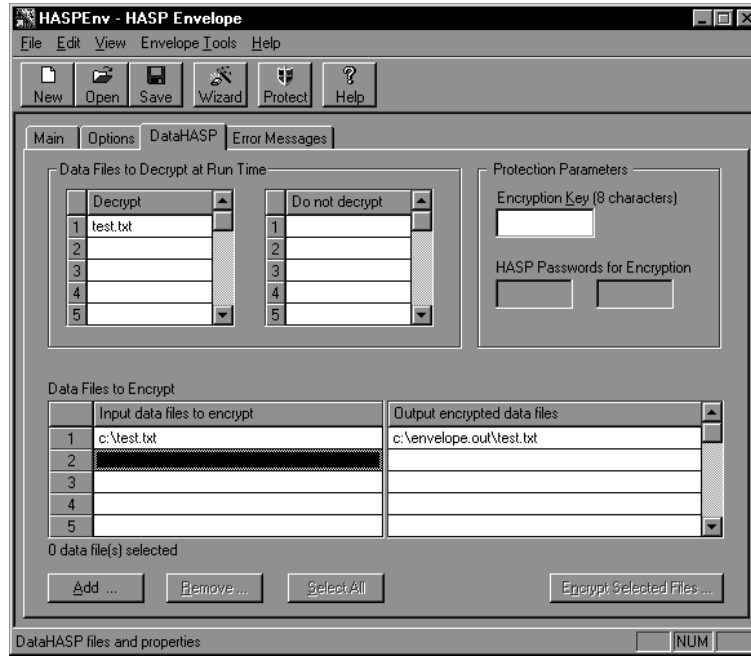
1. Поместите копию приложения (например, Notepad.exe), которое будет иметь доступ к зашифрованным данным, во временный каталог.
2. Создайте файл данных и сохраните его в том же каталоге (например, test.txt).
3. Подключите ключ HASP (с помощью которого вы хотите зашифровать данные) к компьютеру и убедитесь, что драйверы устройства установлены.

4. Запустите утилиту HASP Envelope. Для этого вы можете выбрать **HASP Envelope** из каталога **HASP CD**, либо запустить программу **haspenv.exe**.
5. Выберите **New Envelope Configuration.**

Установка параметров защиты

1. В главной вкладке Envelope введите путь и имя приложения (в нашем случае C:\Notepad.exe) в блоке **Unprotected File.**
2. Введите пароли HASP в соответствующие поля ввода.
3. Заполните оставшиеся поля во вкладках Main, Options и Error Messages в зависимости от ваших нужд. Ниже эти параметры будут рассмотрены подробно.
4. Введите имя файла данных (в нашем случае test.txt), который вы хотите зашифровать (а потом расшифровывать с помощью выбранного приложения), во вкладке DataHASP.

Рис. 5.2. Вкладка DataHASP



5. В поле **Encryption Key** введите восемь знаков (ключ шифрования).

6. Выберите **Add** в поле **Data Files to Encrypt** и выберите файл C:\test.txt.
7. В группе **Data Files to Decrypt at Run Time** введите имя файла test.txt в поле **Decrypt**. Вы можете также использовать маски файлов, например, *.* или *.txt.

Реализация защиты

1. Нажмите на пиктограмму **Protect** или выберите пункт меню [Tools/Protect Application] утилиты Envelope.

   На экране появится окно, информирующее вас о том, что приложение было успешно защищено, и что теперь вы можете зашифровать файлы данных.

2. Нажмите **OK**, затем нажмите кнопку **Encrypt Selected Files** в нижней части окна.

   На экране появится окно, информирующее вас о том, что все выбранные файлы были зашифрованы успешно. Успешно выполненная операция шифрования позволяет вам быть уверенным, что зашифрованные документы будут доступны только с помощью защищенного приложения. Незащищенное приложение не сможет прочитать эти файлы.

   Если вы установили параметры защиты FAS, вас спросят, хотите ли вы сохранить эти параметры. Более полная информация приведена в разделе «Сохранение параметров FAS» на стр. 50.

Защищенное приложение больше не сможет быть запущено в отсутствие корректного ключа HASP. Только защищенное приложение сможет получить доступ к защищенным файлам данных.

Сохранение параметров FAS

После применения защиты HASP Envelope для исполняемых файлов и файлов данных вы можете использовать HASP Envelope для сохранения параметров защиты FAS в память вашего ключа HASP сразу же после завершения процедуры установки защиты.

Вы можете также использовать HASP Edit для сохранения параметров на ваш ключ HASP.

Вы можете сохранить в памяти ключа HASP следующие параметры защиты FAS:

- Для HASP4 M1 и HASP4 M4 – номер программы и количество активаций.
- Для HASP4 Time – номер программы и дату окончания действия лицензии.
- Для HASP4 Net – номер программы, количество активаций и количество лицензий.

FAS не обязательна для ключей, устанавливаемых на отдельную станцию, но обязательна для сетевых версий (HASP4 Net). Поэтому помните, что сохранение параметров FAS в памяти HASP4 Net обязательно.

У вас есть выбор, использовать ли HASP Edit или HASP Envelope для определения параметров защиты FAS и сохранения их в память HASP. Тем не менее, вы должны использовать Envelope для определения номера программы. Присваивая номер программе с помощью HASP Envelope, вы создаете связь между ключом HASP и параметрами защиты вашей программы. Эти параметры сохраняются в памяти HASP. С помощью этой связи защищаемое приложение получает корректные параметры защиты из памяти HASP.

Вы можете сохранять параметры FAS на ключ HASP немедленно после проведения процедуры защиты HASP Envelope или независимо от процесса защиты Envelope.

Сохранение параметров FAS после применения процедуры защиты

Непосредственно после окончания процедуры защиты на экране появляется диалоговое окно. Оно дает вам возможность записать данные в память подключенного ключа HASP.

1. Выберите запись в локальный ключ HASP или в ключ HASP Net.
2. Выберите **Save Parameters**.
3. Чтобы записать одни и те же параметры защиты более чем в один ключ HASP, отключите ключ от системы, подключите другой, а затем выберите **Save Parameters**.
4. Чтобы прекратить запись в ключи HASP, выберите **Close**.

Сохранение параметров FAS независимо от процедуры защиты

1. Установите параметры FAS для использования с конкретным ключом HASP.
2. Выберите **Program HASP** из меню **Envelope Tools**.

   На экране появится диалоговое окно Save FAS Parameters.

1. Выберите запись в локальный ключ HASP или в ключ HASP4 Net.
2. Выберите **Save Parameters**.
3. Чтобы записать одни и те же параметры защиты более чем в один ключ HASP, отключите ключ от системы, подключите другой, а затем выберите **Save Parameters**.
4. Чтобы прекратить запись в ключи HASP, выберите **Close**.

Параметры HASP Envelope

В этом разделе описаны вкладки, поля ввода, меню и пиктограммы HASP Envelope.

Вкладка Main

Вы должны определить параметры вкладки Main для того, чтобы защитить ваше приложение. В случае использования локального ключа HASP заполните все поля ввода, кроме тех, что объединены под заголовком HASP4 Net Parameters. В случае использования ключа HASP4 Net заполните все поля, кроме тех, что объединены под заголовком HASP Passwords. В случае использования обоих типов ключей заполните все поля ввода.

Рис.5.3. Вкладка Main утилиты HASP Envelope

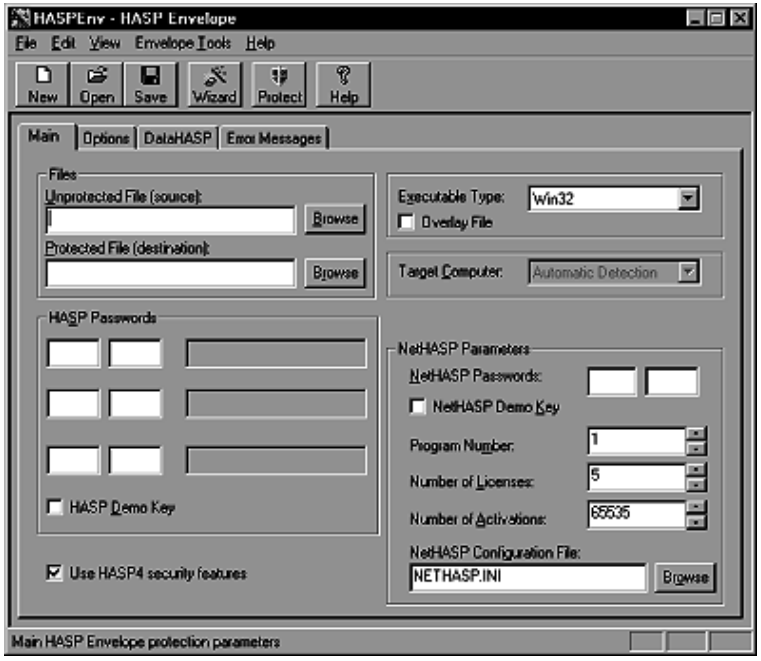


Таблица 5.1. Опции вкладки Main утилиты HASP Envelope

Опция	Описание
Unprotected File (source)	Введите путь и имя файла приложения, которое хотите защитить. Каждый файл должен иметь расширение .exe или .dll. Выберите Browse, чтобы найти файл на диске.
Protected File (destination)	Значение по умолчанию: Имя файла, введенное в поле Unprotected File. HASP Envelope по умолчанию переписывает исходный незащищенный файл при сохранении защищенной версии файла. Чтобы избежать потери исходного файла, присвойте защищенному приложению другое имя. Введите его в поле Protected File.
HASP Password	Введите пароли HASP для всех локальных ключей HASP, с помощью которых вы хотите защищать ваше приложение. Вы можете ввести до трех наборов паролей для защиты одного приложения с помощью трех различных ключей HASP.
HASP Demo Key	Отметьте это поле, чтобы ввести пароли демонстрационного ключа HASP автоматически.
Use HASP4 Security Features	Отметьте это поле, чтобы использовать улучшенные свойства защиты HASP4. Если вам необходима совместимость с поколением HASP-3, вы не должны использовать эту возможность.
Executable Type	HASP Envelope пытается определить тип исполняемого файла автоматически. Если утилита не может определить тип исполняемого файла, в этом поле появится значение **Unknown Type**.
Overlay File	Отметьте это поле в случае, если ваше приложение использует оверлеи или дополнительные данные, присоединяемые к исполняемому файлу.
Target Computer	Значение по умолчанию: IBM PC and compatibles. Задайте тип компьютеров, на котором будет исполняться ваше приложение. Например, если таким компьютером будет NEC, измените значение этого поля на NEC.
NetHASP Passwords	Если вы защищаете приложение с помощью HASP4 Net, введите пароли, которые вы получили для ключа HASP4 Net.
NetHASP Demo Key	Отметьте это поле, чтобы пароли демонстрационного ключа HASP4 Net заполнялись автоматически.

Опция	Описание
Program Number	Значение по умолчанию: 1 Вы можете защитить несколько приложений с помощью одного ключа HASP4 Net. Вы должны присвоить каждому защищаемому HASP4 Net приложению уникальный номер от 1 до 112. Запомните этот номер. Он понадобится вам для того, чтобы сохранить параметры защиты приложения в память ключей HASP4 Net для ваших клиентов.
Number of Licenses	Вы можете поставить ограничение на количество сетевых лицензий с помощью HASP4 Net, используя это поле. Число, введенное здесь, будет сохранено в памяти HASP4 Net. Введите число в это поле или выберите его с помощью стрелок. Выберите опцию **Unlimited,** чтобы присвоить бесконечное число лицензий. Доступные значения зависят от того, какая модель HASP4 Net установлена на компьютере. Вы можете присвоить этому полю значения от 0 до номера модели (например, от 0 до 5 для модели HASP4 Net 5). Чтобы сохранить количество лицензий в памяти HASP4 Net, выберите **Save Parameters** после применения защиты Envelope.
Number of Activations	Вы можете поставить ограничение на количество активаций приложения с помощью HASP4 Net, используя это поле. Число, введенное здесь, сохраняется в памяти HASP4 Net. Введите число в это поле или выберите его с помощью стрелок. Выберите опцию **Unlimited,** чтобы присвоить бесконечное число активаций. Чтобы сохранить количество активаций в памяти HASP4 Net, выберите **Save Parameters** после применения защиты Envelope.
NetHASP Configuration File	Значение по умолчанию: nethasp.ini Вы можете указать имя конфигурационного файла для системы HASP4 Net в этом поле. Выберите **Browse**, чтобы найти необходимый файл на диске. В случае, когда приложение находит файл конфигурации, оно читает этот файл и использует содержащуюся в нем информацию. В случае если файл не обнаружен, приложение использует значения по умолчанию.

Вкладка Options

Вкладка Options содержит дополнительные установки защиты.

Рис. 5.4. Вкладка Options утилиты HASP Envelope

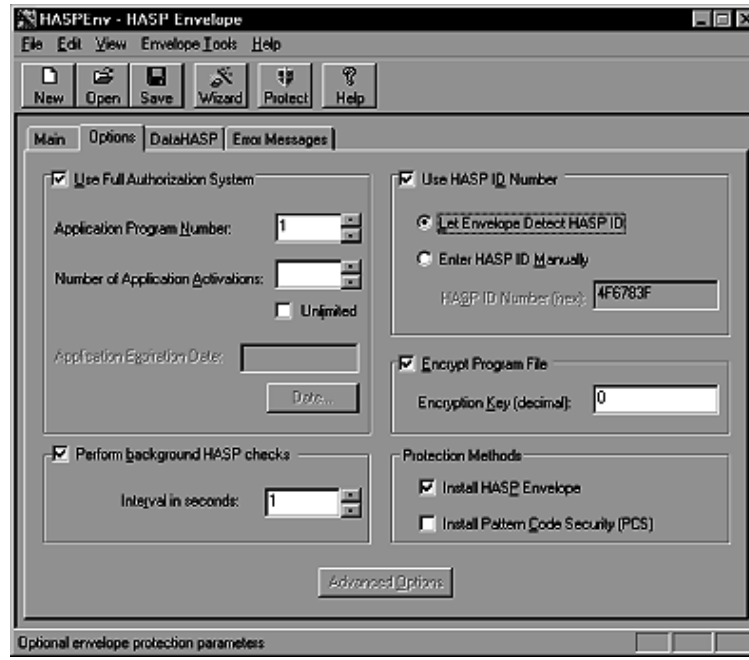


Таблица 5.2. Настройки Вкладки Options утилиты HASP Envelope

Опции	Описание
Use Full Authorization System (FAS)	Значение по умолчанию: Disabled Вы можете использовать FAS с ключами HASP4 M1, HASP4 M4, или HASP4 Time. Для получения более полной информации, обратитесь к разделу «Сохранение параметров FAS» на стр. 50. Чтобы использовать FAS, отметьте поле Use Full Authorization System. Так как FAS является неотъемлемой частью системы защиты HASP4 Net, параметры HASP4 Net FAS вводятся в полях, объединенных заголовком HASP4 Net Parameters на вкладке Main.
Application Program Number	Значение по умолчанию: 1 Если вы используете локальные ключи HASP, чтобы защитить несколько приложений с помощью FAS, присвойте уникальный номер каждому приложению. Запомните этот номер. Он понадобится вам позже для того, чтобы сохранить параметры защиты приложения во время программирования ключей HASP для ваших клиентов. Допустимые значения: 1 - 16 для HASP4 M1, 1 - 112 для HASP4 M4 и 1 - 8 для HASP4 Time.
Number of Application Activations	Если вы используете ключи HASP M1 или HASP4 M4 и систему FAS, введите количество активаций приложения в это поле. Это значение будет сохранено в памяти ключа. Вы можете ввести количество активаций для приложения или выбрать число стрелками. Выберите значение Unlimited, чтобы снять ограничение на количество активаций. Чтобы сохранить количество активаций в памяти HASP, выберите **Save Parameters** после применения защиты Envelope.
Application Expiration Date	Если вы используете ключи HASP4 Time и систему FAS, введите дату окончания действия лицензии в это поле. Это значение будет сохранено в памяти ключа. Выберите Date и выставьте год, месяц и день окончания срока действия лицензии. Чтобы сохранить дату в памяти HASP4 Time, выберите **Save Parameters** после применения защиты Envelope.
Perform Background HASP Checks	Значение по умолчанию: Disabled По умолчанию система HASP проверяет наличие ключа всякий раз при загрузке приложения. Если вы хотите включить повторные проверки во время работы приложения, добавьте фоновые проверки HASP. Чтобы сделать это, отметьте поле Perform Background HASP Checks.

Опция	Описание
Interval in Seconds	В случае если вы используете опцию Background HASP Checks, вы должны указать промежутки времени, через которые происходит проверка наличия ключа. Введите в это поле значение интервала в секундах.
Anti Debug and Reverse Engineering	Значение по умолчанию: Enabled С помощью этой опции вы сможете повысить уровень защищенности своего приложения, запрещая пользователям запускать ваше приложение с помощью отладчика и добавляя дополнительные анти-отладочные модули к исполняемому файлу вашего приложения. Чтобы добавить такую защиту, отметьте поле User Mode Debugger Protection. Чтобы добавить дополнительные модули защиты от отладчика, переместите ползунок Anti Debug Modules вправо
Use HASP ID Number	Значение по умолчанию: Disabled Используйте это поле для того, чтобы приложение запускалось только в том случае, если на компьютере установлен ключ HASP с заранее определенным ID-номером. Чтобы включить эту опцию, отметьте поле **Use HASP ID Number**. Эта опция доступна только для локальных ключей HASP с памятью. Вы можете ввести ID-номер ключа одним из следующих способов: • Выберите **Let Envelope Detect HASP ID,** чтобы прочитать ID-номер того ключа HASP, который подключен к порту на данный момент. • Выберите **Enter HASP ID Manually,** чтобы открыть поле, в которое вы сможете ввести номер ключа вручную. В этом случае вначале вы должны определить номер ключа с помощью утилиты HASP Edit. ID-номер ключа это переменная формата 32-bit unsigned integer. Этот номер должен вводиться в шестнадцатеричной системе счисления. С помощью этого метода вы можете не подключать конкретный ключ HASP к компьютеру во время выполнения процедуры защиты. В случае применения защиты с использованием более чем одного ключа HASP используйте ID-номер того ключа, для которого введены пароли в первое поле паролей. Номера других ключей проверяться не будут.

Опция	Описание
Encrypt Program File	Значение по умолчанию: Enabled Эта опция позволяет вам добавлять дополнительную защиту путем шифрования исполняемого файла. Те приложения, которые записывают информацию в собственный исполняемый файл, не могут быть зашифрованы подобным способом.
Encryption Key (decimal)	Процесс шифрования предполагает наличие нескольких ключей шифрования, выбираемых случайным образом утилитой HASP Envelope. Чтобы обеспечить максимальную защиту, вы можете установить один из этих ключей в любое значение от 0 до 65535.
Encryption Level	Вы можете указать частоту доступа к ключам HASP для шифрования. Переместите ползунок Encryption Level вправо, чтобы увеличить частоту обращений.
Protection Methods	В группе Protection Methods вы можете выбрать метод защиты, который хотите использовать. Вы можете выбрать метод Envelope, применить Защиту структурного кода либо использовать оба метода. В тот момент, когда вы щелкаете по пиктограмме Protect или выбираете пункт меню Envelope [Tools/Protect Application], вы применяете методы защиты, выбранные в этом поле.
Install HASP Envelope	Значение по умолчанию: Enabled Отметьте поле Install HASP Envelope, чтобы применить метод защиты Envelope. Этот метод будет реализован после того, как вы щелкнете по пиктограмме Protect или выберете пункт меню [Tools/ Protect Application].
Install Pattern Code Security (PCS)	Значение по умолчанию: Disabled Отметьте поле Install Pattern Code Security (PCS) чтобы установить PCS и улучшить защиту. Метод PCS будет реализован после того, как вы щелкнете по пиктограмме Protect или выберете пункт меню [Tools/Protect Application].

Вкладка DataHASP

Используйте вкладку DataHASP, чтобы защитить файлы данных. С помощью этой вкладки вы сможете:

- Указать, какие файлы данных вы хотите защитить
- Указать, какие из зашифрованных файлов данных должны быть дешифрованы во время исполнения вашего защищенного приложения
- Зашифровать файлы данных

В дополнение к шифрованию файлов данных вам будет необходимо указать, какие приложения авторизованы для доступа к зашифрованным файлам данных. Вам будет также необходимо защитить эти приложения. Сделайте это, используя соответствующие поля на вкладке Main и других вкладках, относящихся к защите приложений.

В качестве альтернативы вы можете создать ваше собственное приложение и вызывать сервисы API 60, 61, 88 или 89 для шифрования/дешифрования файла данных, который используется любыми способами вашим приложением.

Никогда не шифруйте один и тот же файл дважды. В противном случае, приложение не сможет получить доступ к такому файлу.

Рис.5.5. Вкладка DataHASP утилиты HASP Envelope

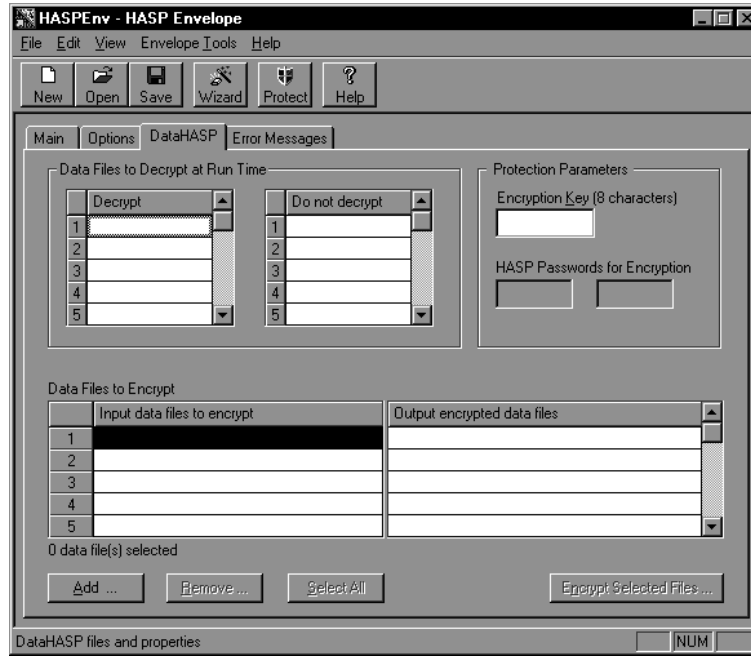


Таблица 5.3. Опции вкладки DataHASP утилиты HASP Envelope

Опция	Описание
Data Files to Decrypt at Run Time	В этой группе полей необходимо ввести имена файлов (или шаблон) для тех файлов данных, которые ваше приложение должно будет дешифровывать во время своего исполнения. Здесь также можно ввести имена тех файлов (из тех, что введены), которые не должны расшифровываться приложением. Например, приложение должно иметь доступ ко всем файлам с именами ***.adb**. Представьте, что приложение должно иметь доступ к файлу **xyz.adb**, который не является файлом данных, зашифрованным ранее, а генерируется самим приложением. В этом случае в поле **Decrypt** введите *.adb. В поле **Do Not Decrypt** введите **xyz.adb**. Таким образом, будут зашифрованы все файлы с расширением **adb**, кроме файла с именем **xyz.adb**.
Decrypt	В это поле необходимо ввести имена файлов (или шаблон с использованием символов **?** и *) для тех файлов данных, которые ваше приложение должно будет дешифровывать во время своего исполнения.
Do not decrypt	В это поле необходимо ввести имена файлов (или шаблон с использованием символов ? и *) для тех файлов данных, которые ваше приложение не должно дешифровывать во время своего исполнения.
Protection Parameters	Введя параметры защиты, вы устанавливаете ключ шифрования для зашифровывания файлов данных.
Encryption Key	Введите ключ защиты (до восьми символов) в поле **Encryption Key**. Этот ключ станет частью алгоритма шифрования.
HASP Passwords for Encryption	Пароли ключа HASP, введенные во вкладке **Main**, автоматически отображаются в поле **HASP Passwords for Encryption**. Открыть зашифрованные файлы можно только тем ключом HASP, которому принадлежат эти пароли.
Data Files to Encrypt	Перед шифрованием файлов данных убедитесь, что вы установили параметры защиты (**Protection Parameters**) во вкладке DataHASP.  Никогда не шифруйте один и тот же файл дважды. В противном случае приложение не сможет получить доступ к такому файлу.

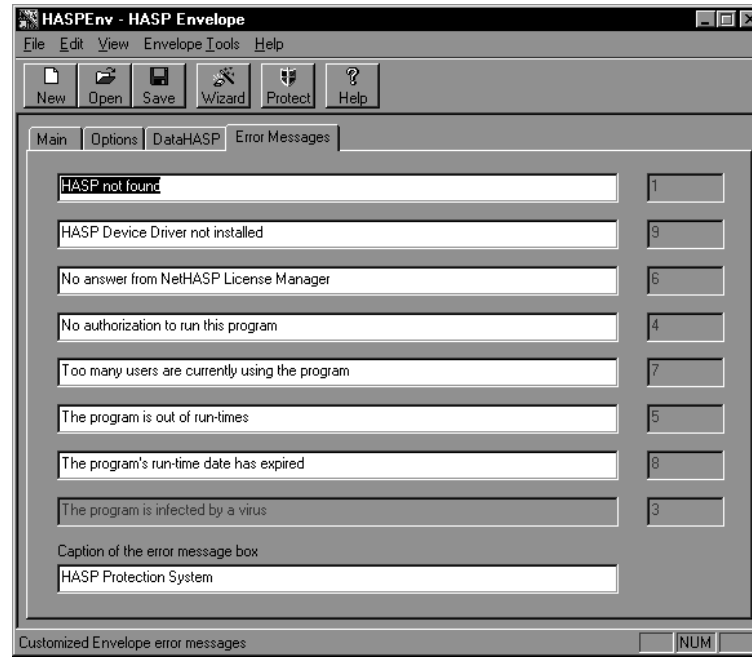
Опция	Описание
Input Data Files to Encrypt	Введите имена файлов (с путями) для шифрования или нажмите **Add,** чтобы выбрать файлы на диске.
Output Encrypted Data Files	DataHASP копирует имена файлов, указанные в полях **Input Data Files** в поля **Encrypt** и помещает их в новый каталог. Новый путь к файлам появляется в этих полях автоматически.

После передачи зашифрованных файлов данных (совместно с приложением) конечным пользователям вы сможете присылать им новые зашифрованные файлы данных. Просто зашифруйте новые файлы данных, убедившись, что их имена совпадают с указанными ранее в полях Data Files to Decrypt at Run Time шаблонами. Если их имена не совпадают с указанными ранее шаблонами, вам придется совершить операцию шифрования всех файлов снова (вы должны будете проделать эту операцию и со старыми, и с новыми файлами).

Вкладка Error Messages

На этой вкладке указаны сообщения об ошибках, которые появляются в случае возникновения проблемы при запуске приложения. Вы можете редактировать эти сообщения. Например, вы можете написать сообщения об ошибках на другом языке, используя поля ввода на этой вкладке.

Рис.5.6. Вкладка Error Messages утилиты HASP Envelope



Ключи командной строки утилиты HASP Envelope

Вы можете сэкономить время, используя утилиту HASP Envelope в режиме командной строки и используя ключи, описанные ниже. Этот режим позволяет вам производить операции шифрования в фоновом режиме с использованием .bat-файлов. Из командной строки вы можете запускать версии DOS, Win16 и Win32 HASP Envelope.

Таблица 5.4. Исполняемые приложения HASP Envelope

Приложение	Имя исполняемого файла
DOS Envelope	Instdos.exe
Win16 Envelope	Instw16.exe
Win32 Envelope	Instw32.exe

Следующая таблица описывает различные ключи командной строки. Если не указано обратное, ключ работает в любой версии HASP Envelope.

Таблица 5.5. Ключи утилиты HASP Envelope

Ключ	Описание
-c	Указывает количество анти-отладочных модулей (1-50). Работает только для Win 16 и Win 32 Envelope.
-cfgfile <filename>	Загружает конфигурационный файл и использует параметры, описанные в нем.
-createcfg <filename>	Генерирует конфигурационный файл. Для Win16 и Win32 Envelope.

Опция	Описание
-dhfilter <filename>	Имена или шаблоны файлов для расшифровки в момент исполнения (стандартное представление, например, *.* или aa??.txt). Могут быть определены максимум восемь файлов или шаблонов. Только для Win32 Envelope.
-dhfilterx <filename>	Имена и шаблоны имен файлов (из тех файлов, которые выбраны для шифрования), которые не должны быть дешифрованы в процессе работы приложения (стандартное представление имен, например, *.* или aa??.txt). Допустимо использовать максимум восемь имен или шаблонов. Только для Win32 Envelope.
-dhkey <key>	Указывает ключ шифрования для файлов данных (до восьми символов). Только для Win32 Envelope.
-drvwait <seconds>	Увеличивает временной интервал для поиска драйвера. Допустимые значения – от 0 до 255. Только для Win32 Envelope.
-enc	Значение по умолчанию. Зашифровывает файлы в момент установки защиты.
-enclevel	Уровень защиты для защиты данных (1-5). Только для Win16 и Win32 Envelope.
-exsecnum <num>	Не шифровать номер секции. Только для Win32 Envelope.
-fas	Использовать FAS.
-fasprgnum <prog number>	Указать номер программы FAS.
-h3easy	Исполнять программу в случае, если установлен любой ключ HASP.
-h3hard	Значение по умолчанию. Исполнять программу только если присутствует заранее указанный ключ HASP.
-h3pass <pass1> <pass2>	Указать пароли HASP.
-hasp4	Использовать свойства защиты HASP4. Только для Win16 и Win32 Envelope.
-help or -?	Показать список всех доступных ключей с краткими описаниями, затем выйти из программы.

Опция	Описание
-highsecoff	Выключить режим повышенной защиты. Значение по умолчанию для защиты с помощью ключа HASP4 Net необходимо для процессоров 286 и ниже. Только для DOS Envelope.
-highsecon	Включить режим повышенной защиты. Значение по умолчанию для защиты с помощью локального ключа HASP4. Только для DOS Envelope.
-ibm	Значение по умолчанию. Целевой компьютер IBM PC или совместимый.
-interval <value>	Указать интервал для фоновых проверок в секундах.
-loader<filename>	Указать имя загрузчика (в случае защиты приложения, использующего оверлеи). Только для DOS Envelope.
-loginx	Для использования с приложениями, которые не производят автоматическое отключение от HASP4 Net (например, Visual Basic). Только для Win32 Envelope.
-mhid <IdNumber>	Указать ID-номер HASP ключа HASP4 M1, HASP4 M4 или HASP4 Time.
-mhpass <pass1> <pass2>	Указать пароли HASP4 M1, HASP4 M4 или HASP4 Time.
-nec	Целевой компьютер NEC.
-netcfg <filename>	Указать имя файла конфигурации HASP4 Net.
-nhpass <pass1><pass2>	Указать пароли HASP4 Net.
-nodbgcheck	Отключить опцию защиты от отладчика. Только для Win32 Envelope.
-noenc	Не зашифровывать файлы во время применения защиты.
-nofas	Значение по умолчанию. Не использовать FAS.
-nohasp4	Отключить опции защиты HASP4
-norandomsize	Отключить опцию Random File Size. Только для DOS Envelope.
-nores	Установить режим повышенной совместимости с резидентными программами DOS. Этот ключ отключает фоновые проверки наличия HASP. Только для DOS Envelope.

Опция	Описание
-nospecial	Значение по умолчанию. Используйте этот ключ для защиты приложений, не использующих оверлеи. Ключ идентичен ключу Special Overlays Mode = NO .Только для DOS Envelope.
-novir	Отключает автоматическую антивирусную защиту. Только для DOS Envelope.
-out <filename>	Переименовать исполняемый файл защищенного приложения.
-ovl	Работа с оверлеями. Только для Win32 Envelope
-pcs	Установить PCS в фоновом режиме.
-prg <filename>	Указать имя файла защищаемого приложения.
-prgnum <program number>	Указать номер программы (для Win16 или DOS Envelope). Используется для FAS либо HASP4 Net; для использования Win32 Envelope только с ключами HASP Net.
-quiz <intensity>	Указать, как часто случайные данные будут записываться и читаться с использованием ключа HASP. Допустимые значения – от 0 до 50. Только для Win32 Envelope.
-randomsize	Значение по умолчанию. Включает опцию Random File Size. Только для DOS Envelope.
-special	Управляет оверлеями и расширениями DOS. Ключ идентичен команде Special Overlays Mode = NO. Только для DOS Envelope.
-special1	Альтернативное управление оверлеями и расширениями DOS. Ключ идентичен команде Special Overlays Mode = YES - Method 1. Только для DOS Envelope.
-vir	Значение по умолчанию. Автоматическая антивирусная защита. Только для DOS Envelope.

Дополнительная информация о HASP4 Net

Защита для сетей и локальных компьютеров

HASP Envelope позволяет вам защищать приложения как для работы в сети, так и для работы на локальных машинах.

Если вы защищаете приложение для работы и в том и в другом случаях, ваше приложение производит следующие проверки:

- Во-первых, приложение проверяет, установлен ли ключ HASP на том компьютере, где запущено приложение.
- В случае, когда приложение не находит ключа HASP на локальной машине, оно производит проверку на присутствие в сети ключа HASP4 Net.

Для того чтобы позволить HASP Envelope защищать одновременно и сетевые установки, и локальные машины, введите корректные пароли локального ключа HASP и ключа HASP4 Net в соответствующие поля ввода.

Время простоя HASP4 Net

Для приложения, защищенного с помощью HASP4 Envelope, время, после которого станция считается простаивающей, равно времени, установленному по умолчанию в HASP LM.

Отключение от HASP4 Net для приложений Win16

HASP Envelope автоматически добавляет подключение к HASP4 Net в начало программы, а отключение от HASP4 Net – в конец. Однако в приложениях Win16, работающих по протоколу TCP/IP, отключение не выполняется. Чтобы оно выполнялось, убедитесь, что установлен какой-либо другой сетевой протокол.

Часто задаваемые вопросы

Вопрос: Как долго приложение, защищенное с помощью HASP Envelope, загружается в оперативную память?

Ответ: Это время зависит от размера приложения и производительности компьютера. В любом случае время загрузки увеличивается не более, чем на несколько секунд.

Вопрос: Влияет ли защита от отладчика HASP на другие приложения, исполняемые на компьютере?

Ответ: Нет. Эти модули не влияют на другие приложения, они полностью прозрачны.

Вопрос: Проверяет ли приложение, защищенное HASP Envelope, наличие ключа во время исполнения?

Ответ: Да. С помощью HASP Envelope можно предусмотреть проверку приложением наличия ключа.

Вопрос: Могу ли я использовать HASP Envelope для защиты с помощью HASP4 M1 или HASP4 M4 и HASP4 Net одновременно?

Ответ: Да. Введите пароли ваших ключей HASP4 M1 и HASP4 M4, а также пароли ключей HASP4 Net, заполните все остальные поля, относящиеся к этим ключам. После применения процедуры защиты приложение сначала будет искать локальные ключи HASP4 M1 и HASP4 M4 на машине, на которой оно было запущено. Если ключи не были обнаружены, приложение произведет проверку присутствия ключа HASP4 Net в сети.

Вопрос: В каких случаях я должен указывать номер программы при использовании HASP Envelope?

Ответ: В случае использования HASP4 Net - всегда! Если вы используете HASP4 M1 или HASP4 M4, то только в том случае, если вы используете систему защиты FAS.

Доступ к ключам с использованием HASP Edit

HASP Edit - это приложение, позволяющее вам осуществлять доступ к ключам HASP.

HASP Edit работает в среде Mac OS 9.x и Mac OS 10.1, а также Windows 95/98/ME/NT/2000.

Ключи, запрограммированные с помощью одного из приложений HASP Edit, могут быть использованы на всех поддерживаемых платформах. В дополнение к этому вы можете также использовать HASP API.

HASP Edit используется в основном для двух типов операций:

- Чтобы подготовить приложение к защите путем шифрования данных и получения ID-номера HASP.
- Для программирования ключей HASP и их подготовки для распространения среди ваших клиентов методом редактирования памяти HASP, установки параметров защиты и установки часов реального времени для HASP4 Time.

HASP Edit для Windows

В данном разделе поясняется использование приложения HASP Edit для Windows при выполнения основных операций. Для получения более подробной информации обращайтесь к файлу справки программы.

Запуск HaspEdit

Чтобы запустить HaspEdit:

1. Подключите ключ HASP к вашему компьютеру.

> Не подключайте более одного ключа, поскольку вы можете осуществлять запись в память лишь одного ключа в одно и то же время. HASP Edit не различает разные ключи с одинаковым кодом разработчика и из-за этого может по ошибке разрушить данные, хранящиеся на ключе, если в системе установлено более одного ключа.

2. Выберите **HaspEdit** в меню **HASP CD** (в каталоге Windows Programs). На экране появится окно HaspEdit:

3. Начните новую сессию HaspEdit или откройте конфигурационный файл HaspEdit (чтобы работать с уже существующим файлом).

Файл конфигурации HaspEdit

Файл конфигурации HaspEdit сохраняет набор параметров HASP, используя HaspEdit. В этом файле вы можете сохранить детали той модели HASP, которую вы используете, пароли ключей, ID-номера, а также параметры защиты FAS. Сохранение этих параметров полезно в случае, когда вы собираетесь запрограммировать несколько ключей, используя одну и ту же информацию.

В тот момент, когда вы выбираете пункт **Save** меню **File**, вы сохраняете параметры HASP в файл, а не в память ключа HASP. Заголовок главного окна HaspEdit теперь будет содержать имя сохраненного файла. Перед закрытием HaspEdit спросит вас, сохранять ли текущие параметры HaspEdit в конфигурационный файл.

Файл конфигурации HaspEdit содержит ваши секретные пароли. Относитесь к этому файлу как будто это исходный код, который вы собираетесь защитить. Обеспечьте его безопасность.

После запуска HaspEdit вы можете либо начать новую сессию HaspEdit, либо загрузить предыдущую, открыв соответствующий файл конфигурации.

Запуск новой сессии HaspEdit

Чтобы запустить новую сессию HaspEdit:

1. Выберите **New HaspEdit Session** и нажмите **OK**.

   На экране появится диалог HASP Password.

2. Введите пароли HASP. Если вы используете ключ HASP demo, выберите **A Hasp demo key**, чтобы ввести пароли HASP demo автоматически.

3. Нажмите **OK**.

   Если были введены правильные пароли (правильные пароли для того ключа, который подключен на данный момент), на экране появится окно Configuration. Теперь вы можете использовать HaspEdit.

Открытие существующего файла конфигурации HaspEdit

Чтобы открыть существующий файл конфигурации:

1. Выберите **Open HaspEdit Configuration** и нажмите **OK**.

   На экране появится диалог Open.

2. Выберите или введите имя файла, который вы хотите открыть.

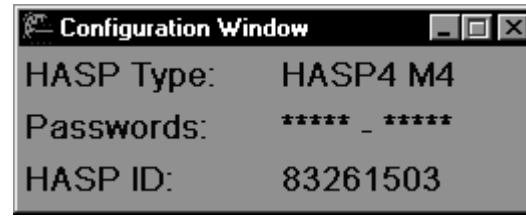
3. Нажмите **Open**.

   В случае если сохраненный пароль и модель ключа совпадают с паролем и моделью подключенного ключа, на экране появится окно Configuration. Теперь вы можете использовать HaspEdit.

Окно Configuration утилиты HaspEdit

Окно Configuration появляется на экране тогда, когда вы открываете файл конфигурации или начинаете новую сессию. В тот момент, когда вы закрываете это окно, вам предлагают закрыть сессию HaspEdit.

Рис.6.1 Окно Configuration утилиты HaspEdit



В этом окне отражается следующая информация:

- Модель HASP .
- Пароли ключа HASP. Эта информация может быть спрятана (для этого нужно отметить пункт Passwords меню View).
- Уникальный ID-номер ключа HASP (десятичный).

Подготовка к защите приложения

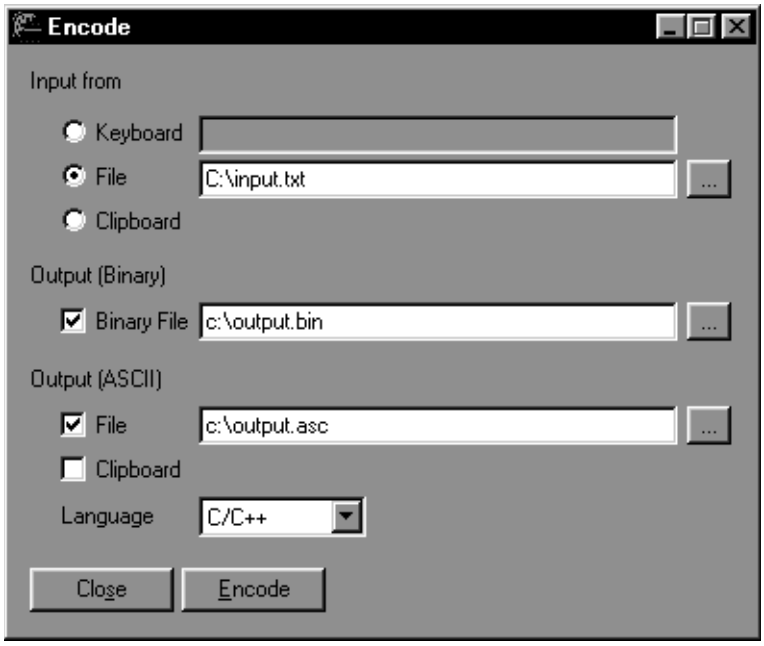
Используйте HaspEdit, чтобы подготовить приложение к защите методом шифрования данных и получения уникального ID-номера HASP ID.

Шифрование данных

Эта функция дает вам возможность зашифровать данные, используя ключ HASP4. После шифрования вы сможете использовать зашифрованные данные в вашем приложении и расшифровывать их во время исполнения, используя соответствующие функции HASP API.

Чтобы зашифровать данные:

1. Выберите пункт **Encoding** меню **Tools** или щелкнете по пиктограмме **Encode**. На экране появится окно **Encode**:



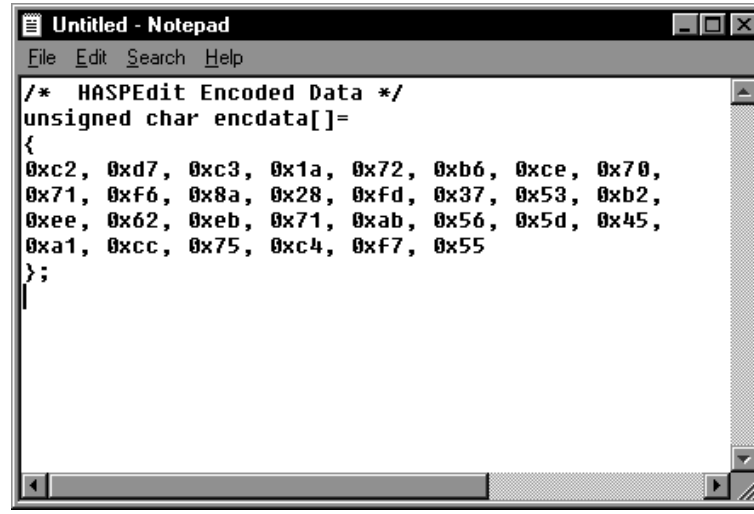
2. Введите или выберите источник данных, который подлежит шифрованию. Вы можете шифровать данные из буфера обмена или из файла. Вы также можете ввести строку прямо с клавиатуры.

3. Введите или выберите имя бинарного файла для зашифрованных данных (необязательно).

   Выходные зашифрованные данные не смогут быть скопированы в текстовый редактор, потому что символы в выходных данных нечитаемые.

4. Введите имя файла ASCII (для сохранения данных) либо выберите **Clipboard** для зашифрованных данных (необязательно).

Данные в формате ASCII выглядят следующим образом:



5. Выберите язык программирования

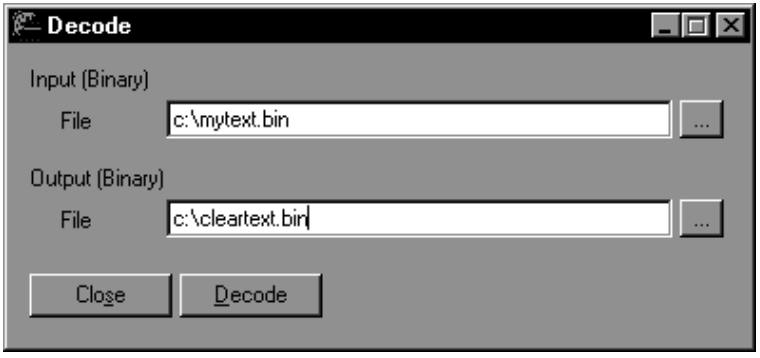
   Эта опция позволит сгенерировать файл заголовков для используемого вами языка. Таким образом, вы сможете использовать зашифрованные данные без каких-либо усилий с вашей стороны.

6. Нажмите **Encode**, чтобы начать шифрование.

Теоретически вы можете зашифровать до 4GB данных за один раз. Примите во внимание, что HaspEdit нуждается в выделении буферов для шифрования и дешифрования данных. Это означает, что единственное ограничение на размер используемых данных накладывается размером оперативной памяти и свободного пространства на жестком диске вашего компьютера.

Расшифровка данных

1. Выберите **Decoding** из меню **Tools** или щелкните по пиктограмме **Decode**. На экране появится окно Decode:



2. Введите или выберите имя бинарного файла данных, содержащего зашифрованные данные.
3. Введите или выберите имя бинарного файла данных для сохранения в нем дешифрованной информации.
4. Нажмите **Decode**, чтобы начать процесс дешифрования.

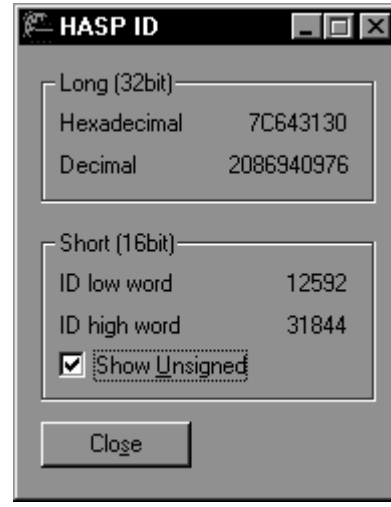
Получение ID-номера HASP

Каждый ключ HASP имеет свой уникальный ID-номер. Вы можете вставить проверку на корректность этого номера в ваше приложение, используя HASP Envelope либо API. Чтобы сделать это, сначала следует использовать HaspEdit, чтобы узнать ID-номер подключенного ключа.

С помощью HASP Edit вы можете получить ID-номер ключей HASP4 с памятью, который является уникальным для каждого ключа. Вы можете вставить проверку на определенный ID-номер HASP в вашем приложении с использованием сервиса 6 (Hasp ID).

Проверка ID–номера ключа HASP с памятью

1. Подключите ключ HASP к вашему компьютеру.
2. Выберите **HASP ID** в меню **Tools** или щелкните по пиктограмме **ID**. На экране появится окно **HASP ID**:



   В этом окне отображается ID-номер (32-разрядное целое число без знака) в шестнадцатеричном и десятичном форматах, а также этот же номер в виде двух 16-разрядных слов.

3. Если вы выбрали защиту с помощью HASP Envelope, вам необходимо использовать ID-номер в шестнадцатеричном формате. Позже вы сможете ввести его в HASP Envelope.
4. Если вы выбрали защиту с помощью HASP API, вам необходимо использовать **ID low word** и **ID high word,** чтобы сравнить их со значениями Par1 и Par2, возвращаемыми процедурой hasp (Par1 and Par2 of Service 6: HASP ID). Также вы можете запомнить ID-номер и сравнить его со значением, вычисляемым по формуле, описанной в Сервисе 6.

Чтобы скопировать содержимое окна HASP ID, выберите Copy в меню Edit.

Установка параметров защиты FAS

Система полного управления доступом FAS позволяет распространять демонстрационные версии приложений, сдавать приложения в аренду и защищать несколько приложений с использованием одного и того же ключа HASP Memory.

В случае использования ключей HASP4 M1 или HASP4 M4 использование FAS необязательно. Вы можете установить эту защиту с помощью HASP Envelope. Использование FAS обязательно в случае работы с ключами HASP4 Net. Опции FAS настраиваются с помощью HASP Envelope и/или HASP API.

Вы можете устанавливать FAS только на ключах HASP со встроенной памятью.

В данном разделе описаны методы программирования ключей HASP в случае применения FAS.

Список приложений FAS

В случае использования FAS вы создаете список защищаемых приложений и параметры защиты для каждого из них.

Вы можете использовать FAS для одновременной защиты:

1. 16 приложений с помощью ключа HASP4 M1
2. 112 приложений с помощью ключа HASP4 M4
3. 8 приложений с помощью ключа HASP4 Time
4. 112 приложений с помощью ключа HASP4 Net

Установите для каждой защищаемой программы свой уникальный номер (с помощью **HaspEdit**) для идентификации приложения и возможности установки индивидуальных параметров защиты для каждого из них.

Параметры защиты FAS

Вы можете установить индивидуальные параметры защиты для каждого приложения (которое находится в списке приложений, защищаемых с помощью FAS) с помощью утилиты HaspEdit. Указанные вами параметры зависят от модели ключа HASP, которую вы используете.

Используйте FAS для:

- Ограничения количества запусков приложения с ключами HASP4 M1 или HASP4 M4.

- Установки даты окончания действия лицензии с ключом HASP4 Time.
- Ограничения количества станций, на которых одновременно может быть запущено приложение, а также количества запусков, с ключами HASP4 Net.

Используйте HaspEdit, чтобы устанавливать и/или модифицировать параметры защиты для каждого приложения. После того, как вы установите параметры защиты для каждого приложения, запишите список приложений и их параметры в память ключа HASP.

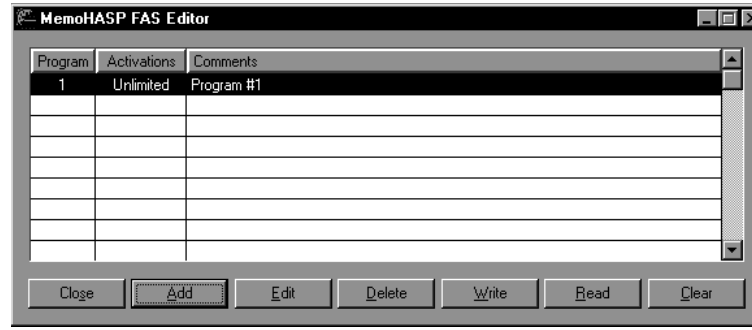
В следующем разделе описаны методы редактирования списка приложений FAS для каждой модели ключей HASP.

Определение списка приложений FAS для ключей HASP4 M1 и HASP4 M4

Вы можете ограничить количество запусков для каждого из приложений из списка (для ключей HASP4 M1 или HASP4 M4).

Определение список приложений FAS для ключей HASP4 M1 и HASP4 M4:

1. Выберите пункт **Full Authorization System (FAS)** в меню **HASP Tools** и нажмите **MemoHASP** либо щелкните по пиктограмме **FAS**. На экране появится окно **MemoHASP FAS Editor**:



2. Нажмите **Add,** чтобы добавить приложения, которые вы хотите защитить. На экране появится окно **Add Program**.

3. В поле ввода **Program** введите уникальный номер программы, которую вы хотите защитить. Значение в этом поле может варьироваться от 1 до 16 для ключа HASP4 M1 и от 1 до 112 для ключа HASP4 M4. Номер программы должен быть идентичным номеру, который вы присвоили приложению во время организации защиты с помощью HASP Envelope.

Следует запоминать номера, которые были присвоены различным приложениям. Этот номер будет необходим при использовании HASP Envelope для идентификации того приложения, которое вы хотите защитить.

4. Введите максимальное количество разрешенных запусков приложения в поле **Activations.** Отметьте поле **Unlimited**, чтобы разрешить неограниченное количество запусков.

5. Добавьте описание программы в поле **Comments** (например, имя исполняемого файла или другие комментарии, облегчающие идентификацию приложения).

Комментарии, введенные в поле Comments утилиты MemoHASP FAS Editor, сохраняются в файле конфигурации HaspEdit, а не в памяти ключа.

6. Нажмите **OK**. Приложение будет добавлено в список утилиты MemoHASP FAS Editor.
7. Нажмите **Write**. На экране появится окно **Write HASP Memory**.
8. Сохраните изменения на ключ HASP. Для этого вы можете:
   - Нажать Write, чтобы сохранить все изменения в память.
   - Отметить **Check Write Selected Programs Only,** чтобы сохранить в память только изменения, касающиеся выбранных приложений, затем нажать **Write**.

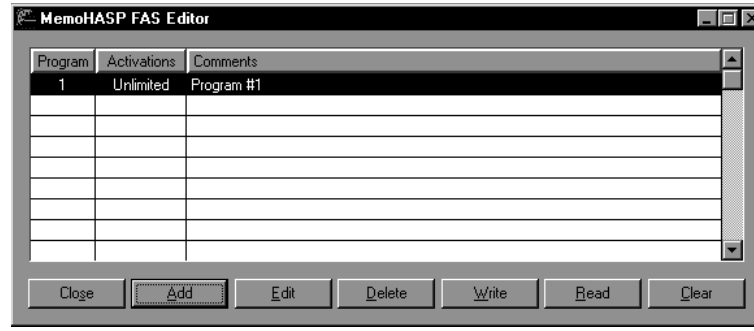
   Все изменения в памяти выделяются на экране красным цветом. Нажатие кнопки **Write** сохраняет изменения в ключ HASP. Все данные, записанные на ключе HASP, отображаются на экране синим цветом.

Установка списка приложений FAS для ключа HASP4 Time

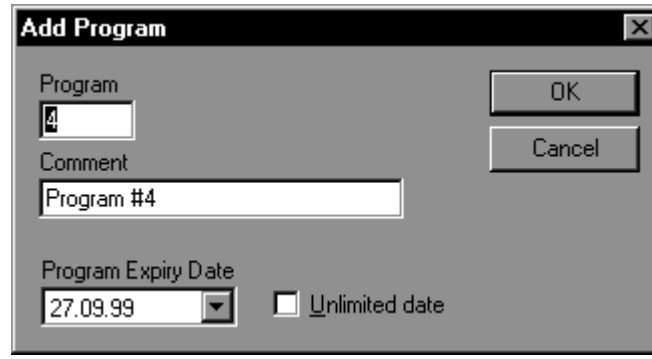
Используйте FAS для установки даты окончания действия лицензии, после наступления которой приложения, защищенные с помощью HASP4 Time, не смогут быть запущены.

Чтобы установить список приложений FAS для ключа HASP4 Time:

1. Выберите пункт **Full Authorization System (FAS)** из меню **HASP Tools** и нажмите **TimeHASP** или щелкните по пиктограмме **FAS**. На экране появится окно TimeHASP FAS Editor:



2. Нажмите **Add,** чтобы добавить выбранные для защиты приложения. На экране появится диалог **Add Program**:



3. В поле ввода **Program** введите уникальный номер программы, которую вы хотите защитить. Для ключа HASP4 Time значение в этом поле может варьироваться от 1 до 8. Номер программы должен быть идентичен номеру, который вы присвоили приложению во время организации защиты с помощью HASP Envelope.

> Следует запомнить, какие номера были присвоены различным приложениям. Эти номера будут необходимы при работе с утилитой HASP Envelope для идентификации того приложения, которое вы хотите защитить.

4. В полях группы **Program Expiration Date** введите день, месяц и год окончания действия лицензии. Отметьте поле **Unlimited Date** в случае, если вы не хотите ограничивать срок действия лицензии.
5. Добавьте описание программы в поле **Comments** (например, имя исполняемого файла или другие комментарии, облегчающие идентификацию приложения).

> Комментарии, введенные в поле Comments утилиты TimeHASP FAS Editor, сохраняются в файле конфигурации HaspEdit, а не в памяти ключа.

6. Нажмите **OK**. Приложение будет добавлено в список утилиты TimeHASP FAS Editor.

7. Нажмите **Write**. На экране появится окно **Write HASP Memory**.
8. Сохраните изменения на ключ HASP4 Time. Вы можете:
   - Нажать **Write**, чтобы сохранить все изменения в память.
   - Отметить **Check Write Selected Programs Only**, чтобы сохранить в память только изменения, касающиеся выбранных приложений, затем нажать **Write**.

   Все изменения в памяти выделяются на экране красным цветом. Нажатие кнопки **Write** сохраняет изменения в ключ HASP. Все данные, записанные на ключе HASP, отображаются на экране синим цветом.

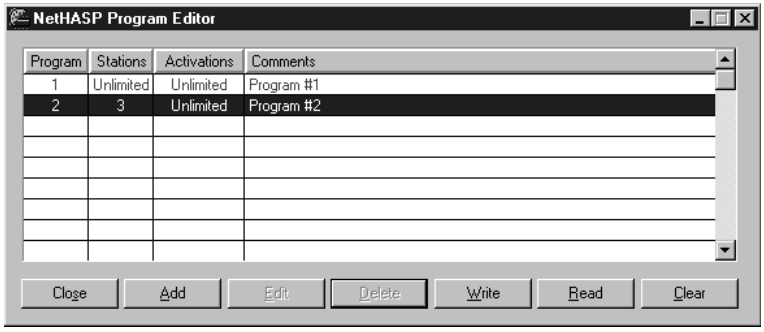
Установка списка приложений FAS для ключа HASP4 Net

С помощью ключа HASP4 Net вы можете ограничить количество станций, на которых приложение может быть запущенно одновременно и/или количество запусков для каждого из приложений, находящихся в списке.

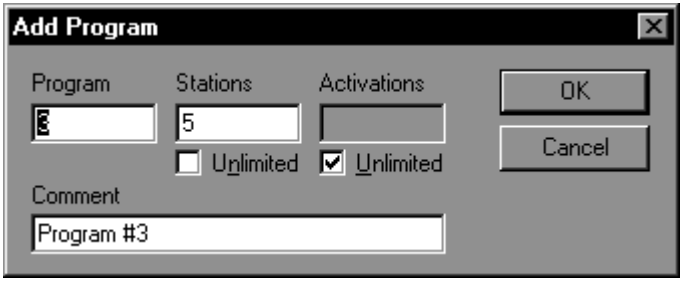
Если используется ключ HASP4 Net, применение FAS обязательно даже в том случае, когда вы защищаете локальное приложение.

Установка списка приложений FAS для ключа HASP4 Net:

1. Выберите пункт **Full Authorization System (FAS)** из меню **HASP Tools** и нажмите **NetHASP** или щелкните по пиктограмме **FAS**. На экране появится окно NetHASP FAS Editor:



2. Нажмите **Add**, чтобы добавить выбранные для защиты приложения. На экране появится диалог **Add Program**:



3. В поле ввода **Program** введите уникальный номер программы, которую вы хотите защитить. Значение в этом поле может варьироваться от 1 до 112 для ключа HASP4 Net. Номер программы должен быть идентичным номеру, который вы присвоили приложению во время защиты с помощью HASP Envelope или HASP API.

Следует запоминать номера, которые были присвоены различным приложениям. Этот номер будет необходим при использовании HASP Envelope для идентификации того приложения, которое вы хотите защитить.

4. В поле **Stations** введите максимальное количество станций, которые могут запускать приложение одновременно.
5. Максимальное количество станций зависит от используемого ключа HASP4 Net. HASP4 Net5 допускает максимально 5 станций, HASP4 Net10 допускает максимально 10 станций и т.д. В случае использования ключа HASP4 NetU вы можете разрешить одновременный доступ любому количеству станций. Для этого отметьте поле **Unlimited**.
6. Введите максимальное количество разрешенных запусков приложения в поле **Activations.** Отметьте поле **Unlimited**, чтобы разрешить неограниченное количество запусков.
7. Добавьте описание программы в поле **Comments** (например, имя исполняемого файла или другие комментарии, облегчающие идентификацию приложения).

Комментарии, введенные в поле Comments утилиты NetHASP FAS Editor, сохраняются в файле конфигурации HaspEdit, а не в памяти ключа.

8. Нажмите **OK**. Приложение будет добавлено в список утилиты NetHASP FAS Editor.
9. Нажмите **Write**. На экране появится окно **Write HASP Memory**.

10. Сохраните изменения на ключ HASP4 Net. Вы можете:

- Нажать **Write**, чтобы сохранить все изменения в память.
- Отметить **Check Write Selected Programs Only**, чтобы сохранить в память только изменения, касающиеся выбранных приложений, затем нажать **Write**.

Все изменения в памяти выделяются на экране красным цветом. Нажатие кнопки **Write** сохраняет изменения в ключ HASP. Все данные, записанные на ключе HASP, отображаются на экране синим цветом.

Редактирование памяти HASP

С помощью утилиты HASPEdit вы можете считывать данные из памяти ключа и осуществлять запись в нее (сохранять пароли, имя покупателя, части программы или любые другие данные).

Обзор окон приложения HASP Memory Editor

Все окна HASP Memory Editor имеют схожий интерфейс. Используйте мышь, чтобы перемещаться от поля к полю в окне и кнопки для выполнения необходимых операций.

С помощью поля **Offset** вы можете увидеть номер позиции выбранного слова в памяти. Вы также можете ввести конкретный номер позиции в это поле для редактирования соответствующего слова.

Вы можете выбрать формат вводимых данных, используя кнопки **Hex** (шестнадцатеричная система) или **Decimal** (десятичная система) на левой панели окна. Если вы хотите ввести данные в формате ASCII, используйте правую панель. Вы можете перемещаться между панелями с помощью мышки. С помощью клавиш **PageUp** и **PageDown** вы можете перемещаться между страницами памяти ключей HASP4 M4, HASP4 Time и HASP4 Net.

Все изменения в памяти выделяются на экране красным цветом. Нажатие кнопки **Write** сохраняет изменения в ключ HASP. Все данные, записанные на ключе HASP, отображаются на экране синим цветом.

Поле **Location Description** указывает на положение слов в памяти HASP. В нем указывается, находится ли слово в области FAS или User. Если слово находится в области FAS, в этом поле указывается идентификационный номер программы (например, (P1) для программы 1, (P2) для программы 2 и т.д.),

которая использует это слово. Указывается, зарегистрировано ли слово или нет (т.е. такого слова нет в списке программ FAS). Если слово зарегистрировано, в кавычках указывается комментарий, который вы присвоили приложению.

Если курсор находится в области FAS и выделяет слово, ассоциированное с программой, защитные параметры которой сохранены в списке программ FAS, будьте внимательны, чтобы не перезаписать это слово какими-либо другими данными.

Редактирование памяти ключей HASP4 M1 and HASP4 M4

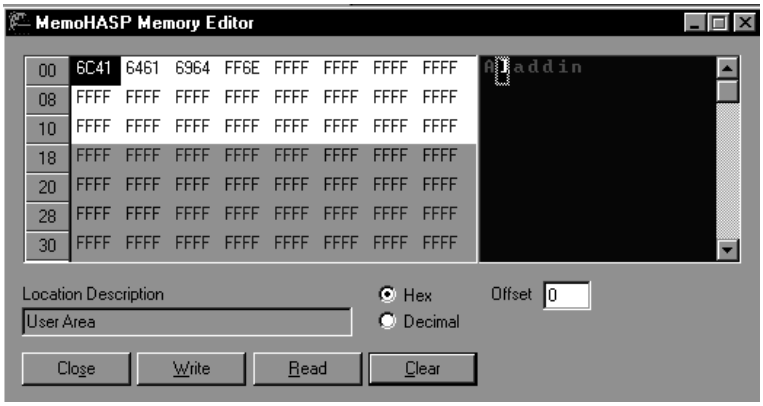
Вы можете использовать HaspEdit, чтобы редактировать память ключей HASP4 M1 (56 слов памяти) и HASP4 M4 (248 слов памяти). Первые 24 слова относятся к области пользователей и могут быть использованы для хранения любых данных. Остальные слова относятся к области FAS.

Память, которая не используется FAS, может быть отведена для записей любых пользовательских данных.

Для того чтобы отредактировать память ключей HASP4 M1 и HASP4 M4:

1. Выберите пункт **HASP Memory** из меню **HASP Tools** и нажмите **MemoHASP** или щелкните по пиктограмме **Memory**.

   На экране появится окно MemoHASP Memory Editor. В нем отображается содержимое памяти ключа.



2. Отредактируйте память.
3. Нажмите **Write**, чтобы сохранить изменения в память ключа.

Редактирование памяти ключа HASP4 Time

Вы можете использовать HaspEdit, чтобы редактировать память ключа HASP4 Time. Память ключа состоит из двух разделов: 16 байт отведены для FAS и 248 слов для области пользователей.

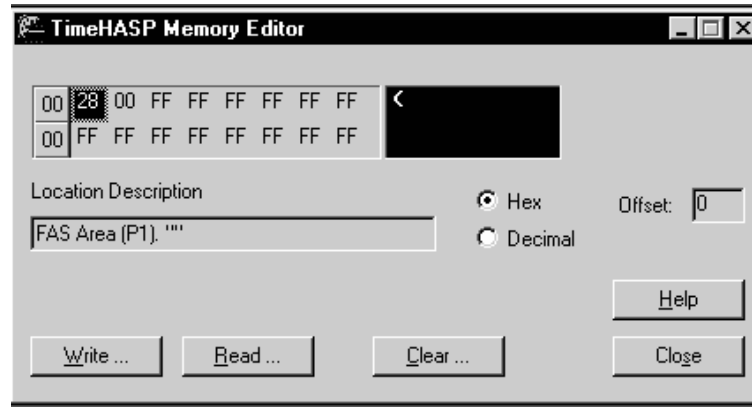
Все 248 слов области пользователей могут быть использованы для хранения любых данных.

Память, которая не используется FAS, может быть отведена для записей любых пользовательских данных.

Чтобы отредактировать память FAS ключа HASP4 Time:

1. Выберите пункт **HASP Memory** из меню **HASP Tools** и нажмите **TimeHASP** или щелкните по пиктограмме **Memory**.

   На экране появится окно TimeHASP Memory Editor. В нем отображается содержимое памяти ключа:

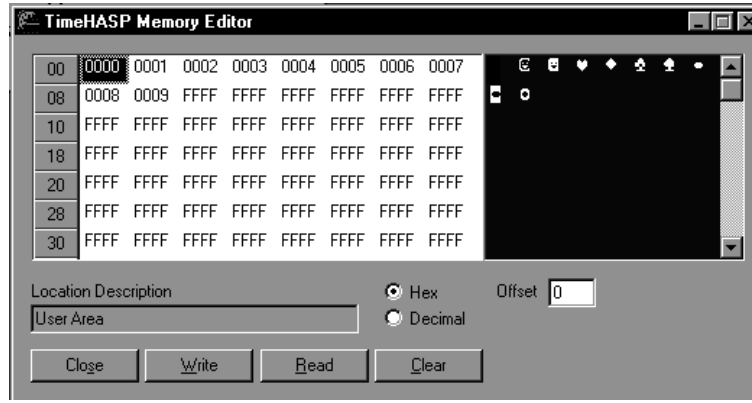


2. Отредактируйте память.
3. Нажмите **Write**, чтобы сохранить изменения в память ключа.

Чтобы отредактировать пользовательскую память ключа HASP4 Time:

1. Выберите пункт **HASP Memory** из меню **HASP Tools** и нажмите **MemoHASP**.

На экране появится окно HASP4 Time Memory Editor. В нем отображается содержимое памяти ключа.



2. Отредактируйте память.
3. Нажмите **Write**, чтобы сохранить изменения в память ключа.

Редактирование памяти ключа HASP4 Net

Вы можете использовать HaspEdit, чтобы редактировать память ключа HASP4 Net (248 слов).

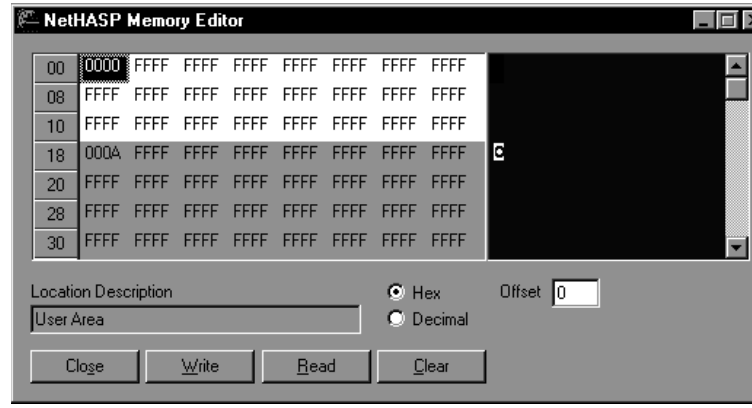
Первые 24 слова относятся к области пользователей и могут быть использованы для хранения любых данных. Остальные слова относятся к области FAS.

Память, которая не используется FAS, может быть отведена для записей любых пользовательских данных.

Чтобы отредактировать память FAS ключа HASP4 Net:

1. Выберите пункт **HASP Memory** из меню **HASP Tools** и нажмите **NetHASP** или щелкните по пиктограмме **Memory**.

   На экране появится окно NetHASP Memory Editor. В нем отображается содержимое памяти ключа:



2. Отредактируйте память.
3. Нажмите **Write**, чтобы сохранить изменения в память ключа HASP4 Net.

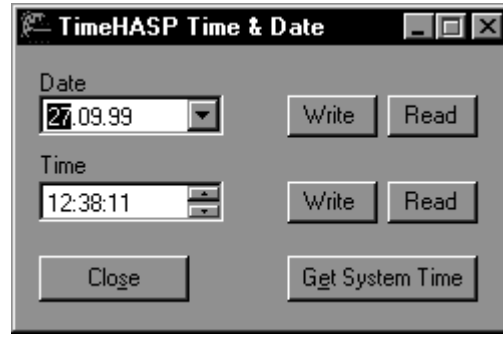
Установка часов HASP4 Time

С помощью утилиты HaspEdit вы можете установить часы реального времени HASP4 Time, а также просматривать и редактировать время часов HASP4 Time.

Чтобы установить часы реального времени HASP4 Time:

1. Выберите пункт **TimeHASP Clock** меню **HASP Tools**.

   На экране появится диалог **TimeHASP Time & Date Properties**:



2. Введите месяц, день и год в поле **Date**.
3. Выберите **Write Date**, чтобы сохранить установленную дату.
4. Введите время в формате hh (часы), mm (минуты) и ss (секунды) в поле **Time**. Чтобы синхронизировать часы с системными часами компьютера, нажмите на кнопку **Load System Time**, а затем нажмите **Write**.

   Часы HASP4 Time используют 24-х часовой формат данных (от 00:00:00 до 23:59:59).
5. Нажмите **Write Time**, чтобы сохранить время.
6. Нажмите **Close**, чтобы выйти из диалога **TimeHASP Time & Date Properties**.

Программирование нескольких ключей HASP

После того, как вы создали защиту вашего приложения, вы можете запрограммировать ключи HASP для подготовки их к передаче пользователям (совместно с поставляемым приложением). Не сцепляйте несколько ключей вместе в момент их программирования. Вместо этого сохраните один и тот же образ памяти в каждый ключ по очереди. HaspEdit позволяет вам запрограммировать несколько ключей с одинаковым набором параметров в Memory Editor и FAS Editor. Используйте один из методов, описанных ниже. Вы можете использовать опции:

- Program Key
- Create Programming Utility

Опция Program Key

Если вы используете опцию **Program Key,** вы программируете каждый ключ с помощью данных, указанных в текущей сессии утилиты HaspEdit.

Сохранение содержимого памяти HaspEdit в несколько ключей:

1. Выберите пункт **Program Key** меню **HASP Tools** или щелкните по пиктограмме **Prog** на панели управления.

   На экране появится окно **Multi-key Programming**.

2. Нажмите **Yes**, чтобы записать память HaspEdit в подключенный ключ HASP. В поле **Write Count** отображается количество запрограммированных на данный момент ключей.

3. Отключите ключ HASP от системы и подключите следующий.

4. Повторяйте шаги 2-3 для всех ключей, которые необходимо запрограммировать.

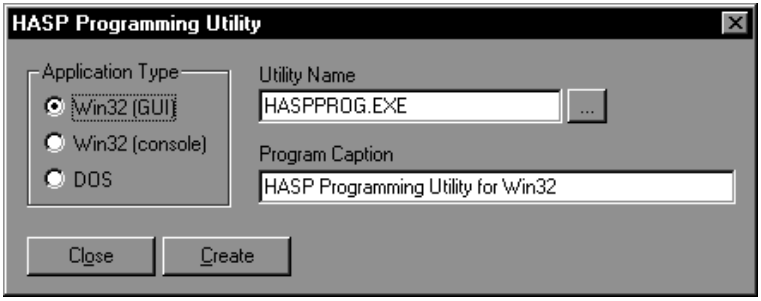
Создание утилиты программирования (Programming Utility)

При создании утилиты программирования вы создаете исполняемый файл для каждого ключа HASP с использованием параметров, которые установлены в текущей сессии утилиты HaspEdit. При создании файла убедитесь, что

окна Memory Editor и/или FAS Editor не закрыты (вы можете минимизировать их, если хотите), потому что они содержат те данные, которые используются утилитой HASP Programming, которая генерирует исполняемый файл.

Чтобы создать утилиту HASP Programming:

1. Выберите пункт **Create Programming Utility** в меню **HASP Tools**. Откроется окно **HASP:**



2. Выберите тип исполняемого файла в группе **Application Type**.
3. Введите имя исполняемого файла в поле **Utility Name** box.

   Имя по умолчанию - *Haspprog.exe*. Вы можете изменить имя и каталог, введя новое имя или нажав кнопку **Browse**.
4. В поле **Program Caption** (для приложения Win32 GUI) или поле **Program Banner** (для приложения Win32 console либо DOS) введите заголовок, который будет появляться во время запуска приложения.
5. Нажмите кнопку **Create**.

Если приложения с таким именем не существует, появится сообщение, подтверждающее создание приложения. Нажмите **OK**, чтобы вернуться в диалог **HASP Programming Utility**.

Если приложение с таким именем уже существует, появится диалог, запрашивающий подтверждение на выполнение операции. Нажмите **Yes**, чтобы заместить существующий файл, или **No** чтобы вернуться в диалог **HASP Programming Utility**.

Теперь вы можете запускать приложение и модифицировать память ключей HASP.

HASP Edit для Mac

В данном разделе поясняется использование приложения HASP Edit для Mac при выполнении основных операций. Для получения более подробной информации обращайтесь к файлу справки программы.

Общие сведения

Запуск HaspEdit:

1. Подключите ключ HASP к вашему компьютеру.

Не подключайте более одного ключа, поскольку вы можете осуществлять запись в память лишь одного ключа в одно и то же время. HASP Edit не различает разные ключи с одинаковым кодом разработчика и из-за этого может по ошибке разрушить данные, хранящиеся на ключе, если в системе установлено более одного ключа.

2. Запустите **HaspEdit.** На экране появится окно **Password.**
3. Введите пароль и подтвердите нажатием **OK**.

Пароли для ключей HASP demo вводятся автоматически по умолчанию. Чтобы изменить эту установку, выберите пункт **Preferences** меню **HASP Edit** и активируйте/деактивируйте эту установку.

Вы можете запускать HASP Edit даже тогда, когда к системе не подключен ключ HASP. Например, это полезно при редактировании шаблонов.

Установка свойств программы

Вы можете задать следующие установки для HASP Edit для Mac:

- Изменить значение, которое записывается в ячейки памяти при стирании памяти.
- Установить корневой каталог для файлов шаблонов.
- Показывать предупреждение, когда дата ключа HASP4 Time сильно отличается от системного времени.
- Включить опцию доступа к демонстрационным ключам с введением их паролей автоматически.
- Периодически искать подключенные ключи.

Чтобы выставить эти параметры, выберите пункт **Preferences** меню **HASP Edit**.

Подготовка к защите приложения

Используйте HaspEdit, чтобы подготовить приложение к защите методом шифрования данных для использования вашим приложением и получения уникального ID-номера HASP.

Шифрование данных

Эта функция дает вам возможность зашифровать данные, используя ключ HASP4. После шифрования вы сможете использовать зашифрованные данные в вашем приложении и расшифровывать их во время исполнения, используя соответствующие функции HASP API.

Шифрование данных:

1. Выберите **Encode/Decode** в дереве навигации.
2. Убедитесь, что нужный ключ подключен к системе.
3. Введите данные или загрузите файл, нажав **Load**.
4. Выберите формат выходных данных.

5. С помощью мыши или клавиатуры выберите секцию данных (не менее 8 байтов), которую вы хотите зашифровать. Нажмите **Encode** для того, чтобы зашифровать данные, затем нажмите **Save As**, чтобы сохранить зашифрованный файл на диске.

Вы можете редактировать файлы и выбирать секции для шифрования. В случае, если длина файла превышает 1024 байта, вы сможете зашифровать файл, но вы не сможете выделять и редактировать секции.

Получение ID-номера HASP

С помощью HASP Edit вы можете получить ID-номер ключей HASP4 с памятью. ID-номер уникален для каждого ключа. Вы можете вставить проверку на наличие определенного ID-номера в вашем приложении с использованием сервиса 6 (HaspID).

Чтобы проверить ID-номер ключа HASP с памятью, подключите ключ HASP к вашему компьютеру. Номер будет выведен в виде **Current HASP.**

Вы можете выбирать между тремя видами представления ключа:

- Шестнадцатеричный
- Десятичный со знаком
- Десятичный без знака

Программирование ключей HASP

Перед тем, как передать ключи пользователям (вместе с защищенным приложением), вам необходимо запрограммировать их.

Вы можете записать в память ключей HASP4 любые ваши данные и/или использовать память для работы системы FAS. Ключи, запрограммированные с помощью HASP Edit для Mac, могут быть использованы на всех поддерживаемых платформах.

Процесс программирования ключей состоит из двух основных шагов:

Сначала вы создаете мастер-ключ или шаблон, который служит базисом для программирования набора ключей. Может оказаться необходимым создание нескольких мастер-ключей в том случае, если вы используете ключи разных моделей.

После создания мастер-ключа вы можете начинать программирование набора ключей для распространения. Для того, чтобы запрограммировать несколько идентичных ключей, используйте утилиту для программирования нескольких ключей

Не подключайте более одного ключа к компьютеру, поскольку вы можете программировать только один ключ в одно и то же время.

Программирование одного ключа HASP

Программирование одного ключа HASP может быть необходимо в случае, если вы используете такой ключ в качестве мастер-ключа, либо тогда, когда вам необходимо подготовить специальный ключ для одного клиента.

Программирование отдельного ключа HASP

1. Подсоедините тот ключ HASP4, который вы хотите использовать в качестве мастер-ключа. В случае, если ключ не отображается в виде **Current HASP**, выберите пункт **Scan for Connected Key** в меню **Edit**.
2. Введите пароли для ключа.
3. Выберите поле **Memory** и отредактируйте память так, как вам это нужно.

   Данные будут отображаться красным цветом до тех пор, пока не будут записаны на ключ либо не сохранены в файл шаблона.
4. Нажмите **Write**, чтобы запрограммировать ключ. Вы можете также сохранить слепок памяти в файл шаблона (см. ниже).

Работа с шаблонами

Файл шаблона HASP Edit сохраняет параметры HASP и следующую дополнительную информацию:

- Тип ключа HASP
- Пароли ключа
- Образ памяти и параметры FAS
- Дополнительные комментарии о параметрах FAS
- Параметры HASP4 Time

Файл шаблона HASP Edit содержит в себе информацию о паролях HASP! Обеспечьте его безопасность.

Вы можете использовать файлы шаблона для программирования нескольких ключей HASP одного и того же типа с одним и тем же содержимым памяти.

Вы можете проделывать следующие операции с шаблонами:

- Создавать, переименовывать и удалять шаблоны
- Записывать вид шаблона в ключ, который подключен к компьютеру на момент записи
- Создавать ссылку на комментарий из шаблона на вид текущего ключа.

Создание шаблона:

1. Выберите пункт **New** меню **File**.
2. Выберите пункт **Save as** меню **File**.
3. Введите имя шаблона и папку, в которой он будет сохранен.

Создание шаблона с подключенного ключа:

1. Подключите ключ и выберите пункт **Scan for Connected Key** меню **Edit**, чтобы определить его.
2. Выберите пункт **Save as** меню **File**.
3. Введите имя шаблона и папку, в которой он будет сохранен.

Программирование одного ключа с помощью вида шаблона:

1. Подключите ключ.
2. Выберите шаблон.
3. Нажмите **Write Key**.

Использование инструмента

Multi Key Programming Tool

В случае, когда вам необходимо запрограммировать несколько ключей, вы сохраняете один и тот же слепок памяти в эти ключи (один за другим), используя память уже запрограммированного ключа или шаблон.

Чтобы использовать Multi Key Programming Tool с помощью мастер-ключа:

1. Подключите ключ, который будет использоваться в качестве мастер-ключа.
2. Выберите **Current HASP** в дереве навигации.
3. Выберите пункт **Multi Key Programming** меню **Tools**.
4. Отключите ключ.
5. Подключите новый ключ и подтвердите.
6. Повторяйте шаги 4-5 для каждого программируемого ключа.

Использование Multi Key Programming Tool с помощью шаблона:

1. Выберите шаблон, который хотите использовать.
2. Выберите пункт **Multi Key Programming** меню **Tools**.
3. Подключите новый ключ и подтвердите.
4. Отключите ключ.
5. Повторяйте шаги 3-4 для каждого программируемого ключа.

Использование FAS

Система FAS позволяет вам защищать несколько приложений с использованием одного и того же ключа HASP Memory, а также устанавливать параметры и ограничения для запуска каждого приложения.

С помощью HASP Edit вы можете установить параметры защиты для каждого приложения из списка программ FAS. Эти параметры зависят от используемой модели ключа.

Используйте FAS для:

- Ограничения количества запусков приложения с ключами HASP4 M1 (до 16 приложений) или HASP4 M4 (до 112 приложений).
- Установки даты окончания действия лицензии с ключом HASP4 Time.
- Ограничения количества станций, на которых одновременно может быть запущено приложение, а также количество запусков, с ключами HASP4 Net.

Просмотр параметров защиты FAS:

- Подключите ключ либо загрузите шаблон.
- Выберите пункт **As FAS List** или **Split** меню **View**.

Установка параметров защиты FAS:

- Подключите ключ либо загрузите шаблон. Затем нажмите FAS.
- Введите количество активаций (HASP4 M1, HASP4 M4, HASP4 Net), станций (HASP4 Net) и дату окончания действия лицензии (HASP4 Time).
- Чтобы установить неограниченное количество активаций, выберите **Unlimited** в меню **Edit** либо нажмите **U**.

Использование памяти с FAS

Программы FAS хранятся в области FAS-памяти ключа HASP. Область данных FAS начинается со слова со смещением в 24.

При использовании ключей HASP4 M1, HASP4 M4 или HASP4 Net каждая программа, защищенная с помощью FAS, требует 2 байта (2 слова) памяти HASP. Каждая пара слов в области памяти FAS зарезервирована под конкретный номер программы. Первые два слова – под программу с номером 1, вторая пара – для программы с номером 2 и т.д.

В случае использования ключей HASP4 Time каждая программа, защищенная с помощью FAS, требует 2 байта (одно слово) дополнительной памяти HASP Time.

Комментарии, введенные в список FAS, сохраняются в файле шаблона, а не в памяти ключа.

Часто задаваемые вопросы

Вопрос: Я использовал ключ HASP4 Net, чтобы защитить локальное приложение. Первые 24 слова в памяти HASP4 Net недостаточны для меня. Что я могу сделать?

Ответ: Оставшиеся 224 слова памяти HASP4 Net зарезервированы для сохранения параметров защиты 112 приложений, которые вы можете защитить с помощью одного ключа HASP4 Net. Так как вы хотите защитить лишь одно приложение, вы можете использовать часть этой памяти для ваших собственных нужд.

Свяжитесь с местным представителем HASP, чтобы получить точную информацию о расположении информации в памяти HASP4 Net. Затем используйте HASP4 Net API для сохранения данных в неиспользуемой части памяти FAS.

Вопрос: Является ли HASP Edit единственной утилитой, с помощью которой я могу программировать ключи HASP?

Ответ: Нет. У вас есть еще несколько возможностей для редактирования. Первая возможность – использовать утилиту автоматического программирования с предопределенными параметрами защиты, которые вы устанавливаете в утилите HASP Edit. Вторая возможность – написать собственную программу редактирования, которая использует HASP API и пишет данные в память ключа HASP.. И, наконец, вы можете использовать опцию «Program HASP», которая сохраняет параметры FAS. Эта опция доступна в утилите Win32 Envelope. Для получения более полной информации об использовании утилиты HASP Envelope вместе с FAS обратитесь к разделу «Сохранение параметров FAS» .

Вопрос: Могу ли я использовать ключи HASP, запрограммированные утилитами HASP Edit для Mac или HaspEdit для Windows на других платформах?

Ответ: Да. Ключи, запрограммированные с помощью одного из приложений HASP Edit, могут быть использованы на всех поддерживаемых платформах. В качестве альтернативы вы можете использовать HASP API для программирования ваших ключей HASP.

Поддержка конечных пользователей

Утилита Aladdin Diagnostic позволяет вашим клиентам собирать информацию об их системах и ключах HASP. Эта информация поможет вам и вашему клиенту решать проблемы с неправильной работой защищенного приложения.

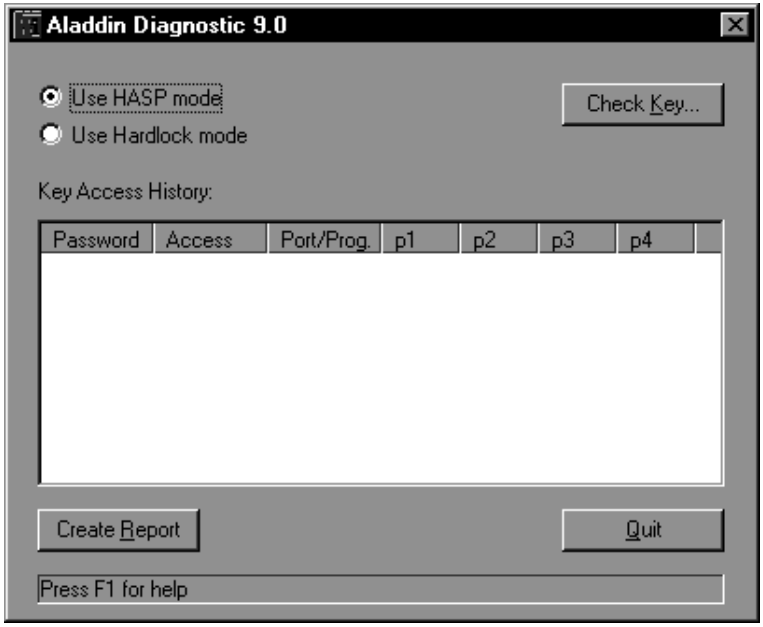
Ваши клиенты могут использовать утилиту Aladdin Diagnostic для:

- Проверки на наличие ключа HASP.
- Создания файла отчета, который содержит данные об устройствах Aladdin и другую необходимую информацию о системе.

Утилита Aladdin Diagnostic работает на следующих операционных системах: Windows 95/98/ME и Windows NT/2000/XP.

Рекомендуйте клиентам, запускающим утилиту Aladdin Diagnostic, активировать опцию **Use HASP mode**, чтобы проверять систему на наличие ключа HASP. В дополнение к этому они могут создавать отчеты с использованием соответствующей информации о системе.

Рис.7.1. Окно Aladdin Diagnostic



Создание отчетов

Ваши клиенты могут создавать файлы отчета, которые содержат данные об устройствах Aladdin и другую необходимую информацию о системе. В случае, если у ваших клиентов возникли проблемы с системой HASP, они могут сообщить об этом либо вам, либо локальному представителю службы технической поддержки HASP.

Чтобы создать отчет:

- Нажмите **Create Report** в главном окне утилиты **Aladdin Diagnostic**.

Файл отчета будет создан автоматически. Он будет сохранен в виде текстового файла в каталоге AKSDIAG. Имя этого файла по умолчанию – NDIAG32.TXT.

Файл открывается автоматически, позволяя вам распечатать его или сохранить под другим именем.

Диагностирование ключей HASP

Чтобы проверить ключ HASP, выберите режим **Use HASP mode** в окне Aladdin Diagnostic и нажмите **Check Key.** На экране появится диалог **Check HASP**.

Диалог **Check HASP** позволяет проверить наличие ключа HASP в системе.

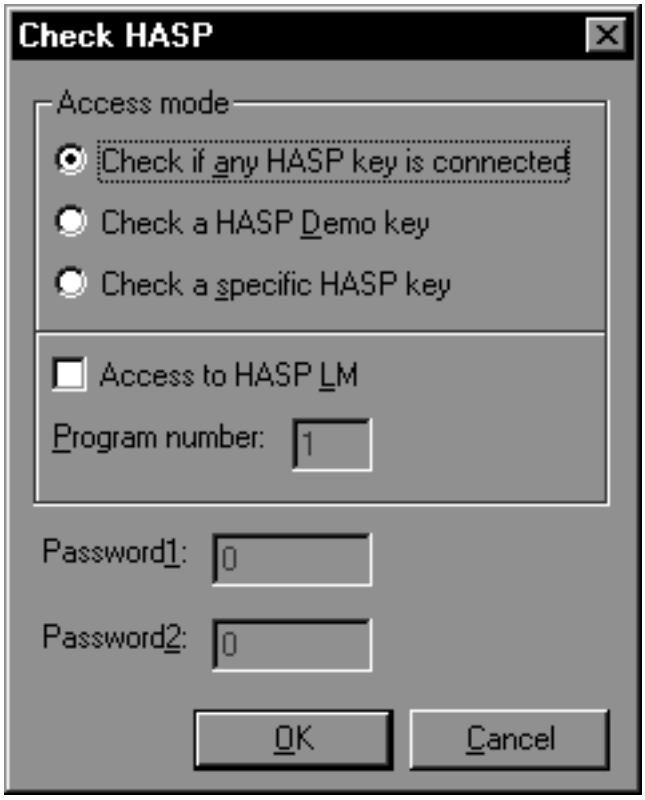
Проверка присутствия ключа HASP

В режиме **Access Mode** диалога **Check HASP** выберите одну из следующих опций:

- Проверить, подключен ли хотя бы один ключ
- Проверить наличие ключа HASP Demo
- Проверить наличие конкретного ключа HASP

Проверка удаленного ключа HASP

Вы можете получить доступ к удаленному ключу HASP, отметив поле **Access to HASP LM** в диалоге **Check HASP**. Введите номер программы, отчет о работе которой хотите получить, и нажмите **OK**.



Проверка подключен ли хотя бы один ключ HASP:

1. Выберите **Check if any HASP key is connected**.
2. Нажмите **OK**.

   На экране появится результат проверки.

1. Нажмите **OK**.

   Детали доступа отображаются в панели **Key Access History** окна утилиты Aladdin Diagnostic.

Чтобы проверить наличие ключа HASP Demo

1. Выберите **Check a HASP Demo key**.
2. Нажмите **OK**.

   На экране появится результат проверки.

3. Нажмите **OK**.

   Детали доступа отображаются в панели **Key Access History** окна утилиты Aladdin Diagnostic.

Чтобы проверить наличие конкретного ключа HASP:

1. Выберите **Check a specific HASP key**.
2. Введите пароли для ключа.
3. Нажмите **OK**.

   На экране появится результат проверки.

4. Нажмите **OK**.

   Детали доступа отображаются в панели **Key Access History** окна утилиты Aladdin Diagnostic.

Пароли HASP являются основой концепции защиты HASP, поэтому вы никогда не должны сообщать их вашим клиентам. Таким образом, последней опцией должны пользоваться только представители вашей компании, которые знают пароли.

Панель Key Access History

Панель Key Access History в главном окне показывает всю историю доступа к ключам HASP.

Информация в этой панели отсортирована по времени доступа. Наиболее поздние попытки доступа отражаются вверху окна. В таблице ниже приведено описание колонок Key Access History и допустимых значений в этих колонках:

Таблица 7.1. Панель Key Access History

Колонка	Значение	Описание
Password	Any	Производилась проверка, подключен ли хотя бы один ключ
	Demo	Производилась проверка на наличие ключа HASP Demo
	Custom	Производилась проверка на наличие конкретного ключа HASP
Access	Local	Ключ HASP обнаружен на локальной машине
	Remote	Ключ HASP обнаружен удаленно
Port/Prog.	Показывает номер порта, к которому подключен ключ HASP. В случае если ключ найден удаленно с использованием HASP LM, указывается также и номер программы. Если ключ не был найден, на экране отображается значение (n/a).	
p1-p4	Показывает статус API	

Часть 3
Использование HASP API

В этой части рассматриваются методы защиты и политики HASP API, а также приводится подробное описание сервисов HASP API.

В главе «Защита при помощи HASP API» на стр. 117 рассматривается процедура hasp() и приводится обзор сервисов HASP API.

В главе «Стратегии защиты» на стр. 133 рассматриваются способы повышения степени защиты.

В главе «Основные сервисы HASP» на стр. 143 описываются сервисы, применимые ко всем локальным ключам HASP.

В главе «Сервисы HASP4 Memory» на стр. 153 описываются сервисы для ключей HASP4 M1, HASP4 M4 и HASP4 Time.

В главе «Сервисы HASP4 Time» на стр. 163 описываются сервисы для ключей HASP4 Time.

В главе «Сервисы HASP4 Net» на стр. 177 описываются сервисы для ключей HASP4 Net.

В главе «Коды статуса HASP API» на стр. 209 приводятся объяснения кодов статуса, возвращаемых HASP.

Защита при помощи HASP API

HASP API является мощным методом защиты, возможности которого зависят от того, как его использовать. Чем более сложными являются вызовы API, тем лучшей будет обеспечиваемая HASP защита.

В этой и последующих главах, описывающих API, рассматривается процедура hasp() и ее сервисы.

API используется для повышения безопасности посредством посылки приложением вызовов HASP. Проверка присутствия HASP и реакция на результаты проверки может осуществляться в любой момент выполнения приложения. Также можно проверять данные, хранящиеся в памяти ключа HASP.

Способ выполнения таких проверок является основополагающим фактором обеспечения безопасности. Рекомендации по осуществлению проверок приводятся в разделе «Стратегии защиты» на стр. 133.

Перед началом реализации защиты при помощи HASP API рекомендуется проверить API-файлы для вашего компилятора. По каждому интерфейсу HASP есть образец приложения, демонстрирующий работу API.

Подготовка к использованию API

Для использования HASP API необходимо установить драйвер устройства HASP. Информация по установке драйверов HASP приводится в разделе «Установка HASP».

Перед включением вызовов API в приложение следует воспользоваться HASP Edit и/или API для решения описанных ниже задач.

Шифрование данных для использования в приложении

Для шифрования данных следует использовать сервис HaspEncodeData или утилиту HASP Edit. Дешифрование этих данных API будет возможно только в случае подключения соответствующего ключа HASP.

Редактирование памяти HASP

Если вы защищаете приложение при помощи ключа HASP с памятью и храните на нем секретную информацию, обратите внимание на то, какие данные хранятся по каждому адресу. API позволяет получать доступ к памяти и выполнять операции считывания и записи с определенным адресом.

Для осуществления записи в память можно использовать утилиту HASP Edit или сервисы WriteWord, WriteBlock и Write-Byte (относится только к ключам HASP4 Time).

Определение номера HASP ID

Ключи HASP с памятью обладают ID-номером. API позволяет проверить наличие определенного ключа HASP при помощи проверки его ID-номера.

Для определения ID-номера можно использовать утилиту HASP Edit.

Использование HASP API

После выполнения при помощи HASP Edit или API всех необходимых операций можно переходить непосредственно к реализации защиты приложения с использованием API, что осуществляется посредством включения вызовов процедуры hasp() в различных местах программного кода.

Основные стратегии использования

Для использования API:

1. Просмотрите программный код, который соответствует вашей среде разработки.
2. Добавьте вызов hasp() в ваш исходный код.
3. Для дешифровки и проверки важных данных, которые использует ваше приложение, используйте API.
4. На основании результатов шага 3 предусмотрите проверку на наличие ошибок и уведомление о них пользователя.
5. Повторите шаги 2-4 несколько раз, включая подобные проверки в несколько различных модулей.
6. Скомпилируйте программный код и свяжите его с объектным файлом HASP или поставляемой библиотекой.
7. Для создания дополнительного уровня защиты используйте утилиту HASP Envelope.

Использование процедуры hasp()

Процедура hasp() используется для включения в приложение защиты при помощи API. Данная процедура проверяет наличие ключа HASP, шифрует данные в реальном времени и позволяет получать доступ к памяти ключа HASP для осуществления операций чтения/записи.

Для ключей HASP4 Net и локальных ключей HASP процедура hasp() осуществляется по-разному.

Параметры для локальных ключей HASP

```
hasp (Service, SeedCode, PortNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

С процедурой hasp() используются следующие девять параметров.

Таблица 9.1. Параметры для локальных ключей HASP

Параметр	Описание
Service	Определяет выполняемую процедурой операцию.
SeedCode	Используется для обеспечения обратной совместимости.
PortNum	Определяет порт, на котором осуществляется поиск ключа HASP.
Password 1	Первый пароль для ключа HASP.
Password 2	Второй пароль для ключа HASP.
Par1 – Par4	Значения Par1 – Par4 изменяются в соответствии с сервисом

Параметры для ключей HASP4 Net

```
hasp (Service, SeedCode, ProgNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

С процедурой hasp() используются следующие девять параметров.

Таблица 9.2. Параметры для ключей HASP4 Net

Параметр	Описание
Service	Определяет выполняемую процедурой операцию.
SeedCode	Используется для обеспечения обратной совместимости.
ProgNum	Номер, присвоенный приложению, в памяти HASP4 Net.
Password 1	Первый пароль для ключа HASP.
Password 2	Второй пароль для ключа HASP.
Par1 – Par4	Значения Par1 – Par4 изменяются в соответствии с сервисом.

Определение порта

Для определения номера параллельного порта или порта USB можно использовать параметр PortNum. Приложения, защищенные HASP4 Net, осуществляют поиск ключа HASP4 Net на всех портах, поэтому если вы используете этот ключ, данный раздел можно пропустить.

Определение параллельного порта

Для определения параллельного порта, на котором будет осуществляться поиск ключа HASP, следует использовать параметр PortNum.

Если вы установите значение параметра PortNum равное 0, то приложение будет осуществлять автоматический поиск на всех параллельных портах в следующей последовательности: 378h, 278h, 3BCh. Автоматический поиск прекращается, когда будет найден ключ HASP.

Возможно осуществление поиска HASP на заданном параллельном порту или получение доступа к порту напрямую (в обход области данных BIOS). В приведенной ниже таблице указаны различные значения параметра Port-Num для осуществления поиска на определенном порту.

Таблица 9.3. Значения параметра PortNum и порты, на которых осуществляется поиск

PortNum	Порт, на котором осуществляется поиск
0	Автоматический поиск на всех портах
1	только LPT1
2	только LPT2
3	только LPT3
101	только 03BCh
102	только 0378h
103	только 0278h
201-255	Определенный HASP на порту USB

Определение USB–порта

Для определения номера USB-порта используется параметр PortNum. Для определения USB-порта зарезервированы целочисленные значения от 201 до 255. Каждый порт, к которому подсоединен ключ, обладает одним из 55 последовательных номеров.

Использование нумерации HASP USB можно проиллюстрировать следующим примером:

1. Вызовите процедуру hasp() при помощи сервиса IsHasp(). Значение P2 представляет собой число проверяемых портов на наличие определенного ключа.
2. Воспринимайте полученное в P2 число как переменную n, где n – число USB-портов, подлежащих проверке.
3. Вызовите процедуру hasp() при помощи сервиса HASP API, который осуществляет чтение памяти или получает ID-номер HASP, передавая номер первого USB-порта параметру PortNum. Также следует проверить данные, подтверждающие присутствие определенного ключа.

4. Если ключ на данном первом порту опознается как необходимый, сохраните номер порта USB. В противном случае следует осуществить переход к шагу 3 и проверить следующий номер порта USB. Возможно осуществление n подобных переходов.

Далее в вашем коде при обмене данными с этим определенным ключом будет использоваться его номер порта.

Проверка на критические ошибки

Для локальных ключей HASP существует два кода критических ошибок, которые необходимо проверить после каждого вызова процедуры hasp(). Эти коды возвращаются в p3:

- «HASP not found» (HASP не найден): ошибка –3. Если происходит эта ошибка, то необходимо попросить пользователя подсоединить (определенный) ключ HASP.
- «Port Busy» (Порт занят): ошибка –6. Необходимо вызвать ключ HASP еще раз через короткий промежуток времени или попросить пользователя подождать до тех пор, пока принтер или иное устройство закончит работу, после чего он сможет продолжить работу с приложением.

Также следует осуществить одну проверку при начале работы приложения с целью определить, установлен драйвер устройства HASP или нет. Это можно сделать, проверив ответы вызова hasp() на наличие ошибок –100, -101, -110 и –111. Если имеет место одна из этих ошибок, то необходимо установить драйвер и продолжить работу с программой.

Сервисы HASP

Информация о том, какой сервис следует использовать с определенной моделью HASP, содержится в таблице 9.4.

Таблица 9.4. Модели HASP и соответствующие им сервисы

Модель HASP	Соответствующие сервисы
HASP4 Std.	Основные сервисы HASP
HASP4 M1, M4	Основные сервисы HASP Сервисы HASP4 Memory
HASP4 Time	Основные сервисы HASP Сервисы HASP4 Memory Сервисы HASP4 Time
HASP4 Net	Сервисы HASP4 Net

Основные сервисы HASP

Основные сервисы HASP можно использовать с HASP4 Std., HASP4 M1, HASP4 M4 и HASP4 Time. Доступны следующие сервисы:

Таблица 9.5. Основные сервисы HASP

Сервис	Название	Операция
1	IsHasp	Проверяет подключение к компьютеру ключа HASP.
5	HaspStatus	Проверяет тип подключенного к компьютеру ключа HASP. Определяет, к какому порту подключен ключ. Проверяет объем памяти ключа HASP. Проверяет версию API.
60	HaspEncodeData	Шифрует посылаемые на подключенный ключ HASP4 данные. Используется совместно с сервисом HaspDecodeData для проверки того, подсоединен ли определенный ключ HASP4 к порту.
61	HaspDecodeData	Дешифрует посылаемые на подключенный ключ HASP4 данные. Используется совместно с сервисом HaspEncodeData для проверки того, подсоединен ли определенный ключ HASP4 к порту.

Сервисы HASP4 Memory

Сервисы HASP4 Memory можно использовать с HASP4 M1, HASP4 M4 и HASP4 Time (область 496 байт). Доступны следующие сервисы.

Таблица 9.6. Сервисы HASP4 Memory

Сервис	Название	Операция
3	ReadWord	Считывает из памяти HASP одно слово данных.
4	WriteWord	Записывает в память HASP одно слово данных.
6	HaspID	Получает ID-номер HASP.
50	ReadBlock	Считывает из памяти HASP блок данных.
51	WriteBlock	Записывает в память HASP блок данных.

Сервисы HASP4 Time

Сервисы HASP4 Time можно использовать с ключами HASP4 Time.

Сервисы 74-77 используются для получения доступа к 16-байтной памяти ключа HASP4 Time.

Сервисы 3, 4, 50 и 51 используются для получения доступа к 496-байтной памяти ключа HASP4 Time.

Доступны следующие сервисы HASP4 Time:

Таблица 9.7. Сервисы HASP4 Time

Сервис	Название	Операция
70	SetTime	Устанавливает часы ключа HASP4 Time на заданное время.
71	GetTime	Получает время от ключа HASP4 Time.
72	SetDate	Устанавливает часы ключа HASP4 Time на заданную дату.
73	GetDate	Получает дату от ключа HASP4 Time.
74	WriteByte	Записывает в память ключа HASP4 Time один байт.
75	ReadByte	Считывает из памяти ключа HASP4 Time один байт.
76	WriteBlock	Записывает в память ключа HASP4 Time блок данных.
77	ReadBlock	Считывает из памяти ключа HASP4 Time блок данных.
78	GetHaspID	Получает ID-номер ключа HASP4 Time.

Сервисы HASP4 Net

Сервисы HASP4 Net можно использовать исключительно с ключами HASP4 Net. Доступны следующие сервисы:

Таблица 9.8. Сервисы HASP4 Net

Сервис	Название	Операция
40	LastStatus	Проверяет статус последнего вызова. Этот сервис используется после каждого вызова процедуры hasp().
42	Login	Запрашивает разрешение HASP LM на активирование приложения. Если вы не используете сервис 85 или 96, данный сервис должен являться первым вызовом процедуры hasp().
43	Logout	Запрашивает HASP LM на завершение сессии.
44	ReadWord	Считывает одно слово данных из памяти HASP4 Net.
45	WriteWord	Записывает одно слово данных в память HASP4 Net.
46	HaspID	Получает ID-номер HASP4 Net.
48	IdleTime	Устанавливает максимальное время простоя станций.
52	ReadBlock	Считывает блок данных из памяти HASP4 Net.
53	WriteBlock	Записывает блок данных в память HASP4 Net.
85	SetConfigFilename	Устанавливает имя файла конфигурации HASP4 Net.
88	HaspEncodeData	Шифрует посылаемые на подключенный ключ HASP4 Net данные. Используется совместно с сервисом Hasp-DecodeData для проверки, подсоединен ли к порту определенный ключ HASP4 Net.
89	HaspDecodeData	Дешифрует посылаемые на подключенный ключ HASP4 Net данные. Используется совместно с сервисом HaspEncodeData для проверки того, подсоединен ли к порту определенный ключ HASP4.
96	SetServerByName	Устанавливает имя HASP LM, к которому приложение будет осуществлять дальнейшие подключения.

Включение локальной и сетевой защиты

API позволяет осуществлять защиту приложения как в сетевой среде, так и на отдельно стоящем компьютере.

Для реализации защиты при помощи API в сетевой среде или на отдельно стоящем компьютере:

1. Используйте сервисы HASP, HASP4 Time или сервисы HASP Memory, чтобы проверить, подсоединен ли соответствующий локальный ключ HASP к локальному порту компьютера.
2. Если соответствующий локальный ключ HASP не был найден, используйте сервис HASP4 Net для поиска ключа HASP4 Net.

Тестовая утилита HASP

При помощи тестовой утилиты HASP можно протестировать сервисы и правильность работы HASP API и HASP4 Net. Данную утилиту также можно использовать при проверке операций HASP.

Для Windows и Win32 файлом запуска утилиты является *haspdemo.exe*.

Тестовая утилита позволяет тестировать ключи HASP всех моделей. Файл *haspdemo.exe* является мульти-объектной программой, которую можно использовать для тестирования ключей с различными версиями HASP API. Для ее запуска выберите **HASP Test for Win16** или **Win32** в меню HASP CD. Путь к файлу выглядит следующим образом:

Utility\Haspdemo\Windows\Win16 или *Win32*

Для выполнения большинства функции данной тестовой утилиты необходим пароль. Пароли являются основой обеспечиваемой HASP безопасности, поэтому пароль не следует раскрывать клиентам. Файл *Haspdemo.exe* должен использоваться только тем персоналом вашей компании, который знает соответствующие пароли.

Часто задаваемые вопросы

Вопрос Как долго происходит проверка HASP?

Ответ Если ключ HASP подключен к отдельному компьютеру, вызов процедуры hasp() занимает 20 миллисекунд. Подключение к HASP4 Net занимает примерно две секунды, в зависимости от сетевого трафика. Такое время доступа к ключу позволяет осуществлять практически любое количество запросов HASP.

Вопрос Каким образом HASP предотвращает трассировку?

Ответ Более 60% процедур были разработаны с целью предотвратить трассировку кода защищаемого приложения. Естественно, мы не можем полностью раскрывать механизм обеспечения безопасности, но вот несколько примеров:

- прерывание работы отладочной программы;
- специальные ловушки для аппаратных отладчиков;
- затрудняющий трассирование самогенерирующийся код;
- временные ловушки.

Кроме того, частое обновление программного обеспечения позволяет использовать новые функции, повышающие степень защиты.

Вопрос

Используемый мной язык программирования или компилятор не совместим ни с каким языком программирования или компилятором, поддерживаемым HASP. Как мне защитить мое программное обеспечение?

Ответ

В этом случае мы можем предложить следующее:

- Попробуйте воспользоваться интерфейсом HASP, который поддерживает такие же типы объектных файлов, что и ваш компилятор.
- Попробуйте использовать DLL (для приложений Windows и Win32).
- Проконсультируйтесь с местным представителем Aladdin. Длинный список поддерживаемых HASP языков и компиляторов постоянно пополняется новыми интерфейсами.
- Используйте утилиту HASP Envelope.

Вопрос

Как можно защитить DLL при помощи HASP?

Ответ

Для защиты библиотек Win32 можно использовать утилиту Win32 Envelope; для защиты библиотек Win16 можно использовать утилиту Win16 Envelope; использование HASP API позволяет защитить библиотеки и Wind16 и Win32.

Существует два варианта защиты библиотек DLL при помощи API:

- Используйте объектные файлы, поставляемые вместе с программным обеспечением HASP, и создайте связь между ними и вашей библиотекой. Включите в ваше приложение вызовы процедуры hasp().
- Вызовите процедуру hasp() из вашей библиотеки. Процедура hasp() определяется в поставляемой нами библиотеке HASP DLL. В комплект поставки вашего приложения включите и собственную библиотеку, и HASP DLL.

Вопрос

Для проверки ID-номера HASP я использовал API. Иногда HaspID выдавал отрицательное число в IDLow. При вычислении ID-номера я получал совсем не тот номер, который показывается HASP Edit.

Ответ

Если IDLow показывает отрицательное число, то для вычисления ID-номера следует использовать следующую формулу:

ID-номер=65536+IDLow+65536*IDHigh

Или же можно использовать HASP Edit, что позволит выяснить короткие (16 бит) части слов IDLow и IDHigh. После этого можно будет напрямую сравнить их со значениями параметров IDLow и IDHigh, возвращаемых процедурой hasp(), без каких-либо вычислений ID-номера.

Вопрос

Где можно найти программу-пример, демонстрирующую использование API?

Ответ

Для большинства распространенных компиляторов такая программа есть на нашем компакт-диске. Например, для Visual C++ ее можно найти в директории win32api\msc.

Стратегии защиты

Система защиты HASP представляет собой комбинацию современных аппаратных и программных средств. Тем не менее, степень защищенности приложения зависит от того, насколько эффективный способ защиты при помощи HASP вы избрали. Вызовы процедуры hasp(), которые вы включаете в программный код, контролируют доступ к защищаемой программе. Поскольку аппаратные средства HASP практически невозможно взломать или воспроизвести, попытки взлома обычно принимают вид трассировки защитного кода и устранения защитных процедур.

Для получения максимальной степени защиты при помощи HASP следует использовать и защиту при помощи API, и при помощи Envelope. Каждый из этих методов обладает своими преимуществами, которые дополняются использованием другого метода.

Для повышения степени защищенности программного обеспечения при помощи API следует воспользоваться рекомендациями данной главы. В ней рассматриваются:

- атаки на схемы защиты программного обеспечения, от которых можно защититься, используя адекватные техники защиты;
- рекомендации по повышению степени защищенности приложений.

Если вам необходима помощь в защите приложений от возможных попыток взлома, мы можем предложить поддержку наших консультантов. Они могут помочь вам по широкому спектру вопросов, включая стратегии защиты и техники их применения. Для получения полной информации о консультационных услугах компании Aladdin обратитесь к вашему дилеру HASP.

Атаки на схемы защиты программного обеспечения

Существует два способа взлома защищенных приложений:

- Изменение вызовов защитной процедуры
- Изменение программного обеспечения производителя ключей

Для изменения вызовов защитной процедуры необходимо изменить защищенный исполняемый файл таким образом, чтобы он не посылал запросы ключу, не проверял результаты ответов ключа или не действовал в соответствии с определенными в программном коде результатами.

Этот тип взлома используется в том случае, если предполагается, что защита выполнена на низком уровне.

Для изменения программного обеспечения производителя необходимо изменить процедуры, которые отвечают за связь с аппаратным ключом. В результате подобного изменения процедуры возвращают ожидаемые результаты даже в том случае, если верный ключ не подключен к компьютеру.

Необходимо заметить, что оба этих метода сильно зависят от конкретного приложения и не могут использоваться в том же виде для взлома других приложений.

Рекомендации по защите приложений

В этом разделе рассматриваются способы защиты от возможных попыток взлома, описанных выше. Для повышения степени защищенности приложения следует использовать как можно больше стратегий защиты.

При разработке защиты помните, что защищаемое приложение в первую очередь предназначено для конечного пользователя, который может просто забыть подсоединить соответствующий ключ к компьютеру. Разрабатываемая схема защиты должна учитывать возможность подобных случаев. Продуманная стратегия защиты должна бороться не с пользователем, а с попытками взлома приложения.

Используйте множественные вызовы

Вставляйте в программный код приложения множественные вызовы процедуры hasp(), что способно существенно затруднить попытки взлома. Комплексные множественные вызовы значительно усложняют отслеживание и взлом схемы защиты.

Чем больше предусмотрено вызовов и проверок возвращаемого кода, тем более сложно их отследить и удалить. Вызовы следует включать в как можно большее количество разных мест программного кода.

Шифруйте внешние и внутренние данные

Зашифруйте некоторые данные, которое использует защищаемое приложение, и увяжите процесс дешифровки с присутствием ключа HASP.

Шифрование данных значительно повышает защищенность приложения. Используйте возможность HASP4 шифровать строки или двоичные данные, что в случае неверной дешифровки окажет влияние на работу приложения. Подобная мера приведет к созданию между защищаемым приложением и HASP связи, разорвать которую будет не просто. Задача удаления всех вызовов и проверок дополниться проблемой дешифровки данных. Нет необходимости шифровать все используемые приложением данные, но определенные ключевые части могут быть зашифрованы. В качестве объекта шифрования можно выбрать заголовки файлов, важные для расчетов постоянные или небольшие поля баз данных. Все, что способно повлиять

на основные функциональные возможности приложения, может являться потенциальным объектом шифрования. При этом необходимо предусмотреть извещение пользователя о том, что соответствующий ключ HASP не подключен, что предохранит ценные данные от повреждения.

Основной процесс шифрования простых наборов данных приводится ниже. Эти основные процедуры могут быть изменены или дополнены в соответствии с вашими требованиями.

1. Подсоедините ключ HASP4.

   Убедитесь в том, что ключ HASP4, который вы хотите использовать для защиты приложения, присоединен к компьютеру.

2. Зашифруйте данные

   Шифрование данных можно осуществить при помощи утилиты HASP Edit. Результатом операции шифрования будет являться двоичный файл. Кроме того, вы можете создать файл в формате Visual Basic или C, что позволит вам получать доступ к зашифрованным данным непосредственно в приложении.

3. Включите зашифрованные данные в приложение.

   Замените в приложении оригинальные незашифрованные данные на зашифрованные.

4. Осуществите дешифрование данных в приложении.

Для дешифрования зашифрованных данных при помощи ключа HASP4 следует использовать сервис 61 – HaspDecode-Data. После этого над данными можно совершать операции. Не забудьте включить проверку ошибок – это предупредит пользователя о том, что ключ HASP4 не подключен.

Следует помнить, что строки вида «HASP not found» («HASP не найден») нельзя шифровать ключами шифрования, связанными с HASP, поскольку эти строки показываются пользователю в том случае, если ключ HASP не присоединен к компьютеру. Для строк такого типа следует использовать любой другой ключ и метод шифрования.

Избегайте повторяющихся схем

Повторяющаяся в программном коде схема легко отслеживается. После того, как вашу схему распознали, ее будет легко найти и взломать в любой части программного кода.

Кроме того, если ваше приложение содержит большое количество исполняемых файлов и библиотек, следует использовать разные схемы защиты для этих файлов. Используйте как можно больше сервисов HASP, вызывая каждый из них повторно; применяйте различные схемы защиты при каждом вызове. Для защиты исполняемых файлов используйте HASP Envelope. В HASP Envelope применяется технология Multi-layer Envelope, которая реализует сочетания различных схем защиты для каждого исполняемого файла.

Отсутствие повторяющихся схем защиты осложнит попытку взлома. Поиск и взлом вызова HASP должен представлять собой каждый раз новую задачу.

Разделяйте шаги вызова

Проверка HASP подразумевает выполнение трех шагов:

- Вызов процедуры hasp().
- Оценка значений, полученных от защитной процедуры.
- Ответ в зависимости от полученных значений.

Для повышения безопасности можно поместить каждый из этих шагов в различные места программного кода приложения. Разделенные шаги гораздо сложнее отследить, чем последовательные. Последовательность действий на то, что HASP не присоединен, должна слегка задерживаться по времени, что будет являться полной неожиданностью для взломщика.

Например, можно увязать проверку на наличие присоединенного ключа HASP с выбором того или иного пункта меню. Позвольте пользователю некоторое время осуществлять действия, даже если ключ не присоединен. Сделайте так, чтобы сообщение «HASP не найден» появилось через некоторое время после операции, т.е. таким образом, чтобы связь между первоначальной проверкой наличия HASP и реакцией на ее результаты была неочевидной.

Шифруйте память HASP

В дополнение к защите, обеспечиваемой чипом ASIC, можно обезопасить память HASP при помощи ее шифрования.

Например, для шифрования памяти ключа HASP в качестве ключа шифрования можно использовать уникальный ID-номер устройства. Убедитесь в том, что в процессе работы память при помощи ID-номера расшифровывается.

Поскольку ключ шифрования каждого ключа HASP является уникальным, процесс дешифровки также будет уникальным. Подобный процесс является серьезным препятствием для копирования содержимого памяти одного ключа HASP на другой ключ, поскольку ID-номер HASP не может быть скопирован. Память «поддельного» ключа HASP дешифруется с использованием неверного ключа, что приводит к неверным результатам дешифровки.

Предусмотрите проверку контрольной суммы программного кода

Проверка контрольной суммы программного кода позволяет определить, вносились ли изменения в защищаемое приложение или нет.

Для осуществления простейшей проверки контрольной суммы:

1. Вычислите контрольную сумму.
2. Сравните ее с верным значением.
3. Если суммы не сходятся, предусмотрите появление сообщения об ошибке. В противном случае программа должна продолжать работать нормально.

   К сожалению, данная техника защиты подвержена нескольким типам попыток взлома:

- Программный код может быть изменен, а процедура проверки контрольной суммы устранена.
- Возможно программирование выдачи корректного результата проверки.

Для этого взломщики обычно определяют алгоритм расчета контрольной суммы и используют различные техники изменения программного кода для компенсации изменений количества байтов.

От подобной попытки взлома можно защититься, применив операцию XOR. Для этого следует использовать алгоритм CRC или иной алгоритм, принимающий во внимание последовательность байтов.

Другим методом защиты является отсутствие операции сравнения результата проверки контрольной суммы с рассчитанным заранее значением. Вместо этого можно использовать результаты проверки контрольной суммы для выполнения действия, которое приводит к возникновению ошибки, если был получен неверный результат расчета.

Например, можно сохранить результат контрольной суммы в виде переменной и использовать ее в последствии в качестве ключа для дешифровки определенных данных. Такой подход обладает преимуществом отложенной реакции. Кроме того, эталонная контрольная сумма не хранится в явном виде в самом защищаемом приложении.

Используйте функциональность программы в качестве реакции на отсутствие ключа HASP

Существует определенный набор действий, которые могут служить реакцией на отсутствие соответствующего устройства HASP. Самым простым действие является вывод на экран подсказки типа «HASP не найден». Однако это сразу же дает понять, что была осуществлена операция по проверке наличия ключа.

Вместо появления подобной подсказки можно предусмотреть последовательность действий, прерывающих нормальное функционирование защищаемого приложения до тех пор, пока соответствующий ключ HASP не будет подсоединен. Например, в случае, если ключ HASP не был найден, можно отключить клавиатуру, а включить ее только в том случае, если будет подключен верный ключ HASP. Незаконный пользователь может подумать, что сбой был вызван ошибкой в программе. Он может и не понять, что была осуществлена проверка HASP, а проблема возникла из-за отсутствия соответствующего ключа.

Однако, предусматривая подобные процедуры при работе приложения, не следует забывать о том, что законный пользователь может по ошибке забыть подключить свой ключ HASP.

Скрывайте пароли

Скрывайте пароли в защищаемом приложении при помощи:

- Их шифрования.
- Их хранения в различных местах программного кода.
- Сравнения различных мест хранения пароля в качестве проверки того, было ли осуществлено изменение программного кода.

Создавайте помехи

Вызовите процедуру hasp() и сделайте так, чтобы она передавала параметры с неверными значениями. Эти значения могут быть получены при помощи генератора случайных чисел, при помощи измерения времени, при получении каких-либо результатов промежуточных вычислений и т.д. Понятно, что такие вызовы процедуры hasp() не должны приводить к какому-либо значимому действию. Создание подобных помех способно создать дополнительные трудности при попытке взлома защищаемого приложения.

Используйте зависимые от HASP данные

При использовании хранящихся в памяти HASP данных перед тем, как продолжить работу, обычно проверяется правильность полученных значений. Однако такие проверки вынуждают вас включать реальные значения непосредственно в приложение. Вследствие этого значения могут быть взломаны.

Для предотвращения несанкционированного доступа к этим данным используйте их в приложении неявным образом. Если HASP подсоединен, то значение верно. В противном случае значение неверно и приводит к ошибке.

Например, вместо проверки считываемых из памяти HASP данных их можно использовать напрямую для выполнения перехода к определенной метке и осуществления операции, описанной в следующем примере псевдокода:

```
Begin

Переменной FLAG присваивается первоначальное
значение.
```

```
Вызовите процедуру hasp( ) при помощи сервиса
ReadWord.

FLAG присваивается значение данных, считанных
из памяти HASP (в данном примере – 100).

Goto FLAG

...

...

...

Label 100:
```

Далее следует выполнение операции, необходимой для нормальной работы программы.

В приведенном выше примере HASP присоединен, корректное значение 100 считывается из его памяти, а программа продолжает свое нормальное функционирование, перейдя к метке 100. Если HASP не подключен, то программа не осуществляет перехода к метке и не выполняет требуемую операцию.

Такая стратегия предотвращает использование оператора IF, что существенно затрудняет процесс трассировки.

Используйте HASP Envelope

HASP Envelope создает защитный щит вокруг вашего приложения. Защита Envelope выполняет шифрование и применяет антиотладочные функции, что существенно затрудняет взлом.

Если вы защищаете несколько исполняемых файлов, HASP Envelope будет использовать различные схемы защиты в каждом случае. Изменение схем защиты значительно затрудняет процесс взлома защищаемого приложения.

Изменяйте стратегии

Для поддержания высокого уровня защиты следует часто менять используемые схемы. Применяйте различные методы защиты в различных версиях защищаемого приложения. Регулярно обновляйте ваши инструменты защиты.

Компания Aladdin постоянно совершенствует предлагаемые средства защиты. Периодически посещайте наш сайт в Интернете и получайте самую свежую информацию о новых функциональных возможностях HASP. Кроме того, можно связаться с организацией, в которой вы приобрели ключи HASP, и получить последние обновленные версии программного обеспечения для этих ключей.

Всегда будьте на один шаг впереди тех, кто пытается взломать ваш программный продукт.

Основные сервисы HASP

В данной главе рассматриваются основные сервисы HASP API, которые могут применяться к стандартному HASP4, HASP4 M1 и HASP4 M4. Каждый сервис описывается подробно.

В таблице 11.1 приводятся параметры процедуры hasp() для каждого сервиса. Каждому рассматриваемому в таблице сервису соответствует две строки:

- Строка вызова, C, в которой приводятся параметры, передаваемые процедуре hasp().
- Строка возврата, R, в которой приводятся параметры, возвращаемые процедурой hasp().

Значения Par1, Par2, Par3 и Par4 изменяются в зависимости от сервиса.

Все параметры являются 16-разрядными в 16-разрядных приложениях, а в 32-разрядных приложениях – 32-разрядными.

Таблица 11.1. Основные сервисы и параметры HASP

Сервис (№)	SeedCode		PortNum	Password1	Password2	Par1	Par2	Par3	Par4
IsHasp (1)	C		PortNum						
	R					HASP найден	Порты, на которых осуществляется поиск	Статус	
HaspStatus (5)	C		PortNum	Password1	Password2				
	R					Объем памяти	Тип HASP	Реальный PortNum	Версия объекта HASP

Сервис (№)	SeedCode		PortNum	Password1	Password2	Par1	Par2	Par3	Par4
HaspEncodeData (60)	C		PortNum	Password1	Password2	0	Размер буфера	Сегмент буфера	Смещение буфера
	R							Статус	
HaspDecodeData	C		PortNum	Password1	Password2	0	Размер буфера	Сегмент буфера	Смещение буфера
	R							Статус	

Параметр «Сегмент буфера» применим только к 16-разрядным приложениям.

Сервис 1: IsHasp

Описание	Проверяет, подсоединен ли ключ HASP к компьютеру	
Для каких ключей предназначен	HASP Std., HASP4 M1, HASP4 M4, HASP4 Time	
Синтаксис	hasp (Service, SeedCode, PortNum, Password1,Password2,Par1, Par2, Par3, Par4)	
Используемые параметры	Сервис	1
	PortNum	Значение указывает порты для поиска ключа HASP (см. раздел «Определение порта»).
Возвращаемые значения	Par1	Значение показывает был ли найден HASP 0 – HASP не подсоединен к компьютеру 1 – HASP подсоединен к компьютеру
	Par2	Значение указывает порты для поиска ключа HASP (см. раздел «Определение порта») .
	Par3	Код, показывающий статус (см. раздел «Коды статуса HASP»).
Комментарии	Сервис 1 предназначен для определения того, подсоединен ли какой-либо ключ HASP к компьютеру. Сервис IsHasp всегда следует использовать с другими сервисами HASP API. Для определения того, подключен ли верный ключ HASP (с вашим кодом разработчика), следует использовать сервис 61, HaspDecodeData. Для выполнения простой проверки наличия ключа используйте сервис 5, HaspStatus.	

Сервис 5: HaspStatus

Описание

Проверяет тип ключа HASP, подсоединенного к компьютеру. Также проверяет, к какому порту подключен ключ.

Для каких ключей предназначен

HASP Std., HASP4 M1, HASP4 M4, HASP4 Time

Синтаксис

hasp (Service, SeedCode, PortNum, Password1,Password2,Par1, Par2, Par3, Par4)

Используемые параметры

Сервис	5
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP.
Password2	Второй пароль HASP.

Возвращаемые значения

Par1	Объем памяти 1 - HASP4 M1 4 – HASP4 M4 0 – другой ключ
Par2	Тип HASP 0 – HASP4 Std. 1 – HASP4 M1 или HASP4 M4 5 – HASP4 Time
Par3	Реальный номер порта (см. раздел «Определение порта»)
Par4	Объектная версия HASP – текущая версия API

Комментарии

- Для минимизации времени поиска используйте значение параметра ActualPortNum, полученное в Par3, и передайте его через параметр PortNum последующему вызову процедуры hasp().
- Если ключ HASP4 Net подключен к локальному порту, данный сервис опознает его как ключ HASP4 M4. Объем памяти, возвращаемой в Par1, равен 4, модель HASP, возвращаемая в Par2 – 1.
- Устаревшие ключи TimeHASP-1 в Par2 возвращают значение 3. Если при защите приложения используются эти ключи, то необходимо внести в него изменения с тем, чтобы они могли обрабатывать значения статуса 3 или 5.

Сервис 60: HaspEncodeData

Описание

Шифрует данные при помощи подключенного ключа HASP.

Для каких ключей предназначен

HASP Std., HASP4 M1, HASP4 M4, HASP4 Time

Синтаксис

hasp (Service, SeedCode, PortNum, Password1,Password2,Par1, Par2, Par3, Par4)

Используемые параметры

Сервис	60
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP.
Password2	Второй пароль HASP.
Par1	0 (зарезервировано)
Par2	Размер буфера. Размер шифруемого буфера в байтах (минимум 8 байт).
Par3	Сегмент буфера. Адрес сегмента буфера. Применим только в 16-разрядных приложениях.
Par4	Смещение буфера. Адрес смещения буфера.

Возвращаемые значения

Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP» на стр. 209)

Комментарии

Содержимое программного буфера шифруется при помощи подсоединенного ключа HASP.

Этот сервис может использоваться только с ключами модельного ряда HASP4. При подключении более ранней модели ключа HASP шифрование будет невозможно, а сервис возвратит код ошибки.

Сервис 61: HaspDecodeData

Описание

Дешифрует данные при помощи подключенного ключа HASP.

Для каких ключей предназначен

HASP Std., HASP4 M1, HASP4 M4, HASP4 Time

Синтаксис

hasp (Service, SeedCode, PortNum, Password1,Password2,Par1, Par2, Par3, Par4)

Используемые параметры

Сервис	61
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP.
Password2	Второй пароль HASP.
Par1	0 (зарезервировано)
Par2	Размер буфера. Размер дешифруемого буфера в байтах (минимум 8 байт).
Par3	Сегмент буфера. Адрес сегмента буфера. Применим только в 16-разрядных приложениях.
Par4	Смещение буфера. Адрес смещения буфера.

Возвращаемые значения

Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP» на стр. 209).

Комментарии

Содержимое программного буфера дешифруется при помощи подсоединенного ключа HASP.

Этот сервис может использоваться только с ключами модельного ряда HASP4. При подключении более ранней модели ключа HASP шифрование будет невозможно, а сервис возвратит код ошибки.

Сервисы HASP4 Memory

В данной главе рассматриваются сервисы HASP API, которые могут применяться к HASP4 M1, HASP4 M4 и HASP4 Time. Каждый сервис описывается подробно.

В таблице 12.1 на стр. 154 приводятся параметры процедуры hasp() для каждого сервиса. Каждому рассматриваемому в таблице сервису соответствует две строки:

- Строка вызова, C, в которой приводятся параметры, передаваемые процедуре hasp().
- Строка возврата, R, в которой приводятся параметры, возвращаемые процедурой hasp().

Значения Par1, Par2, Par3 и Par4 изменяются в зависимости от сервиса.

Все параметры являются 16-разрядными в 16- разрядных приложениях, а в 32-разрядных приложениях – 32-разрядными.

Вследствие того, что компьютеры Macintosh работают с форматом Big Endian, а системы на базе Intel – в формате Little Endian, значения ReadWord и WriteWord на Macintosh будут отличаться от тех же значений в системах на базе Intel.

Этот факт следует принимать во внимание при работе данных сервисов. В качестве альтернативы можно использовать сервисы ReadBlock и WriteBlock, которые оперируют с endian независимо.

Таблица 12.1. Сервисы и параметры HASP4 Memory

Сервис (№)		SeedCode	PortNum	Password1	Password2	Par1	Par2	Par3	Par4
ReadWord (3)	C		PortNum	Password1	Password2	Адрес			
	R						Данные	Статус	
WriteWord (4)	C		PortNum	Password1	Password2	Адрес	Данные		
	R							Статус	
HaspID (6)	C		PortNum	Password1	Password2				
	R					IDLow	IDHigh	Статус	

Сервис (№)		SeedCode	PortNum	Password1	Password2	Par1	Par2	Par3	Par4
ReadBlock (50)	C		PortNum	Password1	Password2	Начальный адрес	Длина блока	Сегмент буфера	Смещение буфера
	R							Статус	
WriteBlock (51)	C		PortNum	Password1	Password2	Начальный адрес	Длина блока	Сегмент буфера	Смещение буфера
	R							Статус	

Параметр «Сегмент буфера» применим только к 16- разрядным приложениям.

Сервис 3: ReadWord

Описание

Считывает одно слово данных из памяти HASP.

Для каких ключей предназначен

HASP4 M1, HASP4 M4, HASP4 Time

Синтаксис

```
hasp (Service, SeedCode, PortNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	3
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP.
Password2	Второй пароль HASP.
Par1	Адрес, т.е. тот адрес памяти HASP, из которого вы хотите осуществить считывание: 0-55 – HASP4 M1 0-247 - HASP4 M4 0-247 – HASP Time

Возвращаемые значения

Par2	Данные. Одно слово данных, считанных из памяти HASP.
Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP» на стр. 209).

Сервис 4: WriteWord

Описание

Записывает одно слово данных в память HASP.

Для каких ключей предназначен

HASP4 M1, HASP4 M4, HASP4 Time

Синтаксис

```
hasp (Service, SeedCode, PortNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	4
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP.
Password2	Второй пароль HASP.
Par1	Адрес, т.е. тот адрес памяти HASP, из которого вы хотите осуществить считывание: 0-55 – HASP4 M1 0-247 - HASP4 M4 0-247 – HASP Time

Возвращаемые значения

Par2	Данные. Одно слово данных.
Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP» на стр. 209).

Сервис 6: HaspID

Описание

Определяет ID-номер HASP

Для каких ключей предназначен

HASP4 M1, HASP4 M4, HASP4 Time

Синтаксис

```
hasp (Service, SeedCode, PortNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	6
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP.
Password2	Второй пароль HASP.

Возвращаемые значения

Par1	IDLow – низкое (наименее важное) слово ID-номера.
Par2	IDHigh – высокое (наиболее важное) слово ID-номера.
Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP» на стр. 209).

Комментарии

ID-номер представляет собой длинную целочисленную величину (32 бита). Если IDLow и IDHigh беззнаковые, он вычисляется следующим образом:

ID-номер=IDLow+65536*IDHigh

Если IDLow и IDHigh знаковые, то следует компенсировать отрицательное значение прибавлением 65535.

Сервис 50: ReadBlock

Описание

Считывает один блок данных из памяти HASP.

Для каких ключей предназначен

HASP4 M1, HASP4 M4, HASP4 Time

Синтаксис

```
hasp (Service, SeedCode, PortNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	50
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP.
Password2	Второй пароль HASP.
Par1	Начальный адрес, т.е. тот адрес памяти HASP, с которого начинается считывание: 0-55 – HASP4 M1 0-247 - HASP4 M4 0-247 – HASP Time
Par2	Длина блока – размер блока, измеряемый в словах.
Par3	Сегмент блока – адрес сегмента буфера. Применим только к 16-разрядным приложениям.
Par4	Смещение буфера – адрес смещения программного буфера (переменная). Размер буфера должен быть не меньше размера блока.

Возвращаемые значения	Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP» на стр. 209).
Результат	Содержимое памяти HASP считывается в программный буфер.	

Сервис 51: WriteBlock

Описание

Записывает один блок данных в память HASP.

Для каких ключей предназначен

HASP4 M1, HASP4 M4, HASP4 Time

Синтаксис

```
hasp (Service, SeedCode, PortNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	51
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP.
Password2	Второй пароль HASP.
Par1	Начальный адрес, т.е. тот адрес памяти HASP, с которого начинается запись блока: 0-55 – HASP4 M1 0-247 - HASP4 M4 0-247 – HASP Time
Par2	Длина блока – размер блока, измеряемый в словах.
Par3	Сегмент блока – адрес сегмента буфера. Применим только к 16-разрядным приложениям.

	Par4	Смещение буфера – адрес смещения программного буфера (переменная). Размер буфера должен быть не меньше размера блока.
Возвращаемые значения	Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP» на стр. 209)
Результат	Содержимое программного буфера записывается в память HASP.	

Сервисы HASP4 Time

В данной главе рассматриваются сервисы HASP API, которые могут применяться к HASP4 M1, HASP4 M4 и HASP4 Time. Каждый сервис описывается подробно.

Срок службы батарейки HASP4 Time составляет 3-5 лет. При разработке защиты при помощи API рекомендуется предусмотреть вызовы, которые будут проверять нормальное функционирование часов, а при их остановке – совершать соответствующие ситуации действия.

В таблице 13.1 приводятся параметры процедуры hasp() для каждого сервиса. Каждому рассматриваемому в таблице сервису соответствует две строки:

- Строка вызова, C, в которой приводятся параметры, передаваемые процедуре hasp().
- Строка возврата, R, в которой приводятся параметры, возвращаемые процедурой hasp().

Значения Par1, Par2, Par3 и Par4 изменяются в зависимости от сервиса.

Все параметры являются 16-разрядными в 16-разрядных приложениях, а в 32-разрядных приложениях – 32-разрядными.

Вследствие того, что компьютеры Macintosh работают с форматом Big Endian, а системы на базе Intel – в формате Little Endian, значения ReadWord и WriteWord на Macintosh будут отличаться от тех же значений в системах на базе Intel.

Этот факт следует принимать во внимание при работе данных сервисов. В качестве альтернативы можно использовать сервисы ReadBlock и WriteBlock, которые оперируют с endian независимо.

Таблица 13.1 Сервисы и параметры HASP4 Time

Сервис (№)		**SeedCode**	**PortNum**	**Password1**	**Password2**	**Par1**	**Par2**	**Par3**	**Par4**
SetTime (70)	C		PortNum	Password1	Password2	Секунда	Минута		Час
	R							Статус	
GetTime (71)	C		PortNum	Password1	Password2				
	R					Секунда	Минута	Статус	Час

Сервис (№)		SeedCode	PortNum	Password1	Password2	Par1	Par2	Par3	Par4
SetDate (72)	C		PortNum	Password1	Password2	День	Месяц		Год
	R							Статус	
GetDate (73)	C		PortNum	Password1	Password2				
	R					День	Месяц	Статус	Год
WriteByte (74)	C		PortNum	Password1	Password2	Адрес	Данные		
	R							Статус	
ReadByte (75)	C		PortNum	Password1	Password2	Адрес			
	R						Данные	Статус	

Сервис (№)		SeedCode	PortNum	Password1	Password2	Par1	Par2	Par3	Par4
WriteBlock (76)	C		PortNum	Password1	Password2	Начальный адрес	Длина блока	Сегмент буфера	Смещение буфера
	R							Статус	
ReadBlock (77)	C		PortNum	Password1	Password2	Начальный адрес	Длина блока	Сегмент буфера	Смещение буфера
	R							Статус	
HaspID (78)	C		PortNum	Password1	Password2				
	R					IDLow	IDHigh	Статус	

Параметр «Сегмент буфера» применим только к 16-разрядным приложениям.

Сервис 70: SetTime

Описание

Устанавливает время на часах HASP4 Time.

Для каких ключей предназначен

HASP4 Time

Синтаксис

```
hasp (Service, SeedCode, PortNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	70
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP4 Time.
Password2	Второй пароль HASP4 Time.
Par1	Секунда. Количество секунд, которые вы устанавливаете (00-59).
Par2	Минута. Количество минут, которое вы устанавливаете (00-59).
Par4	Час. Количество часов, которое вы устанавливаете (00-23).

Возвращаемые значения

Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP» на стр. 209).

Сервис 71: GetTime

Описание

Получает время от часов HASP4 Time.

Для каких ключей предназначен

HASP4 Time

Синтаксис

```
hasp (Service, SeedCode, PortNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис

PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP4 Time.
Password2	Второй пароль HASP4 Time.

Возвращаемые значения

Par1	Секунда. Количество секунд на часах HASP4 Time (00-59).
Par2	Минута. Количество минут на часах HASP4 Time (00-59).
Par4	Час. Количество часов на часах HASP4 Time (00-23).

Сервис 72: SetDate

Описание

Устанавливает дату на часах HASP4 Time.

Для каких ключей предназначен

HASP4 Time

Синтаксис

```
hasp (Service, SeedCode, PortNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	72
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP4 Time.
Password2	Второй пароль HASP4 Time.
Par1	День. Количество дней, которое вы устанавливаете (1-31).
Par2	Месяц. Количество месяцев, которое вы устанавливаете (1-12).
Par4	Год. Количество лет, которое вы устанавливаете (0-99).

Возвращаемые значения

Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP» на стр. 209).

Комментарии

Значение «год» может находиться в диапазоне 0-99. При этом значения 92-99 отражают 1992-1999 года, а 00-91 – 2000-2091 года.

Сервис 73: GetDate

Описание

Получает дату от часов HASP4 Time.

Для каких ключей предназначен

HASP4 Time

Синтаксис

```
hasp (Service, SeedCode, PortNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	73
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP4 Time.
Password2	Второй пароль HASP4 Time.

Возвращаемые значения

Par1	День. Количество дней, считанных с часов HASP4 Time (1-31).
Par2	Месяц. Количество месяцев, считанных с часов HASP4 Time (1-12).
Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP» на стр. 209).
Par4	Год. Количество лет, считанных с часов HASP4 Time (0-99).

Комментарии

Значение «год» может находиться в диапазоне 0-99. При этом значения 92-99 отражают 1992-1999 года, а 00-91 – 2000-2091 года.

Сервис 74: WriteByte

Описание

Записывает один байт данных в память HASP4 Time.

Для каких ключей предназначен

HASP4 Time

Синтаксис

```
hasp (Service, SeedCode, PortNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	74
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP4 Time.
Password2	Второй пароль HASP4 Time.
Par1	Адрес, т.е. адрес памяти HASP4 Time, в который вы хотите осуществить запись.
Par2	Данные – один байт данных.

Возвращаемые значения

Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP» на стр. 209).

Комментарии

Данный сервис позволяет осуществлять запись в 16-байтную память HASP4 Time. Для записи в память на 248 слов следует использовать Сервис 4.

Сервис 75: ReadByte

Описание

Считывает один байт данных из памяти HASP4 Time.

Для каких ключей предназначен

HASP4 Time

Синтаксис

```
hasp (Service, SeedCode, PortNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	75
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP4 Time.
Password2	Второй пароль HASP4 Time.
Par1	Адрес, т.е. адрес памяти HASP4 Time, из которого вы хотите осуществить чтение.

Возвращаемые значения

Par2	Данные – один байт данных, считываемых из памяти HASP4 Time.
Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP» на стр. 209).

Комментарии

Данный сервис позволяет осуществлять запись в 16-байтную память HASP4 Time. Для записи в память на 248 слов следует использовать Сервис 3.

Сервис 76: WriteBlock

Описание

Записывает один блок данных в память HASP4 Time.

Для каких ключей предназначен

HASP4 Time

Синтаксис

```
hasp (Service, SeedCode, PortNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	76
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP4 Time.
Password2	Второй пароль HASP4 Time.
Par1	Начальный адрес – определяет начальный адрес памяти HASP4 Time для записи блока (0-15).
Par2	Длина блока – размер блока данных в байтах (максимальное значение – 16).
Par3	Сегмент буфера – адрес сегмента буфера. Применим только к 16-разрядным приложениям.
Par4	Смещение буфера – адрес смещения программного буфера (переменная). Размер буфера должен быть не меньше размера блока.

Возвращаемые значения

Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP» на стр. 209).

Комментарии

Данный сервис записывает блок данных в 16-битную память HASP4 Time. Для записи блока данных в память на 248 слов следует использовать сервис 51.

Сервис 77: ReadBlock

Описание

Считывает один блок данных из памяти HASP4 Time.

Для каких ключей предназначен

HASP4 Time

Синтаксис

```
hasp (Service, SeedCode, PortNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	77
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP4 Time.
Password2	Второй пароль HASP4 Time.
Par1	Начальный адрес – определяет начальный адрес памяти HASP4 Time для считывания блока (0-15).
Par2	Длина блока – размер блока данных в байтах (максимальное значение – 16).
Par3	Сегмент буфера – адрес сегмента буфера. Применим только к 16-разрядным приложениям.
Par4	Смещение буфера – адрес смещения программного буфера (переменная). Размер буфера должен быть не меньше размера блока.

Возвращаемые значения

Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP» на стр. 209).

Комментарии

Данный сервис считывает блок данных из 16-битной память HASP4 Time. Для считывания блока данных из памяти на 248 слов следует использовать сервис 50.

Сервис 78: HaspID

Описание

Определяет ID-номер HASP

Для каких ключей предназначен

HASP4 Time

Синтаксис

```
hasp (Service, SeedCode, PortNum,
Password1,Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	78
PortNum	Значение указывает порт, на котором осуществляется поиск HASP.
Password1	Первый пароль HASP4 Time.
Password2	Второй пароль HASP4 Time.

Возвращаемые значения

Par1	IDLow – низкое (наименее важное) слово ID-номера.
Par2	IDHigh – высокое (наиболее важное) слово ID-номера.
Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP» на стр. 209).

Комментарии

ID-номер представляет собой длинную целочисленную величину (32 бита). Если IDLow и IDHigh беззнаковые, он вычисляется следующим образом:

ID-номер=IDLow+65536*IDHigh

Если IDLow и IDHigh знаковые, то следует компенсировать отрицательное значение прибавлением 65535.

Сервисы HASP4 Net

В данной главе рассматриваются сервисы HASP API, которые могут применяться к HASP4 Net. Каждый сервис описывается подробно.

В таблице 14.1 приводятся параметры процедуры hasp() для каждого сервиса. Каждому рассматриваемому в таблице сервису соответствует две строки:

- Строка вызова, C, в которой приводятся параметры, передаваемые процедуре hasp().
- Строка возврата, R, в которой приводятся параметры, возвращаемые процедурой hasp().

Значения Par1, Par2, Par3 и Par4 изменяются в зависимости от сервиса.

Все параметры являются 16-разрядными в 16-разрядных приложениях, а в 32-разрядных приложениях – 32-разрядными.

Использование сервисов HASP4 Net

API ключей HASP4 Net требует, чтобы некоторые сервисы вызывались следующим образом:

1. Можно вызвать процедуру hasp() при помощи SetCofigFilename, а затем при помощи сервиса LastStatus.
2. Можно вызвать процедуру hasp() при помощи SetServerByName, а затем при помощи сервиса LastStatus.
3. Вызовите процедуру hasp() сервисом Login, а затем сервисом LastStatus.
4. После применения сервиса Login можно вызывать любой другой сервис HASP4 Net. После каждого вызова сервиса следует вызывать процедуру hasp() при помощи сервиса LastStatus.

При использовании протокола TCP/IP 16-разрядное приложение Windows, осуществляющее вызов процедуры hasp(), получает контроль автоматически в процессе обработки процедуры. Приложение получает контроль в рамках очереди обработки сообщений скорее, чем при помощи инструкций, следующих за вызовом процедуры hasp().

Не осуществляйте вызов процедуры hasp() повторно до тех пор, пока не будет получен контроль при помощи инструкций, следующих за вызовом – это может привести к зависанию приложения. Во избежание подобной ситуации следует использовать флаги, предотвращающие выполнение новых вызовов процедуры hasp() до тех пор, пока не будут обработаны предыдущие вызовы.

Таблица 14.1. Сервисы и параметры HASP4 Net

Сервис (№)		SeedCode	PortNum	Password1	Password2	Par1	Par2	Par3	Par4
LastStatus (40)	C								
	R					Сетевой статус	Системная ошибка	Код предупреждения	
Login (42)	C	SeedCode	ProgNum	Password1	Password2				
	R								
Logout (43)	C		ProgNum	Password1	Password2				
	R							Статус	
ReadWord (44)	C		ProgNum	Password1	Password2	Адрес			
	R						Данные	Статус	

Сервис (№)		SeedCode	PortNum	Password1	Password2	Par1	Par2	Par3	Par4
WriteWord (45)	C		ProgNum	Password1	Password2	Адрес			
	R							Статус	
HaspID (46)	C		ProgNum	Password1	Password2				
	R					IDLow	IDHigh	Статус	
IdleTime (48)	C	IdleTime	ProgNum	Password1	Password2				
	R								
ReadBlock (52)	C		ProgNum	Password1	Password2	Начальный адрес	Длина блока	(Сегмент буфера)	Смещение буфера
	R							Статус	

Сервис (№)		SeedCode	PortNum	Password1	Password2	Par1	Par2	Par3	Par4
WriteBlock (53)	C		ProgNum	Password1	Password2	Начальный адрес	Длина блока	(Сегмент буфера)	Смещение буфера
	R							Статус	
SetConfigFile-name	C						Размер буфера	(Сегмент буфера)	Смещение буфера
	R								
HaspEncode-Data (88)	C		ProgNum	Password1	Password2		Размер буфера	(Сегмент буфера)	Смещение буфера
	R							Статус	

Сервис (№)		SeedCode	PortNum	Password1	Password2	Par1	Par2	Par3	Par4
HaspDecode-Data (89)	C		ProgNum	Password1	Password2		Размер буфера	(Сегмент буфера)	Смещение буфера
	R							Статус	
SetServer-ByName (96)	C						Размер буфера	(Сегмент буфера)	Смещение буфера
	R								
HaspQueryLi-cense (104)	C		ProgNum	Password1	Password2				
	R					Активная лицензия	Максимальное число лицензий	Тип ключа	Число оставшихся активаций

Параметр «Сегмент буфера» применим только к 16-разрядным приложениям.

Сервис 40: LastStatus

Описание

Проверяет статус предыдущего вызова процедуры hasp(). Следует осуществлять вызов сервиса LastStatus каждый раз после того, как был вызван любой другой сервис HASP4 Net.

Для каких ключей предназначен

HASP4 Net

Синтаксис

```
hasp (Service, SeedCode, ProgNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис 40

Возвращаемые значения

Par1 Сетевой статус – код, показывающий статус предыдущего вызова процедуры hasp(). Если последний вызов был успешным, возвращается 0. В случае обратного, возвращается код ошибки (см. таблицу 15.3 «Коды статуса HASP4 Net»).

Par2 Системная ошибка – код ошибки в зависимости от ситуации. Например:

- При коммуникационной ошибке NetBIOS возвращается особый код ошибки.
- При ошибке в файле конфигурации HASP4 Net возвращается номер строки файла, где произошла ошибка.

Par3 Предупреждение – код предупреждения (см. таблицу «Коды предупреждений HASP4 Net»).

Комментарии

В случае возникновения ошибки ваше приложение должно показывать Сетевой статус, Системную ошибку и Предупреждение.

Сервис 42: Login

Описание

Осуществляет подключение HASP4 Net. Защищаемое приложение запрашивает лицензию у HASP LM.

Для каких ключей предназначен

HASP4 Net

Синтаксис

```
hasp (Service, SeedCode, ProgNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	42
ProgNum	Номер, присваиваемый приложению в памяти HASP4 Net при помощи утилиты HASP Edit (см. раздел «Доступ к ключам с использованием HASP Edit»).
Password1	Первый пароль HASP4 Net.
Password2	Второй пароль HASP4 Net.

Возвращаемые значения

Для проверки того, был ли данный сервис выполнен успешно, следует использовать Сервис 40: LastStatus.

Комментарии

Процесс подключения описан в разделе «Как работает HASP4 Net?» на стр. 222.

Данный сервис следует выполнить перед вызовом любого сервиса. Исключения составляют: SetConfigFilename, SetServerByName и LastStatus.

Если вы вызываете процедуру hasp() с сервисом login с одной станции более одного раза, HASP LM не будет регистрировать станцию и приложение во второй раз. Иными словами, HASP License Manager не будет предоставлять приложению дополнительную лицензию.

При запуске защищаемого приложения после зависания компьютера приложение уже будет зарегистрировано, а HASP License Manager не будет предоставлять приложению дополнительную лицензию.

При использовании одного протокола или совместном использовании NetBIOS и IPX многие процессы Win32 в среде Windows NT или Windows95/98/ME требуют только одной лицензии. Тем не менее, эти же процессы в среде этих же операционных систем требуют двух лицензий, если используется протокол TCP/IP вместе с NetBIOS и/или IPX.

Сервис 43: Logout

Описание

Осуществляет отключение от HASP4 Net. При вызове сервиса logout, HASP License Manager удаляет станцию и приложение из регистрационной таблицы HASP4 Net. В результате лицензия высвобождается.

Для каких ключей предназначен

HASP4 Net

Синтаксис

```
hasp (Service, SeedCode, ProgNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	42
ProgNum	Номер, присваиваемый приложению в памяти HASP4 Net при помощи утилиты HASP Edit (см. раздел «Доступ к ключам с использованием HASP Edit»).
Password1	Первый пароль HASP4 Net.
Password2	Второй пароль HASP4 Net.

Возвращаемые значения

Для проверки того, был ли данный сервис выполнен успешно, следует использовать Сервис 40: LastStatus.

Комментарии

Процесс отключения имеет следующие особенности:

- Приложения Win32 требуют одного отключения на каждое подключение.
- Приложения Mac требуют только одного отключения вне зависимости от количества подключений.

  При невыполнении сервиса logout станция и приложение остаются в регистрационной таблице HASP4 Net. При этом:

- Количество подключенных станций и приложений остается неизменным, а лицензия не высвобождается. Если предусматривается одновременная работа защищаемого приложения на n станциях, а n станций осуществили подключение к HASP4 Net, то ни на одной другой станции приложение работать не сможет до тех пор, пока одна из этих n станций не выполнит сервис logout.
- Если та же самая станция повторно активирует приложение, то приложением будет использоваться уже предоставленная лицензия.
- При достижении максимально возможного числа станций, работающих с приложением, и попытке еще одной станции запустить приложение HASP License Manager осуществит поиск простаивающей станции в регистрационной таблице. Если такая станция будет найден, последует ее удаление из регистрационной таблицы. В результате лицензия немедленно освободиться и сможет быть выдана запрашивающей станции. По умолчанию время простоя составляет 36 часов. Его величину можно изменить при помощи Сервиса 48: IdleTime.

Сервис 44: ReadWord

Описание

Считывает одно слово данных из памяти HASP4 Net.

Для каких ключей предназначен

HASP4 Net

Синтаксис

```
hasp (Service, SeedCode, ProgNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	44
ProgNum	Номер, присваиваемый приложению в памяти HASP4 Net при помощи утилиты HASP Edit (см. раздел «Доступ к ключам с использованием HASP Edit» на стр. 71).
Password1	Первый пароль HASP4 Net.
Password2	Второй пароль HASP4 Net.
Par1	Адрес, т.е. адрес памяти HASP4 Net, из которого будет осуществляться считывание (0-247).

Возвращаемые значения

Для проверки того, был ли данный сервис выполнен успешно, следует использовать Сервис 40: LastStatus.

Par2	Данные, т.е. данные, считываемые из памяти HASP4 Net.
Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP API» на стр. 209).

Сервис 45: WriteWord

Описание

Записывает одно слово данных в память HASP4 Net.

Для каких ключей предназначен

HASP4 Net

Синтаксис

```
hasp (Service, SeedCode, ProgNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	45
ProgNum	Номер, присваиваемый приложению в памяти HASP4 Net при помощи утилиты HASP Edit (см. раздел «Доступ к ключам с использованием HASP Edit» на стр. 71).
Password1	Первый пароль HASP4 Net.
Password2	Второй пароль HASP4 Net.
Par1	Адрес, т.е. адрес памяти HASP4 Net, в который будет осуществляться запись (0-247).
Par2	Данные, одно слово данных.

Возвращаемые значения

Для проверки того, был ли данный сервис выполнен успешно, следует использовать Сервис 40: LastStatus.

Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP API» на стр. 209).

Комментарии

Смещения 24 и выше памяти ключей HASP4 Net зарезервированы за списком программ HASP4 Net. Если вы осуществите запись по любому из этих смещений, возможна утеря параметров защиты приложения, которые были ранее сохранены в этой области памяти.

Сервис 46: HaspID

Описание

Определяет ID-номер HASP4 Net.

Для каких ключей предназначен

HASP4 Net

Синтаксис

```
hasp (Service, SeedCode, ProgNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	46
ProgNum	hasp (Service, SeedCode, ProgNum, Password1, Password2, Par1, Par2, Par3, Par4)
Password1	Первый пароль HASP4 Net.
Password2	Второй пароль HASP4 Net.

Возвращаемые значения

Для проверки того, был ли данный сервис выполнен успешно, следует использовать Сервис 40: LastStatus

Par1	IDLow – низкое (наименее важное) слово ID-номера.
Par2	IDHigh – высокое (наиболее важное) слово ID-номера.
Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP API» на стр. 209).

Комментарии

ID-номер представляет собой длинную целочисленную величину (32 бита). Если IDLow и IDHigh беззнаковые, он вычисляется следующим образом:

ID-номер=IDLow+65536*IDHigh

Если IDLow и IDHigh знаковые, то следует компенсировать отрицательное значение прибавлением 65536.

Сервис 48: IdleTime

Описание

Осуществляет контроль над станциями, которые больше не используют защищаемое приложение. Сервис IdleTime позволяет задать время простаивания. Если станция, использующая защищаемое приложение, в течение заданного времени простаивания не обращается к ключу HASP4 Net, то HASP License Manager рассматривает ее как простаивающую.

Для каких ключей предназначен

HASP4 Net

Синтаксис

```
hasp (Service, SeedCode, ProgNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	48
IdleTime	Время простаивания в минутах (0-65535). Станция, которая не обращается к ключу в течение заданного времени, рассматривается как простаивающая.
ProgNum	Номер, присваиваемый приложению в памяти HASP4 Net при помощи утилиты HASP Edit (см. раздел «Доступ к ключам с использованием HASP Edit» на стр. 71).
Password1	Первый пароль HASP4 Net.
Password2	Второй пароль HASP4 Net.

Возвращаемые значения

Для проверки того, был ли данный сервис выполнен успешно, следует использовать Сервис 40: LastStatus.

Комментарии

При достижении максимально возможного числа станций, работающих с приложением, и попытке еще одной станции запустить приложение HASP LM осуществит поиск простаивающей станции в регистрационной таблице. Если такая станция будет найдена, последует ее удаление из регистрационной таблицы. В результате лицензия немедленно освободится и сможет быть выдана запрашивающей станции.

Защищаемое приложение, работающее на станции, которая была удалена из регистрационной таблицы, в случае попытки получить доступ к HASP4 Net выдает код ошибки (135 или 139).

Если станция обращается к HASP4 Net в течение заданного времени простаивания, счетчик времени простаивания обнуляется.

Для использования сервиса IdleTime необходимо вызвать процедуру hasp() после того, как был выполнен сервис Login. Если вы самостоятельно не установили время простаивания, HASP LM будет использовать его значение по умолчанию, равное 36 часам.

Сервис 52: ReadBlock

Описание

Считывает один блок данных из памяти HASP4 Net в программный буфер.

Для каких ключей предназначен

HASP4 Net

Синтаксис

```
hasp (Service, SeedCode, ProgNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	52
ProgNum	Номер, присваиваемый приложению в памяти HASP4 Net при помощи утилиты HASP Edit (см. раздел «Доступ к ключам с использованием HASP Edit» на стр. 71).
Password1	Первый пароль HASP4 Net.
Password2	Второй пароль HASP4 Net.
Par1	Начальный адрес. Определяет начальный адрес памяти HASP4 Net, из которого будет осуществляться считывание (0-247).
Par2	Размер блока. Размер блока данных в словах (максимальное значение - 24 слова).
Par3	Сегмент буфера. Адрес сегмента программного буфера (переменная). Применим только к 16-разрядным приложениям.
Par4	Смещение буфера. Адрес смещения программного буфера (переменная). Размер буфера должен быть не меньше размера блока данных.

Возвращаемые значения

Для проверки того, был ли данный сервис выполнен успешно, следует использовать Сервис 40: LastStatus.

Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP API» на стр. 209).

Сервис 53: WriteBlock

Описание

Записывает один блок данных из программного буфера в память HASP4 Net.

Для каких ключей предназначен

HASP4 Net

Синтаксис

```
hasp (Service, SeedCode, ProgNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	53
ProgNum	Номер, присваиваемый приложению в памяти HASP4 Net при помощи утилиты HASP Edit (см. раздел «Доступ к ключам с использованием HASP Edit»).
Password1	Первый пароль HASP4 Net.
Password2	Второй пароль HASP4 Net.
Par1	Начальный адрес. Определяет начальный адрес памяти HASP4 Net, в который будет осуществляться запись (0-247).
Par2	Размер блока. Размер блока данных в словах (максимальное значение - 24 слова).
Par3	Сегмент буфера. Адрес сегмента программного буфера (переменная). Применим только к 16-разрядным приложениям.
Par4	Смещение буфера. Адрес смещения программного буфера (переменная). Размер буфера должен быть не меньше размера блока данных.

Возвращаемые значения

Для проверки того, был ли данный сервис выполнен успешно, следует использовать Сервис 40: LastStatus.

Комментарии

Смещения 24 и выше памяти ключей HASP4 Net зарезервированы за списком программ HASP4 Net. Если вы осуществите запись по любому из этих смещений, возможна утеря параметров защиты приложения, которые были ранее сохранены в этой области памяти.

Сервис 85: SetConfigFilename

Описание

Определяет конфигурационный файл HASP4 Net, содержащий настройки параметров защищаемого приложения.

Для каких ключей предназначен

HASP4 Net

Синтаксис

```
hasp (Service, SeedCode, ProgNum, Password1,
Password2, Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	85
Par2	Размер буфера. Размер буфера в байтах, содержащего имя файла конфигурации HASP4 Net.
Par3	Сегмент буфера. Адрес сегмента буфера, содержащего имя файла конфигурации HASP4 Net. Применимо только к 16-разрядным приложениям.
Par4	Смещение буфера. Адрес смещения буфера, содержащий имя файла конфигурации HASP4 Net.

Возвращаемые значения

Для проверки того, был ли данный сервис выполнен успешно, следует использовать Сервис 40: LastStatus.

Комментарии

Если вы используете сервис SetConfigFilename, его необходимо вызвать до вызова сервиса Login. Если вы не используете этот сервис, то защищаемое приложение либо не использует файл конфигурации HASP4 Net, либо использует файл nethasp.ini в том случае, если сможет найти его. Подробную информацию о файле конфигурации HASP4 Net можно найти в разделе «Настройка клиентов HASP4 Net» на стр. 263.

Сервис 88: HaspEncodeData

Описание

Шифрует данные при помощи подключенного ключа HASP4 Net.

Для каких ключей предназначен

HASP4 Net

Синтаксис

```
hasp (Service, SeedCode, ProgNum, Password1,
Password2,Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	88
ProgNum	Номер, присваиваемый приложению в памяти HASP4 Net при помощи утилиты HASP Edit (см. раздел «Доступ к ключам с использованием HASP Edit» на стр. 71).
Password1	Первый пароль HASP4 Net.
Password2	Второй пароль HASP4 Net.
Par1	0 (зарезервировано)
Par2	Размер буфера. Размер шифруемого буфера в байтах (минимум 8 байт).
Par3	Сегмент буфера. Адрес сегмента буфера. Применим только к 16-разрядным приложениям.
Par4	Смещение буфера. Адрес смещения буфера. Применим только к 16-разрядным приложениям.

Возвращаемые значения

Для проверки того, был ли данный сервис выполнен успешно, следует использовать Сервис 40: LastStatus.

Par3 — Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP API»).

Комментарии

Содержимое программного буфера шифруется при помощи подсоединенного ключа HASP.

Этот сервис может использоваться только с ключами модельного ряда HASP4. При подключении более ранней модели ключа HASP шифрование будет невозможно, а сервис возвратит код ошибки.

Этот сервис не может быть применен к приложениям DOS.

Если данные шифруются приложением Win16, максимальный размер буфера составит 64Кб. Данные, зашифрованные приложением Win32, размер которых превышает 64Кб, не могут быть расшифрованы API Win16.

Шифрование в процессе работы следует использовать в минимальных объемах. Вместо этого пользователю следует посылать заранее зашифрованные файлы и данные. Для получения дополнительной информации по этому вопросу см. Главу 13 на стр. 163.

Сервис 89: HaspDecodeData

Описание

Дешифрует данные при помощи подключенного ключа HASP4 Net.

Для каких ключей предназначен

HASP4 Net

Синтаксис

```
hasp (Service, SeedCode, ProgNum, Password1,
Password2,Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис	42
ProgNum	Номер, присваиваемый приложению в памяти HASP4 Net при помощи утилиты HASP Edit (см. раздел «Доступ к ключам с использованием HASP Edit» на стр. 71).
Password1	Первый пароль HASP4 Net.
Password2	Второй пароль HASP4 Net.
Par1	0 (зарезервировано)
Par2	Размер буфера. Размер дешифруемого буфера в байтах (минимум 8 байт).
Par3	Сегмент буфера. Адрес сегмента буфера. Применимо только к 16-разрядным приложениям.
Par4	Смещение буфера. Адрес смещения буфера.

Возвращаемые значения

Для проверки того, был ли данный сервис выполнен успешно, следует использовать Сервис 40: LastStatus.

Par3	Статус, т.е. код, показывающий статус операции (см. раздел «Коды статуса HASP API» на стр. 209).

Комментарии

Содержимое программного буфера дешифруется при помощи подсоединенного ключа HASP.

Этот сервис может использоваться только с ключами модельного ряда HASP4. При подключении более ранней модели ключа HASP шифрование будет невозможно, а сервис возвратит код ошибки.

Если данные шифруются приложением Win16, максимальный размер буфера составит 64Кб. Данные, зашифрованные приложением Win32, размер которых превышает 64Кб, не могут быть расшифрованы API Win16.

Сервис 96: SetServerByName

Описание

Определяет имя HASP License Manager, к которому будет подключаться защищаемое приложение.

Для каких ключей предназначен

HASP4 Net

Синтаксис

```
hasp (Service, SeedCode, ProgNum, Password1,
Password2,Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис 96

Par2 Размер буфера. Размер дешифруемого буфера в байтах (минимум 8 байт).

Par3 Сегмент буфера. Адрес сегмента буфера. Применимо только к 16-разрядным приложениям.

Par4 Смещение буфера. Адрес смещения буфера.

Возвращаемые значения

Для проверки того, был ли данный сервис выполнен успешно, следует использовать Сервис 40: LastStatus.

Комментарии

Если вы используете сервис SetServerByName, его необходимо вызвать до вызова сервиса Login.

Для выбора имени HASP LM его следует загрузить с ключом **–srvname** (см. раздел «Ключи HASP License Manager» на стр. 251).

При вызове сервиса SetServerByName процедура hasp() осуществляет поиск HASP LM с определенным именем. Если HASP LM с таким именем будет найден, его место расположения запоминается. При осуществлении подключения к ключу HASP4 Net происходит доступ к HASP LM с определенным местом расположения.

Имя HASP License Manager может содержать до семи символов. Регистр не учитывается.

Сервис 104: HaspQueryLicense

Описание

Запрашивает текущие свойства лицензии у ключа HASP4 Net.

Для каких ключей предназначен

HASP4 Net

Синтаксис

```
hasp (Service, SeedCode, ProgNum, Password1,
Password2,Par1, Par2, Par3, Par4)
```

Используемые параметры

Сервис 42

ProgNum — Номер, присваиваемый приложению в памяти HASP4 Net при помощи утилиты HASP Edit (см. раздел «Доступ к ключам с использованием HASP Edit» на стр. 71).

Password1 — Первый пароль HASP4 Net.

Password2 — Второй пароль HASP4 Net.

Возвращаемые значения

Для проверки того, был ли данный сервис выполнен успешно, следует использовать Сервис 40: LastStatus.

Par1 — Текущее число активных лицензий для данного ProgNum.

Par2 — Максимальное число лицензий для данного ProgNum.

Par3 — Тип ключа HASP4 Net.

Par4 — Число оставшихся активаций для данного ProgNum (-1 – не ограничено).

Сервис 120: GetProtocol

Описание

возвращает протоколы и методы поиска, доступные на станции, где загружается приложение, защищенное на NetHASP

Для каких ключей предназначен

NetHASP

Синтаксис

Hasp (Service, SeedCode, ProgNum, Psw1, Psw2, Par1, Par2, Par3, Par4)

Используемые параметры

Service = 120

Возвращаемые значения

Par1: Маска (16-bit) протоколов, установленных в файле nethasp.ini

Par2: Маска (16-bit) методов поиска, установленных в файле nethasp.ini

Par3: Маска (16-bit) протоколов, установленных на рабочей станции

Комментарии

GetProtocol может вызываться в любое время независимо от использования других сервисов. Для получения списка возможных масок, возвращаемых данным сервисом, следует обратиться к разделу «Маски протоколов».

Сервис 121: GetProtocol

Описание

Устанавливает протоколы и методы поиска на рабочей станции

Для каких ключей предназначен

NetHASP

Синтаксис

```
Hasp (Service, SeedCode, ProgNum, Psw1, Psw2,
Par1, Par2, Par3, Par4)
```

Используемые параметры

Service = 121

Возвращаемые значения

Par1: Маска (16-bit) протоколов, устанавливаемых для рабочей станции

Par2: Маска (16-bit) протоколов, устанавливаемых для рабочей станции

Комментарии

SetProtocol перезаписывает любые назначения, сделанные через файл nethasp.ini, касающиеся протоколов и методов поиска. Сервис SetProtocol должен вызываться перед выполнением операции NetHASP LOGIN. Для получения списка возможных масок, используемых данным сервисом, следует обратиться к разделу «Маски протоколов».

Маски протоколов

Данная таблица содержит список возможных масок, используемых в сервисах NetHASP №120 (Par1 и Par3) и №121(Par1):

Значение маски протокола (decimal)	Описание
0	Нет протокола
1	Протокол IPX
2	Протокол NetBIOS
4	Протокол TCP/IP

Когда установлено или устанавливается более одного протокола, значение маски считается как операция OR между масками всех необходимых протоколов. Например, если установлены или устанавливаются протоколы IPX и TCP/IP, значение маски будет:
00000001 OR 00000100 = 00000101 (decimal 5)

Данная таблица содержит список возможных масок, используемых в сервисах NetHASP №120 (Par2) и №121 (Par2):

Значение маски методов поиска (decimal)	Описание
0	Метод по умолчанию (автопоиск)
1	IPX, метод поиска localnet
4	IPX с использованием SAP (Bindery)
8	UDP/IP с запрещением Broadcast
32	IPX с запрещением Broadcast
128	Метод поиска TCP

Если устанавливается протокол TCP/IP или UDP/IP, то необходимо указать имя или IP-адрес менеджера лицензий NetHASP в файле nethasp.ini.

Сервис 125: RetToDefault

Описание

Перестраивает HASP API на его настройки по умолчанию

Для каких ключей предназначен

NetHASP

Синтаксис

```
Hasp (Service, SeedCode, ProgNum, Psw1, Psw2,
Par1, Par2, Par3, Par4)
```

Используемые параметры

Service = 125

Комментарии

RetToDefault может вызываться в любое время независимо от использования других сервисов. После его вызова необходимо выполнить операцию NetHASP LOGIN для того, чтобы получить доступ к ключу NetHASP. После успешного выполнения LOGIN будут заново проверены все протоколы и перечитан файл nethasp.ini

Коды статуса HASP API

Коды статуса для всех ключей HASP

При использовании API в целях получения доступа к ключу HASP процедура hasp() возвращает в Par3 код статуса. Приведенные ниже таблицы содержат перечень возвращаемых кодов статуса в зависимости от типа используемого ключа HASP.

Таблица 15.1. Коды статуса для всех ключей HASP

Код	Описание
0	Операция выполнена успешно.
-1	Превышено время ожидания: операция записи не была завершена.
-2	Адрес выходит за рамки границ.
-3	Ключ HASP с определенным паролем не был найден.
-4	Был найден ключ HASP, но он не является ключом MemoHASP.
-5	Операция записи не была завершена.

Код	Описание
-6	Параллельный порт недоступен. Активно другое устройство, присоединенное к этому порту, например, принтер. Повторите вызов API через несколько секунд.
-7	Размер буфера недостаточен. Эта ошибка характерна для сервисов, которые предъявляют требования к минимальному размеру буфера.
-8	Аппаратные средства не поддерживают данный сервис. Этот сервис требует ключа HASP4.
-9	Неверный указатель сервиса.
-10	Доступ к ключу запрещен вследствие того, что приложение запущено на Citrix Winframe или Windows Terminal Server (приложение может быть запущено только на экране консоли).
-11	Доступ к ключу запрещен вследствие того, что приложение запущено на Citrix Winframe или Windows Terminal Server (для определения того, что приложение запущено на экране консоли необходим service pack 4+).
-12	Неверно указан параметр сервиса.
-100	Невозможно открыть драйвер устройства HASP. Относится только к приложениям Win32. Установите драйвер устройства HASP.
-101	Невозможно прочесть драйвер устройства HASP. Относится только к приложениям Win32.
-102	Невозможно закрыть драйвер устройства HASP. Относится только к приложениям Win32.
-110	Невозможно открыть драйвер устройства HASP. Относится к приложениям DOS и Windows, пытающимся получить доступ к драйверу устройства HASP. Установите драйвер устройства HASP.
-111	Невозможно прочесть драйвер устройства HASP. Относится к приложениям DOS и Windows, пытающимся получить доступ к драйверу устройства HASP.

Код	Описание
-112	Невозможно закрыть драйвер устройства HASP. Относится к приложениям DOS и Windows, пытающимся получить доступ к драйверу устройства HASP.
-120	Недостаточно памяти DOS. Относится к приложениям DOS и Windows, защищаемых локальными ключами. Попытайтесь освободить память DOS.
-121	Ошибка при освобождении памяти DOS. Относится к приложениям DOS и Windows, защищаемых локальными ключами.
-999	Неверный сервис.

Коды статуса для ключей HASP4 Time

Таблица 15.2. Коды статуса для ключей HASP4 Time

Код	Описание
0	Операция успешно завершена.
-20	Неверный день.
-21	Неверный месяц.
-22	Неверный год.
-23	Неверное количество секунд.
-24	Неверное количество минут.
-25	Неверное количество часов.
-26	Неверный адрес: адрес выходит за границы 0-15.
-27	Превышено время ожидания: операция записи не была завершена.
-28	Ключ HASP с определенным паролем не был найден.
-29	Был найден ключ HASP, но он не является ключом HASP4 Time.

Коды статуса для ключей HASP4 Net

При вызове процедуры hasp() при помощи сервиса LastStatus в Par1 (NetStatus) возвращается код статуса, определяющий состояние предыдущего вызова. При возникновении ошибки в Par2 (System Error) возвращается ее код. Процедурой hasp() также может возвращаться код предупреждения в Par3.

Если процедура hasp() возвращает код ошибки, выполнение сервиса прерывается.

Коды LastStatus разделены на две группы:

- **Коды статуса 1-127** показывают ошибки обмена данными между защищаемым приложением и HASP LM, а также ошибки в параметрах, задаваемых процедуре hasp().
- **Коды статуса 129-151** показывают ошибки, возникающие после установления связи клиент-сервер.

В приведенной ниже таблице перечислены возможные значения Par1 (NetStatus).

Таблица 15.3. Коды статуса HASP4 Net

Код	Описание
0	Операция завершена успешно.
1	Протоколы IPX, NetBIOS или TCP/IP установлены неверно.
2	Коммуникационная ошибка – невозможно получить номер сокета (относится к протоколам IPX и TCP/IP). Проверьте правильность установки протокола.
3	Коммуникационная ошибка. NetBIOS – невозможно начать сессию. IPX – невозможно получить адрес HASP License Manager. Проверьте правильность установки протокола. TCP/IP – невозможно установить соединение с сервером. Проверьте правильность указанного адреса сервера.
4	HASP LM не был найден. Проверьте путь к адресному файлу и наличие разрешения на чтение.
5	Невозможно прочитать адресный файл HASP LM.

Код	Описание
6	Невозможно закрыть адресный файл HASP LM.
7	Коммуникационная ошибка – невозможно послать пакет. Проверьте правильность установки протокола.
8	Сеть перегружена. IPX – сеть перегружена или были найдены неверные адресные файлы. В последнем случае следует удалить все копии файлов *haspaddr.dat* и *newhaddr.dat*.
10	Осуществлен вызов процедуры hasp() одним из сервисов без предварительного вызова сервиса Login.
11	Коммуникационная ошибка – ошибка адаптера (относится только к NetBIOS). Проверьте правильность установки протокола.
15	Не был найден ни один активный HASP LM.
18	Невозможно осуществить подключение к HASP4 Net, поскольку не был осуществлен вызов сервиса SetServerByName.
19	Синтаксическая ошибка в файле конфигурации. В Par2 указывается номер строки, в которой произошла ошибка. Если сервисом LastStatus в Par2 возвращается 0, то это означает, что значение переменной было указано неверно.
20	Ошибка при обращении к файлу конфигурации HASP4 Net. Не удалось осуществить операцию «открыть файл» или «читать файл». Возможной причиной может являться неспособность системы обрабатывать большее количество файлов. В Par2 сервиса LastStatus возвращается код ошибки операционной системы.
21	HASP4 Net не хватает памяти. Данная ошибка характерна для интерфейсов HASP4 Net под Windows и надстройки DOS. Попробуйте освободить память DOS.
22	HASP4 Net не сумел освободить память DOS. Данная ошибка характерна для интерфейсов HASP4 Net под Windows и надстройки DOS.
23	Неверный адрес памяти HASP4 Net.

Код	Описание
25	Ошибка загрузки winsock.dll (относится только к TCP/IP и IPX).
26	Ошибка при выгрузке winsock.dll из памяти (относится только к TCP/IP и IPX).
28	Ошибка запуска winsock.dll (относится только к TCP/IP и IPX).
30	Невозможно закрыть сокет (относится только к TCP/IP и IPX).
33	Попытка вызвать сервис SetProtocol без предварительного подключения к ключу.
34	Доступ к ключу запрещен, поскольку приложение запущено на Citrix Winframe или Windows Terminal Server (для определения того, запущено ли приложение на экране консоли необходим service pack 4+).
129	Соответствующий ключ HASP4 Net не подключился к HASP LM.
130	Указанный номер программы не содержится в списке программ в памяти ключа HASP4 Net.
132	Ошибка записи в память HASP4 Net.
133	Текущий запрос на подключение превышает возможное число одновременно работающих приложений.
134	Текущий запрос на подключение превышает число разрешенных запусков приложения.
135	Вызвана процедура hasp() с сервисом Logout без предварительного вызова сервиса Login.
136	HASP License Manager занят. Это может быть вызвано тем, что станция с ключом HASP4 Net плохо приспособлена для работы в сети.
137	Отсутствует свободное место в регистрационной таблице HASP4 Net.

Код	Описание
138	Внутренняя ошибка HASP4 Net. Количество станций, получивших лицензию, превышает число станций, поддерживаемых данной моделью HASP4 Net.
139	Компьютер с подключенным ключом HASP4 Net завис, или был перезагружен, либо была вызвана процедура hasp() с любым сервисом, за исключением сервиса 40, 85 или 96, без предварительного вызова сервиса Login.
140	HASP LM не обслуживает сеть, в которой находится ваша станция.
141	Неверно указан сервис, либо новая версия HASP API осуществляет обмен данными с более старой версией HASP LM.
142	HASP LM с именем, указанным в конфигурационном файле HASP4 Net, не был найден.
150	Не был найден HASP LM с указанным именем. Код данной ошибки возвращается сервисом SetServer-ByName.
151	Были найдены два или более HASP LM с определенным именем. Код данной ошибки возвращается сервисом SetServerByName.
152	Операция шифрования невозможна, поскольку аппаратные средства не поддерживают данный сервис.
153	Операция дешифрования невозможна, поскольку аппаратные средства не поддерживают данный сервис.
155	Была найдена старая версия HASP LM. Используемая версия API требует новой версии HASP LM.

В приведенной ниже таблице содержится перечень возможных предупреждающих кодов, возвращаемых в Par3 при использовании сервиса LastStatus. Исполнение сервиса завершается, несмотря на предупреждение.

Таблица 15.4. Коды предупреждений HASP4 Net

Код	Описание
1	Включена поддержка протокола IPX в *nethasp.ini* или в переменной среды NETHASPPROTOCOL, однако сам протокол IPX не установлен. Предупреждение может появиться при подключении к HASP4 Net.
2	Включена поддержка протокола NetBIOS в *nethasp.ini* или в переменной среды, однако сам протокол NetBIOS не установлен. Предупреждение может появиться при подключении к HASP4 Net.
3	Включена поддержка протоколов NetBIOS и IPX в *nethasp.ini* или в переменной среды, однако сами протоколы не установлены. Предупреждение может появиться при подключении к HASP4 Net.
4	Включена поддержка протокола TCP/IP в *nethasp.ini* или в переменной среды, однако сам протокол TCP/IP не установлен. Предупреждение может появиться при подключении к HASP4 Net.
5	Включена поддержка протоколов TCP/IP и IPX в *nethasp.ini* или в переменной среды, однако сами протоколы не установлены. Предупреждение может появиться при подключении к HASP4 Net.
6	Включена поддержка протоколов TCP/IP и NetBIOS в *nethasp.ini* или в переменной среды, однако сами протоколы не установлены. Предупреждение может появиться при подключении к HASP4 Net.
7	Включена поддержка протоколов TCP/IP, NetBIOS и IPX в *nethasp.ini* или в переменной среды, однако сами протоколы не установлены. Предупреждение может появиться при подключении к HASP4 Net.

Код	Описание
18	HASP LM возвратил статус OK, но вследствие задержки подключение к HASP4 Net могло быть не закончено. В этом случае необходимо осуществить повторное подключение к HASP4 Net. Если при подключении к HASP4 Net возвращается код ошибки 135, то подключение было осуществлено. Данное предупреждение может появляться при подключении к HASP4 Net.
19	В файле *nethasp.ini* указано неверное ключевое слово или значение, либо новое ключевое слово не распознается более старой версией API. Данное предупреждение может появляться при подключении к HASP4 Net.
20	Протоколы TCP и UDP были заданы в файле *nethasp.ini* вместе с другими протоколами. Если IP-адрес HASP LM не определен в том же файле, то клиентское приложение будет работать с другим определенным протоколом. Тем не менее, появится извещение о том, что TCP или UDP не используются из-за отсутствия IP-адреса. Данное предупреждение может появиться при использовании сервиса SetProtocol.
22	HASP API не может освободить память. Данное предупреждение может появиться после выполнения любого сервиса HASP4 Net API.

Коды HASP4 Net LastStatus постоянно обновляются. Информация о последних обновлениях содержится в файле помощи HASP LM.

Часть 4
Использование HASP в сети

В данной части рассматривается система HASP4 Net, а также соответствующие инструменты и процедуры.

В главе «Основные концепции HASP4 Net» на стр. 221 рассматриваются основные определения и концепции, используемые при защите программного обеспечения при помощи HASP4 Net.

В главе «Защита приложений при помощи HASP4 Net» на стр. 235 описываются методы защиты приложений с использованием HASP4 Net.

В главе «Управление лицензиями HASP4 Net» на стр. 241 приводится процедура установки HASP License Manager, обеспечивающего предоставление лицензий в сети.

В главе «Настройка клиентов HASP4 Net» на стр. 263 описывается процесс настройки приложения, защищаемого HASP4 Net (т.е. клиента HASP4 Net), при помощи файла конфигурации.

В главе «Мониторинг лицензий HASP4 Net» на стр. 275 рассматривается использование утилиты Aladdin Monitor, которая позволяет осуществлять централизованное администрирование ключей HASP4 Net и приложений HASP License Manager.

В главе «Адаптация HASP4 Net к сети» на стр. 281 содержится информация по адаптации HASP4 Net к сети при помощи определения перечня станций, на которых будет работать защищаемое приложение, и определения времени ожидания.

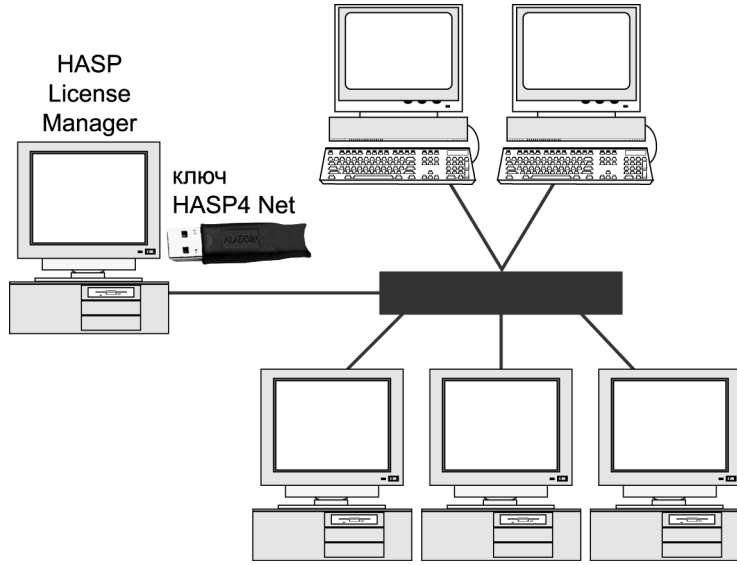
Основные концепции HASP4 Net

В данной главе объясняются понятия и концепции, связанные с защитой программного обеспечения при помощи HASP4 Net. Перед ее прочтением следует ознакомиться с общими концепциями и понятиями HASP. Если вы не используете ключи HASP4 Net, эту главу можно пропустить.

Что такое HASP4 Net?

Ключ HASP4 Net представляет собой модификацию ключа HASP, предназначенную для использования в сети. Использование ключа HASP4 Net позволяет эффективно контролировать число копий защищаемых приложений, запущенных одновременно на находящихся в сети компьютерах, а также защищать приложение от несанкционированного использования.

Рисунок 17.1. Система HASP4 Net



Как работает HASP4 Net?

При активации приложение, находящееся на рабочей сетевой станции, обращается к HASP LM и запрашивает разрешение на запуск. Затем HASP LM осуществляет проверку наличия верного ключа HASP и проверяет в памяти ключа HASP4 Net следующее:

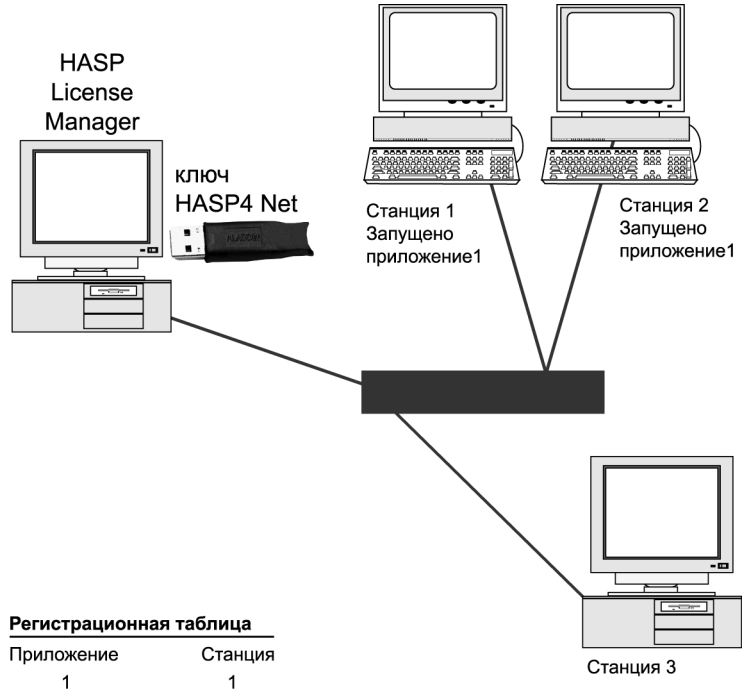
- Защищаемое приложение имеет лицензию на запуск.
- Число, ограничивающее возможность одновременного запуска защищаемого приложения, не превышено.

Если результаты проверки удовлетворительны, HASP LM разрешает работу приложения и обновляет регистрационную таблицу HASP4 Net. В противном случае HASP LM возвращает код ошибки.

Приведенные ниже рисунки иллюстрируют работу HASP4 Net. Условная сеть на рисунках состоит из четырех станций. На одной из станций работает HASP LM; к ней также подключен ключ HASP4 Net.

Защищаемое приложение 1 может быть запущено на пяти станциях. На рисунках отображается процесс активации защищаемого приложения 1 на станции 3.

Рисунок 17.2. До подключения к HASP4 Net...

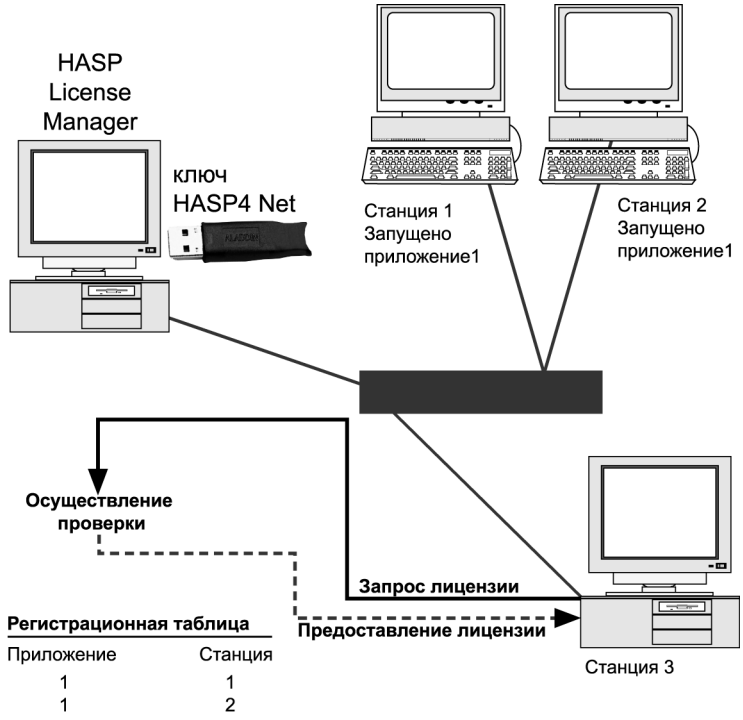


Регистрационная таблица

Приложение	Станция
1	1
1	2

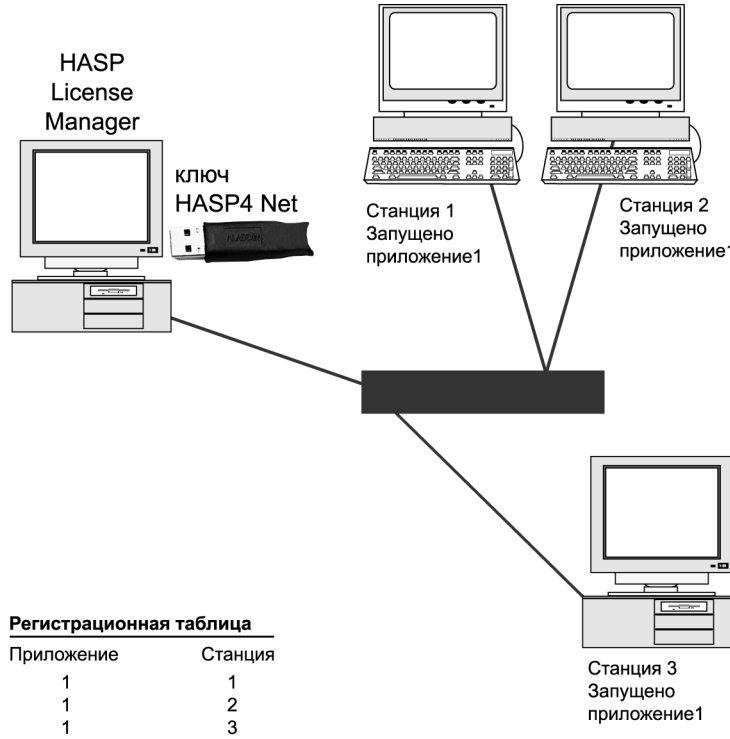
На станциях 1 и 2 запущено защищаемое приложение, что отражается в регистрационной таблице. Из пяти станций, обладающих правом запуска защищаемого приложения в одно и то же время, две станции уже занесены в регистрационную таблицу.

Рисунок 17.3. Осуществлено подключение к HASP4 Net...



На станции 3 загружается защищаемое приложение 1, которое получает доступ к HASP LM. HASP LM определяет наличие у станции 3 лицензии на активацию приложения.

Рисунок 17.4. После подключения к HASP4 Net...



Регистрационная таблица

Приложение	Станция
1	1
1	2
1	3

Проверки, осуществленные HASP LM, дали положительные результаты, и приложению 1 предоставлено право на запуск на станции 3. Регистрационная таблица соответствующим образом обновляется. Теперь в нее занесена и станция 3.

Подготовка защиты

Защита приложений при помощи HASP4 Net

Вы можете выбрать метод защиты приложения. Возможными вариантами является защита приложения для локального использования, для использования в сети, а также для обоих типов одновременно. Для применения HASP4 Net следует воспользоваться одним из следующих методов:

- Предусмотрите использование HASP4 Net при работе с HASP Envelope.
- Используйте API-сервисы HASP4 Net вместо или в дополнение к другим сервисам API.

Более подробная информация по этому вопросу приведена в разделе «Защита приложений при помощи HASP4 Net» (на стр. 235).

Выдача лицензий при помощи ключа HASP4 Net

Ключи HASP4 Net позволяют:

- Ограничивать число станций, на которых запускается защищаемое приложение.
- Ограничивать число активаций каждого защищаемого приложения.

Ключи HASP4 Net выпускаются в нескольких модификациях. Номер модели ключа HASP4 Net определяет максимальное число лицензий для каждого приложения (т.е. максимальное число станций, на которых одновременно может быть запущено защищаемое приложение). Например, ключ HASP4 Net5 рассчитан на предоставление 0-5 лицензий. Другие ключи позволяют предоставлять 10, 20, 50 или 100 лицензий. У ключа HASP4 NetU число лицензий не ограничено. Ключи HASP4 Net поставляются в версии для параллельного порта или порта USB.

Использование HASP4 Net

Установка HASP4 Net

Для работы ключа HASP4 Net в сети необходимо выполнить следующие действия:

- Установите соответствующий драйвер или демон HASP и подсоедините ключ HASP4 Net к компьютеру.
- На этом же компьютере необходимо установить и запустить HASP LM (подробная информация об этой программе содержится в разделе «Управление лицензиями HASP4 Net» на стр. 241).
- При необходимости настройте HASP LM и клиента HASP4 Net (подробная информация приводится в разделах «Настройка HASP License Manager» на стр. 251 и «Настройка клиентов HASP4 Net» на стр. 263).

Управление лицензиями при помощи HASP License Manager

Программа HASP License Manager служит для поддержки обмена данными между ключом HASP4 Net и защищаемым приложением. Обмен данными может обеспечиваться между несколькими защищаемыми приложениями и несколькими ключами HASP4 Net, подсоединенными к компьютеру.

Программа HASP License Manager может работать в следующих средах: Windows 95/98/ME, Windows NT/2000/XP, Mac OS X, Linux и Novell Netware 3.12 и выше.

HASP LM создает регистрационную таблицу, в которой указываются все приложения, осуществившие подключение к ключу HASP4 Net. В списке идентифицируется каждое защищаемое приложение и станция, на которой было запущено защищаемое приложение. Приложения и станции остаются в регистрационной таблице до тех пор, пока не будет осуществлено их отключение от ключа HASP4 Net.

Более подробная информация об этом содержится в разделе «Управление лицензиями HASP4 Net» (на стр. 241).

Запрос лицензии клиентом HASP4 Net

Защищаемое приложение выступает в роли клиента HASP4 Net. Оно осуществляет запрос лицензии у HASP4 Net и обменивается данными с HASP LM. При активировании защищаемого приложения, им осуществляется подключение к HASP4 Net для получения доступа к HASP LM и для получения разрешения на запуск. Приложение информирует HASP LM о завершении работы при помощи отключения от HASP4 Net.

Не следует путать подключение и отключение к HASP4 Net со стандартным сетевым подключением и отключением.

Работа клиента HASP4 Net может осуществляться в следующих средах: Windows 95/98/ME, Windows NT/2000/XP, Mac OS 8.6, Mac OS 9.x, Mac OS X, Linux.

Возможна индивидуальная настройка клиентов HASP4 Net. Более подробная информация содержится в разделе «Настройка клиентов HASP4 Net» на стр. 263.

Мониторинг лицензий при помощи Aladdin Monitor

Aladdin Monitor позволяет осуществлять централизованное администрирование приложений HASP LM и ключей HASP4 Net.

- Возможно выполнение следующих операций:
- Проверка свойств HASP LM.
- Проверка ключей HASP4 Net.
- Запуск и прерывание сервисов HASP LM.

Работа Aladdin Monitor может осуществляться в следующих средах: Windows 98/ME, Windows NT/2000/XP. Программой поддерживаются протоколы TCP/IP и IPX (см. также раздел «Мониторинг лицензий HASP4 Net» на стр. 275).

Передача ключей HASP4 Net

Передача ключей HASP4 Net

Наряду с защищаемым приложением клиентам необходимо передать:

- Ключ HASP4 Net
- Драйвер устройства HASP
- HASP License Manager (с настроенным файлом *nhsev.ini*)
- Aladdin Monitor
- Сконфигурированный файл *nethasp.ini* для клиента HASP4 Net.

Помощь конечному пользователю

Утилита Aladdin Diagnostic позволяет собрать информацию о системе и ключе HASP. Собранная при помощи этой утилиты информация призвана помочь вам и вашему клиенту в разрешении проблем, которые могут возникнуть при использовании защищаемого приложения.

Клиенты могут использовать утилиту Aladdin Diagnostic для:

- проверки наличия ключа HASP;
- создания файла отчета, содержащего данные об устройствах Aladdin и информацию о системе.

Работа Aladdin Diagnostic может осуществляться в следующих средах: Windows 95/98/ME, Windows NT/2000/XP (см. также раздел «Помощь конечным пользователям» на стр. 109).

Поддерживаемые протоколы, платформы и операционные системы

HASP4 Net является кросс-платформенным решением, которое поддерживает следующие операционные системы.

Таблица 17.1. Платформы, поддерживаемые HASP4 Net

	Windows 3.x	Windows 95/98/ ME/NT/2000/XP	Mac OS 8.6	Mac OS 9.x	Mac OS X	Novell	Linux
Клиент HASP4 Net	да	да	да	да	да		да
HASP License Manager	да	да			да	да	да
Aladdin Monitor		да					
Aladdin Diagnostic	да	да					

Для обмена данными между клиентом HASP4 Net и HASP License Manager могут использоваться следующие протоколы.

Таблица 17.2. Протоколы HASP4 Net

	Windows	Apple Macintosh	Novell	Linux
IPX	поддерживается		поддерживается	
TCP/IP (UDP/IP)	поддерживается	поддерживается		поддерживается
NetBIOS	поддерживается			

При упоминании в тексте протокола TCP/IP речь идет также и о протоколе UDP/IP.

Часто задаваемые вопросы

Вопрос Нужно ли устанавливать HASP4 Net на сетевом сервере?

Ответ Нет. Ключ HASP4 Net и HASP LM могут устанавливаться на любой рабочей станции в рамках сети. Для поддержания работы приложений, запущенных на сетевых машинах, данная рабочая станция должна быть активна; на ней должен быть запущен HASP LM.

Вопрос Можно ли узнать, какие станции получают доступ к ключу HASP4 Net?

Ответ Да. Утилита Aladdin Monitor позволяет получить информацию о всех станциях, запустивших приложение и осуществивших подключение к HASP4 Net (см. также раздел «Мониторинг лицензий HASP4 Net» на стр. 275).

Вопрос Если к одной станции подключить два ключа HASP4 Net5 с одинаковым кодом разработчика, будет ли общее число лицензий равняться 10?

Ответ Нет. При подключении к одной рабочей станции двух ключей HASP4 Net с одинаковым кодом разработчика задействован будет только один ключ. Для получения десяти лицензий при помощи двух ключей HASP4 Net5 ключи должны быть подключены к разным рабочим станциями, на которых запущен HASP LM. Кроме того, можно использовать ключ HASP4 Net10.

Вопрос У моего клиента уже есть ключ HASP4 Net от другого производителя программных продуктов. Этот ключ подключен к рабочей станции, а на самой рабочей станции запущен HASP License Manager. Что необходимо сделать для установки моего ключа HASP4 Net?

Ответ Необходимо присоединить ваш ключ HASP4 Net к тому же компьютеру. Уже загруженный HASP License Manager будет обслуживать оба ключа HASP4 Net.

Вопрос Будет ли ключ HASP4 Net работать через Интернет?

Ответ Да. Ключ HASP4 Net может работать через Интернет при помощи протокола TCP/IP.

Вопрос Если, например, у меня есть ключ HASP4 Net20, могу ли я предоставить доступ только 17 пользователям?

Ответ Да. Использование HASP Edit позволяет задать любое количество пользователей, вплоть до 20 (в данном случае).

Вопрос На какой рабочей станции следует установить драйвер устройства HASP при работе с HASP4 Net?

Ответ Только на той станции, к которой подключен ключ HASP4 Net.

Драйвер устройства HASP предназначен для связи ключа HASP и защищаемого приложения. Поскольку защищаемое HASP4 Net приложение обменивается данными с HASP LM, который, в свою очередь, получает доступ к ключу HASP4 Net, драйвер устройства HASP следует устанавливать только на той станции, где запускается HASP LM.

Защита приложений при помощи HASP4 Net

В данной главе описываются возможности защиты при помощи ключа HASP4 Net.

Возможными вариантами является защита приложения для локального использования, для использования в сети, а также для обоих типов одновременно. Для применения HASP4 Net следует воспользоваться одним из следующих методов:

- Предусмотрите использование HASP4 Net при работе с HASP Envelope.
- Используйте API-сервисы HASP4 Net вместо или в дополнение к другим сервисам API.

Возможности HASP Envelope при использовании с HASP4 Net

Использование HASP Envelope позволяет защищать приложение без изменения его программного кода. Информация об основных процедурах в рамках реализации данного варианта защиты приводится в разделе «Защита с помощью HASP Envelope», на стр. 43.

HASP Envelope и приложения Win32

В дополнение к обычным процедурам:

- Введите пароль HASP4 Net.
- Определите номер программы.
- Определите количество лицензий и число активаций для данного программного файла. Если вы не хотите ограничивать какое-либо значение, следует использовать опцию **Unlimited**. Данная информация имеет значение только в том случае, если вы желаете записать ее в память подсоединенного ключа HASP4 Net.
- Определите имя файла конфигурации HASP4 Net (см. также раздел «Настройка клиентов HASP4 Net», на стр. 263).

Если вы хотите защитить свое приложение как для локального, так и для сетевого использования, можно определить сетевые параметры HASP4 Net *в дополнение* к параметрам для локального использования.

Командная строка HASP Envelope

В командной строке HASP Envelope можно использовать следующие ключи:

Таблица 18.1. Ключи HASP Envelope

Ключ	Функция
-nhpass <password1> <password2>	Определяет пароли HASP4 Net.
-prgnum <program number>	Определяет уникальный номер программы (1-112).
-netcfg <filename>	Определяет файл конфигурации HASP4 Net.

API-сервисы HASP4 Net

При реализации защиты приложения, предназначенного для использования в сети с добавлением в программный код вызовов API, следует использовать сервисы HASP4 Net. Подробное описание сервисов приводится в разделе «Сервисы HASP4 Net» (стр. 177).

Если вы хотите защитить приложение для использования локально или в сети, следует использовать и сервисы HASP4 Net, и сервисы памяти, и/или основные сервисы.

Таблица 18.2. API-сервисы HASP4 Net

Сервис	Название	Операция
40	LastStatus	Проверяет статус последнего вызова. Данный сервис следует использовать после каждого вызова процедуры hasp().
42	Login	Запрашивает разрешение на активацию приложения у HASP LM. Данный сервис должен быть первым вызовом процедуры hasp() (за исключением случаев использования сервисов 85 и 96).
43	Logout	Запрашивает завершение сессии HASP4 Net у HASP LM.
44	ReadWord	Считывает одно слово данных из памяти HASP4 Net.
45	WriteWord	Записывает одно слово данных в память HASP4 Net.
46	HaspID	Позволяет получить ID-номер HASP4 Net.
48	IdleTime	Задает максимальное время простоя станции.
52	ReadBlock	Считывает один блок данных из памяти HASP4 Net.
53	WriteBlock	Записывает один блок данных в память HASP4 Net.
85	SetConfig-Filename	Определяет имя файла конфигурации HASP4 Net.
88	HaspEncode-Data	Шифрует данные, посылаемые на подключенный ключ HASP4 Net. Данный сервис следует использовать вместе с сервисом HASPDecodeData в целях осуществления проверки присутствия в сети определенного ключа HASP4 Net.

Сервис	Название	Операция
89	HaspDecod-eData	Дешифрует данные, посылаемые на подключенный ключ HASP4 Net. Данный сервис следует использовать вместе с сервисом HASPEcodeData в целях осуществления проверки присутствия в сети определенного ключа HASP4 Net.
96	SetServer-ByName	Определяет имя HASP LM, к которому защищаемое приложение будет осуществлять подключение.

Управление лицензиями HASP4 Net

В данном разделе описывается процесс управления лицензиями HASP4 Net при помощи HASP License Manager.

HASP License Manager предназначен для обеспечения обмена данными между ключом HASP4 Net и защищаемым приложением. При этом обмен данными может обеспечиваться между несколькими приложениями и несколькими ключами HASP4 Net, подключенными к компьютеру.

HASP LM предназначен для работы в следующих средах: Windows 95/98/ME, Windows NT/2000/XP, Mac OS X, Linux и Novell Netware 3.12 и выше.

Как работает HASP License Manager?

HASP License Manager создает регистрационную таблицу, в которой указываются все приложения, осуществившие подключение к ключу HASP4 Net. В списке идентифицируется каждое защищаемое приложение и станция, на которой было запущено защищаемое приложение. Приложения и станции остаются в регистрационной таблице до тех пор, пока не будет осуществлено их отключение от ключа HASP4 Net.

Для слежения за количеством сетевых станций, на которых одновременно запущено защищаемое приложение (а их количество не должно превышать значения, заданного разработчиком программного обеспечения), HASP LM использует регистрационную таблицу. По умолчанию регистрационная таблица может отслеживать подключения до 250 приложений.

HASP License Manager для Windows

Существуют версии HASP License Manager для Windows 95/98/ME/2000/NT/2000/XP (в виде исполняемого файла) и для Windows NT/2000/XP (в виде сервиса).

HASP LM для Windows поддерживает TCP/IP, IPX и NetBIOS. Протоколы могут загружаться и выгружаться при помощи графического интерфейса или ключей командной строки HASP LM.

Установка HASP License Manager в среде Windows

Оба типа HASP LM могут быть установлены при помощи файла *lmsetup.exe*.

Соответствующий HASP LM следует устанавливать на ту сетевую станцию, к которой подключен ключ HASP4 Net.

Настройка установленного HASP LM может осуществляться одним из следующих методов:

- Запустите HASP LM с соответствующими ключами (см. раздел «Ключи HASP License Manager» на стр. 251).
- Используйте файл конфигурации *nhsrv32.ini* (см. раздел «Настройки файла конфигурации nhsrv.ini» на стр. 248).
- Используйте установочный API HASP License Manager (только в виде сервиса Win32), см. раздел «Установочный API HASP License Manager». на стр. 257.

Установка в среде Windows 95/98/ME

Исполняемым файлом HASP LM для Windows является *nhsrvw32.exe*. Для его установки следует использовать файл *lmsetup.exe*.

1. Установите драйвер устройства HASP, подключите ключ HASP4 Net к сетевой станции.
2. Установите HASP LM, запустив с компакт-диска файл *lmsetup.exe* и следуя инструкциям мастера установки.

Установка в среде Windows NT/2000/XP

Исполняемым файлом HASP LM для Windows NT/2000/XP является *nhsrvice.exe*. Для его установки следует использовать файл *lmsetup.exe*.

HASP LM рекомендуется устанавливать в виде сервиса NT.

1. Установите драйвер устройства HASP, подключите ключ HASP4 Net к сетевой станции.
2. Установите HASP LM, запустив с компакт-диска файл *lmsetup.exe* и следуя инструкциям мастера установки. В качестве типа установки выберите **Service**.

Процедуру установки сервиса HASP License Manager можно внедрить в программный код защищаемого приложения. Для этого следует использовать установочный API HASP License Manager API, который находится в директории *utility\servers\win32\service\lmapi*.

Запуск и завершение работы HASP License Manager

Приложение HASP License Manager

Для запуска приложения HASP LM выберите соответствующий пункт меню **Start** или воспользуйтесь **Windows Explorer**. Приложение HASP LM будет находиться в активном состоянии, если загружен какой-либо протокол и к компьютеру подключен ключ HASP4 Net. Для завершения работы HASP LM следует выбрать **Exit** в основном меню.

Сервис HASP License Manager

Для запуска приложения HASP LM выберите соответствующий пункт меню **Start** или воспользуйтесь **Windows Explorer**.

Для завершения работы сервиса HASP LM используйте стандартные процедуры Windows Service.

Для запуска и завершения работы сервиса HASP LM можно также использовать Aladdin Monitor.

Управление HASP License Manager

Управление HASP LM может осуществляться при помощи графического интерфейса или командной строки.

Для открытия окна графического интерфейса следует дважды щелкнуть мышкой по пиктограмме красного ключа HASP4 Net.

Основное окно HASP LM содержит следующую информацию:

- Версия HASP LM

- Статус каждого протокола (**loaded, unloaded** либо **failed to load**), а также дату и время последнего изменения статуса
- Статус HASP LM (активный или неактивный)

Основное окно HASP LM можно закрыть, щелкнув по кнопке закрытия в правом верхнем углу окна. Тем не менее, HASP LM будет продолжать работать, а его пиктограмма будет находиться на панели задач.

Для выхода из программы в панели меню следует выбрать **Exit**. Если HASP LM был установлен как сервис Windows NT, этот способ неприменим.

Загрузка протоколов

Для включения поддержки протокола его следует выбрать в меню **Load** (выбрать можно только тот протокол, который был установлен на компьютере).

Отключение протокола

Для отключения протокола его следует выбрать в меню **Remove**.

Просмотр Activity Log

Для просмотра событий HASP License Manager следует выбрать в панели меню **Activity Log**. Откроется соответствующее окно. Для просмотра событий по определенному протоколу его следует выбрать из списка.

HASP License Manager для MAC

Существует версия HASP LM для Mac OS X, использующая для обмена данными протокол TCP/IP.

HASP LM для Mac состоит из демона и графического интерфейса. Управление HASP LM для Mac может осуществляться при помощи графического интерфейса или командной строки.

Установка HASP License Manager

Для установки HASP LM в среде MAC OS X следует использовать установочную утилиту HASP License Manager Install.

Для установки HASP LM необходимо обладать правами администратора.

1. Дважды щелкните на файле *HASP License Manager Install*, находящемся в директории *License Manager* установочного компакт-диска.
2. Введите пароль администратора.
3. Выберите место установки.
4. Выберите **Install**.

Запуск и завершение работы HASP License Manager

Для запуска HASP LM следует выбрать данное приложение в соответствующем меню и запустить демон, выбрав **Start Daemon** в окне приложения.

Для автоматической загрузки HASP LM следует выбрать опцию Activate in System startup.

Управление HASP License Manager

Управление HASP LM может осуществляться при помощи графического интерфейса или командной строки (см. раздел «Ключи HASP License Manager на на стр. 251).

В окне HASP LM для Mac показывается следующая информация:

- Имя сервера и его IP-адрес
- Доступные ключи
- Был ли запущен демон при загрузке системы
- Статус демона

Доступны следующие опции:

- Установочные ключи (только в том случае, если не выполняется процесс демона)
- Запуск и остановка демона
- Запуск демона при загрузке системы

Определение имени сервера

HASP License Manager можно присвоить до шести имен сервера.

Запущенному демону имя сервера присвоить нельзя, поскольку имена присваиваются при запуске демона.

Для присвоения имени сервера активируйте опцию **SRVNAMES** и введите до шести имен. Разделите имена при помощи пробелов, двоеточия или точки с запятой. Имена будут присвоены при запуске демона.

Постарайтесь избежать использования символов, которые не относятся к ASCII, поскольку их коды могут отличаться в зависимости от системы. Регистр символов при присвоении имени не учитывается.

Настройка файла конфигурации

Настройка файла конфигурации для HASP LM может осуществляться при помощи файла конфигурации. Для определения имени файла конфигурации и пути к нему следует активировать опцию **CFGFILE** и ввести их. Информация о файле конфигурации приводится в разделе «Настройки файла конфигурации nhsrv.ini» на стр. 248.

Запуск и остановка демона

Для запуска и остановки демона следует использовать кнопки в окне приложения.

Автоматическая активация демона

Демон может запускаться автоматически при загрузке системы. Для этого следует активировать опцию **Activate in System startup**.

Работа HASP License Manager на сервере файлов Novell

Исполняемым файлом HASP LM для Windows является *haspserv.nlm*. Обмен данными осуществляется по протоколу IPX.

Использование USB-ключей с Novell не поддерживается.

Загрузка HASP License Manager

Для загрузки HASP LM:

1. Подсоедините ключ HASP к серверу Novell.
2. Скопируйте файл *haspserv.nlm* в системную директорию.
3. Загрузите HASP LM. Для этого введите **load haspserv.** Появится окно HASP LM.

Для автоматической загрузки HASP LM следует добавить load haspserv в файл *autoexec.ncf*, находящийся в директории *sys:system*.

Удаление HASP License Manager

Для удаления HASP LM следует ввести **unload haspserv**.

Работа HASP License Manager на сервере Linux

HASP LM для Linux работает с использованием протокола TCP/IP и может обслуживать клиентов, работающих под Win32, Linux и Mac.

Установка HASP License Manager для Linux

1. Установите драйвер HASP и демон askusbd
2. Откройте терминальное окно и перейдите в директорию с дистрибутивом HASP LM
3. запустите с правами root

   ```
   ./dinst
   ```

   LM будет установлен и будут настроены скрипты для автоматического запуска при старте системы.
4. Если на Linux-сервере установлен межсетевой экран, убедитесь, что траффик через порт 745/udp разрешен. В зависимости от версии Linux вы можете использовать утилиты ipfwadm или ipchains для запроса/изменения настроек.

Настройка HASP License Manager

При установке или работе с HASP LM может возникнуть необходимость в его приспособлении к сетевому окружению. Для настройки можно использовать один из следующих методов:

- Работать с HASP LM с использованием ключей.
- Использовать файл конфигурации *nhsrv.ini*.
- Использовать установочный API License Manager (только сервис Win32).

Ключи HASP License Manager

HASP LM можно запустить с помощью различных ключей, которые зададут используемые протоколы и процедуры работы с клиентами HASP4 Net.

Таблица 19.1. Ключи HASP License Manager

Ключ	Действие	Novell	Windows	Mac	Linux
-?	Вызывает список доступных ключей	да			да
-addrpath=<path>	Определяет место сохранения файла *haspaddr.dat*. По умолчанию файл сохраняется в той директории, откуда был загружен HASP LM.	да	да		
-c	Определяет местонахождения файла конфигурации HASP LM.			да	да
-help	Вызывает список доступных ключей.	да		да	
-ipx	Инструктирует систему HASP4 Net использовать протокол IPX с SAP.	да	да		

Ключ	Действие	Novell	Windows	Mac	Linux
-ipxnosap	Инструктирует систему HASP4 Net использовать протокол IPX без SAP. При использовании HASP LM для Win32 другие протоколы можно загрузить с помощью ключей –tcpip или –netbios. В этом случае HASP LM создает файл *newaddr.dat*, в котором содержится адрес станции, на которой запущен HASP LM. При загрузке HASP LM с одним из этих ключей обмениваться данными с ним смогут только те защищаемые приложения, которые имеют доступ к файлу *newaddr.dat*.	да	да		
-ipxsocket-num= <number>	Данный ключ следует использовать в тех случаях, когда необходимо изменить сокет, который используется для обмена данными HASP LM. Сокет по умолчанию – 7483 (шестнадцатеричное значение).	да	да		
-localnet	Данный ключ следует использовать только в том случае, если вы хотите, чтобы HASP LM обслуживал станции исключительно в локальной сети. Если HASP LM получает запросы от станций, которые не входят в локальную сеть, им возвращается код ошибки 140.	да	да		
-nbname= <name>	Присваивает HASP LM имя NetBIOS. Действие ключа идентично –nethaspnb-name.		да		
-netbios	Данный ключ позволяет использовать системе HASP4 Net исключительно протокол NetBIOS. При использовании HASP LM для Win32 другие протоколы можно загрузить с помощью ключей –tcpip или –ipxnosap.		да		
-portnum= <number>	Если используется протокол TCP/IP, данный ключ позволяет задать порт, который будет использовать HASP LM. Порт по умолчанию – 475.	да	да		

Ключ	Действие	Novell	Windows	Mac	Linux
-ipxnosap	Инструктирует систему HASP4 Net использовать протокол IPX без SAP. При использовании HASP LM для Win32 другие протоколы можно загрузить с помощью ключей –tcpip или –netbios. В этом случае HASP LM создает файл *newaddr.dat*, в котором содержится адрес станции, на которой запущен HASP LM. При загрузке HASP LM с одним из этих ключей обмениваться данными с ним смогут только те защищаемые приложения, которые имеют доступ к файлу *newaddr.dat*.	да	да		
-ipxsocket-num= <number>	Данный ключ следует использовать в тех случаях, когда необходимо изменить сокет, который используется для обмена данными HASP LM. Сокет по умолчанию – 7483 (шестнадцатеричное значение).	да	да		
-localnet	Данный ключ следует использовать только в том случае, если вы хотите, чтобы HASP LM обслуживал станции исключительно в локальной сети. Если HASP LM получает запросы от станций, которые не входят в локальную сеть, им возвращается код ошибки 140.	да	да		
-nbname= <name>	Присваивает HASP LM имя NetBIOS. Действие ключа идентично –nethaspnb-name.		да		
-netbios	Данный ключ позволяет использовать системе HASP4 Net исключительно протокол NetBIOS. При использовании HASP LM для Win32 другие протоколы можно загрузить с помощью ключей –tcpip или –ipxnosap.		да		
-portnum= <number>	Если используется протокол TCP/IP, данный ключ позволяет задать порт, который будет использовать HASP LM. Порт по умолчанию – 475.	да	да		

Ключ	**Действие**	**Novell**	**Windows**	**Mac**	**Linux**
-saptofile	В случае использования этого ключа HASP LM создает файл *newaddr.dat*, в котором содержится адрес станции, на которой запущен HASP LM.	да			
-srvname=<name> [,name]	Присваивает HASP LM одно или несколько имен IPX, TCP/IP или NetBIOS. Может быть присвоено не более шести имен.	да	да	да	да
-tcpip	Данный ключ позволяет использовать системе HASP4 Net исключительно протокол TCP/IP. При использовании HASP LM для Win32 другие протоколы можно загрузить с помощью ключей –ipx или –netbios.		да		
-use-lananum=<x> [,x]	Инструктирует HASP LM работать с определенными номерами коммуникационного канала.		да		
-userlist	Ограничивает число пользователей, обслуживаемых HASP LM. Значение по умолчанию – 250.		да		
-user	Запуск LM в режиме пользователя.				да
-queue	Использовать дополнительный процесс.				да

Настройки файла конфигурации nhsrv.ini

Настройка HASP LM может осуществляться при помощи файла конфигурации *nhsrv.ini*. Копия этого файла устанавливается вместе с утилитами HASP.

Последовательность поиска

Файл *nhsrv.ini* и исполняемый файл HASP LM можно поместить в одну и ту же директорию или в разные места, в зависимости от последовательности поиска *nhsrv.ini*, описанной в приведенной ниже таблице.

Таблица 19.2. Последовательность поиска *nhsrv.ini*

Операционная система	Последовательность поиска
Windows 95/98/ME	Директория с исполняемым файлом Текущая директория Системный каталог Windows Директория Windows Путь
Windows NT4/2000/XP	Директория с исполняемым файлом Текущая директория Системный каталог Windows Директория Windows Путь
Novell	Текущая директория
Mac	Текущая директория. Имя файла конфигурации и путь к нему можно задать при помощи ключа –c.

Ключевые слова в разделе [NHS_SERVER]

Настройка HASP LM осуществляется при помощи внесения изменений в ключевые слова секции [NHS_SERVER] файла *nhsrv.ini*.

nhs_ip_limit

Возможные значения	<IpAddr>, <IpAddr>,...
Описание	Определяет диапазон сетевых станций, обслуживаемых HASP LM. Применяется при использовании HASP LM с Win32, Novell и Mac. Например: 10.1.1.1, 10.1.1.*, 10.1.1.1/32, 10.1.1.1/24

nhs_ip_limit

Возможные значения	<IpAddr-SubMask>,<IpAddr-SubMask>,...
Описание	Определяет IP-адрес одной или более сетевой карты, которые будет обслуживать HASP LM. Применяется при использовании HASP LM с Win32. Например: 10.1.1.111-255.255.0.0

Установочный API HASP License Manager

API обеспечивает выполнение ряда функций, позволяющих устанавливать и удалять сервис HASP License Manager.

Установка с HaspLMInstall

Назначение

Устанавливает и/или вводит соответствующие значения регистра для сервиса HASP LM в среде Windows NT.

Структура

```
DWORD HaspLMInstall(
DWORD InstallMode //Режим установки
LPSTR LMPath //Путь к HASP License Manager
LPSTR CmdLineSwitches //Ключи командной строки
);
```

Параметры

InstallMode	Определяет характеристики процесса установки.
LMPath	Полный путь к исполняемому файлу HASP LM. Используется Service Control Manager для поиска HASP LM. При введении нулевой строки функцией будет использоваться путь к DLL, а применяться будет имя исполняемого файла HASP LM по умолчанию – *nhservw32.exe*. Исполняемый файл находится в папке *Utilities\ Servers\ Win32\ Service* компакт-диска.
CmdLineSwitches	Пустая строка или иная строка, содержащая ключи командной строки, используемые при загрузке HASP LM (см. раздел «Ключи HASP License Manager» на стр. 251).

Опции InstallMode	LM_SERVICE_INSTALL	Устанавливает HASP LM в виде сервиса Windows NT.
	LM_SERVICE_START	Данная опция позволяет запускать HASP LM без перезагрузки. Вызовите HaspLMInstall() с данным параметром после вызова с LM_SERVICE_INSTALL или вызовите LM_SERVICE_INSTALL с ORED.
Возвращаемые значения	В случае успешного выполнения функции будет возвращено значение LM_SUCCESS. Если выполнить функцию не удалось, то будет возвращено значение LM_FAIL. Информацию об ошибке можно получить, вызвав HaspLMLastErrorEx().	
Замечания	Данная функция не копирует файлы HASP LM.	

Удаление с HaspLMRemove()

Назначение	Удаляет соответствующие значения из регистра или деинсталлирует сервис.	
Структура	DWORD HaspLMRemove(DWORD RemoveMode //Режим удаления LPSTR LMPath //Для будущего использования);	
Параметры	RemoveMode	Определяет характеристики процесса удаления.
	LMPath	Для будущего использования. В настоящий момент значение – NULL.
Опции RemoveMode	LM_REMOVE_SERVICE	Удаляет сервис HASP LM в среде Windows NT.
	LM_REMOVE_SERVICE_UNLOAD	Удаляет сервис HASP LM в среде Windows NT из памяти. Сервис остается установленным и будет запущен при следующей загрузке системы.

Возвращаемые значения В случае успешного выполнения функции будет возвращено значение LM_SUCCESS. Если выполнить функцию не удалось, то будет возвращено значение LM_FAIL. Информацию об ошибке можно получить, вызвав HaspLMLastErrorEx().

Замечания Удаление сервиса HASP LM в среде Windows NT приведет к посылке команды HASP LM к завершению работы.

Вывод информации с HaspLMInfo()

Назначение Позволяет получить информацию об установленном сервисе HASP LM и другую информацию общего характера.

Структура

```
DWORD HaspLMInfo(
LPLMINFO lpLMInfo, // Адрес структуры информации
);
```

Параметры

lpLMInfo — Указывает на структуру LMINFO, которая получает информацию, касающуюся установленного сервиса HASP LM.

Возвращаемые значения В случае успешного выполнения функции будет возвращено значение LM_SUCCESS. Если выполнить функцию не удалось, то будет возвращено значение LM_FAIL. Информацию об ошибке можно получить, вызвав HaspLMLastErrorEx().

Запрос статуса с HaspLMLastError()

Назначение Позволяет получить информацию о последнем вызове функции HASP License Manager API.

Структура

```
DWORD HaspLMLastError(
DWORD *System Error // Адрес системной ошибки
LPSTR ErrorStr // Адрес описывающего ошибку буфера
DWORD ErrorStrSize // Размер описывающего ошибку буфера);
```

Параметры	*System Error	Указывает на переменную, содержащую номер системной ошибки.
	ErrorStr	Указывает на буфер, который получит описание последней ошибки сервиса HASP LM.
	ErrorStrSize	Размер буфера ErrorStr (в байтах).

Возвращаемые значения

В случае успешного выполнения функции будет возвращено значение LM_SUCCESS. Если выполнить функцию не удалось, то будет возвращено значение LM_FAIL.

Сообщения об ошибках

Функциями HASP LM Installation возвращаются следующие сообщения об ошибках:

Таблица 19.3. Коды ошибок установочного API HASP LM

Ошибка	Описание
CLOSE_KEY_FAIL	Невозможно закрыть ключ.
CLOSE_SERVICE_FAIL	Невозможно закрыть сервис.
CLOSE_SERVICE_MANAGER_FAIL	Ошибка при закрытии менеджера базы данных сервиса.
CONTROL_SERVICE_FAIL	Невозможно контролировать сервис.
CREATE_SERVICE_FAIL	Невозможно создать сервис.
DELETE_SERVICE_FAIL	Невозможно удалить сервис.
FREE_LIB_FAIL	Невозможно освободить DLL.
GET_DATE_FAIL	Невозможно получить дату.
GET_PROC_ADDR_FAIL	Невозможность получить адрес от DLL.
INVALID_PARAM	Неверный параметр.
LOAD_LIB_FAIL	Невозможно загрузить DLL.
OPEN_KEY_FAIL	Невозможно открыть ключ.
OPEN_SERVICE_FAIL	Невозможно открыть сервис.
OPEN_SERVICE_MANAGER_FAIL	Невозможно открыть менеджер базы данных сервиса.
SERVICE_NOT_SUPPORTED	Данный сервис не поддерживается.
SET_VALUE_FAIL	Невозможно установить значение.
START_SERVICE_FAIL	Невозможно начать выполнение сервиса.

Настройка клиентов HASP4 Net

В данном разделе описывается процесс настройки приложения, защищаемого HASP4 Net (т.е. клиента HASP4 Net) при помощи файла конфигурации.

Если клиент находит соответствующий файл конфигурации, то происходит считывание и использование параметров. В противном случае используются значения по умолчанию.

Использование файла конфигурации позволяет настроить поиск HASP LM.

Именем файла конфигурации HASP4 Net по умолчанию является *nethasp.ini*. Копия данного файла включается в состав утилит HASP и HASP API. Если вы хотите изменить имя файла *nethasp.ini*, новое имя следует использовать и при защите приложения при помощи HASP Envelope или HASP API.

Последовательность поиска файла конфигурации

Последовательность поиска файла конфигурации зависит от операционной системы и типа приложения.

Защищаемое приложение осуществляет поиск файла конфигурации при выполнении первого сервиса HASP4 Net. Поиск осуществляется в следующей последовательности:

Таблица 20.1. Последовательность поиска файла конфигурации

Тип приложения/ операционная система	Последовательность поиска
Win16	Текущая директория, директория Windows, системный каталог Windows, директория с исполняемым файлом, директории, перечисленные в переменной окружения PATH
Win32	Директория с исполняемым файлом, текущая директория, системный каталог Windows, директория Windows, директории, перечисленные в переменной окружения PATH
Mac OS 8.6, 9.1, Mac OS X (только приложения Carbon)	Текущая директория
Mac OS X	Текущая директория, директория текущего пользователя, остальные директории.

В среде Mac OS X поиск файла *nethasp.ini* осуществляется без точки. Если операционной системой учитывается регистр символов, убедитесь в том, что название файла *nethasp.ini* указывается в нижнем регистре.

Разделы файла конфигурации

Файл конфигурации HASP4 Net содержит четыре раздела:

- [NH_COMMON] – для общих настроек
- [NH_IPX] – для протокола IPX
- [NH_NETBIOS] – для протокола NetBIOS
- [NH_TCPIP] – для протокола TCP/IP

В разделе [NH_COMMON] содержатся глобальные настройки для всех разделов файла конфигурации. Во всех остальных разделах содержатся настройки, влияющие на выполнение операций с конкретным протоколом.

Определение ключевых слов

В каждом разделе можно использовать ключевые слова для данного раздела или общие для всех разделов. Использование общего для всех разделов ключевого слова в разделе для одного из трех протоколов обладает большим приоритетом, чем настройки в разделе [NH_COMMON] (относящиеся к этому протоколу).

Для определения дополнительных настроек конкретного протокола следует использовать ключевые слова для конкретного раздела.

Настройки API и Envelope обладают большим приоритетом, чем настройки файла конфигурации.

Каждая строка файла конфигурации начинается с точки с запятой (;). Для использования строки ее следует удалить. Если вы хотите добавить к строке комментарии, впереди них следует ставить точку с запятой.

В имени файла конфигурации HASP4 Net и используемых в нем ключевых словах регистр не имеет значения (за исключением Mac OS X с зависимой от регистра файловой системой).

Далее описываются разделы файла конфигурации HASP4 Net. Для каждого ключевого слова указываются возможные значения и приводится их краткое описание. В случае существования значения по умолчанию оно также указывается.

[NH_COMMON]

Ключевые слова раздела [NH_COMMON]

nh_ipx

Возможные значения	включено, выключено
Описание	Использовать протокол IPX.

nh_netbios

Возможные значения	включено, выключено
Описание	Использовать протокол NetBIOS.

nh_tcpip

Возможные значения	включено, выключено
Описание	Использовать протокол TCP/IP.

Общие ключевые слова раздела [NH_COMMON]

nh_session

Возможные значения	<число>
Описание	Определяет максимальный период времени, в течение которого защищаемое приложение пытается установить связь с HASP LM.
Значение по умолчанию	2 секунды

nh_send_rcv

Возможные значения	<число>

Описание	Устанавливает для HASP LM максимальное время получения или отправки пакета.
Значение по умолчанию	1 секунда

[NH_IPX]

Ключевые слова раздела [NH_IPX]

nh_use_bindery

Возможные значения	включено, выключено
Описание	Использовать протокол IPX с bindery. Игнорируется Win32 API. Данный ключ заменяет ключ NH_USE_SAP.
Значение по умолчанию	выключено

nh_use_broadcast

Возможные значения	включено, выключено
Описание	Использовать широковещательный механизм IPX.
Значение по умолчанию	включено

nh_bc_socket_num

Возможные значения	<число>

Описание	Определяет номер сокета для широковещательного механизма. Число указывается в шестнадцатеричном виде.
Значение по умолчанию	7483H

nh_use_int

Возможные значения	2F_NEW, 7A_OLD
Описание	2F_NEW означает, что протокол IPX будет использовать прерывание 2Fh. 7F_OLD означает, что протокол IPX будет использовать прерывание 7Ah.
Значение по умолчанию	2F_NEW

nh_server_name

Возможные значения	<имя>, <имя2>,...
Описание	Обменивается данными с HASP LM с определенным именем. Максимально – 6 имен; каждое имя может состоять максимально из 7 символов.

nh_search_method

Возможные значения	localnet, internet
Описание	Определяет то, будет ли приложение обмениваться данными только с HASP LM, находящимся в локальной сети, или с любыми другими HASP LM.
Значение по умолчанию	internet

nh_datfile_path

Возможные значения	<путь>
Описание	Определяет местонахождения адресного файла HASP LM.

Общие ключевые слова раздела [NH_IPX]

nh_session

Возможные значения	<число>
Описание	Определяет максимальный период времени, в течение которого защищаемое приложение пытается установить связь с HASP LM.
Значение по умолчанию	2 секунды

nh_send_rcv

Возможные значения	<число>
Описание	Устанавливает для HASP LM максимальное время получения или отправки пакета.
Значение по умолчанию	1 секунда

[NH_NETBIOS]

Ключевые слова раздела [NH_NETBIOS]

nh_nbname

Возможные значения	<имя>
Описание	Присваивает имя HASP LM. Максимально: 1 имя, содержащее до восьми символов (регистр символов не имеет значения).

nh_uselananum

Возможные значения	<число>
Описание	Устанавливает номер коммуникационного канала, который будет использоваться в качестве коммуникационного канала.

Общие ключевые слова раздела [NH_NETBIOS]

nh_session

Возможные значения	<число>
Описание	Определяет максимальный период времени, в течение которого защищаемое приложение пытается установить связь с HASP LM.
Значение по умолчанию	2 секунды

nh_send_rcv

Возможные значения	<число>
Описание	Устанавливает для HASP LM максимальное время получения или отправки пакета.
Значение по умолчанию	1 секунда

[NH_TCPIP]

Ключевые слова раздела [NH_TCPIP]

nh_server_addr

Возможные значения	<адрес1>,<адрес2>
Описание	Устанавливает IP-адреса всех HASP LM. Возможно использование неограниченных адресов и множественных строк. Пример формата адреса: IP-адрес: 192.114.176.65 Имя локального узла: ftp.aladdin.co.il

nh_server_name

Возможные значения	<имя1>,<имя2>
Описание	Обменивается данными с HASP LM с определенным именем. Максимально – 6 имен; каждое имя может состоять максимально из 7 символов.

nh_port_number

Возможные значения	<номер>
Описание	Устанавливает номер порта TCP/IP (опционально).
Значение по умолчанию	475

nh_tcpip_method

Возможные значения	TCP, UDP
Описание	Посылает пакет TCP или UDP.
Значение по умолчанию	UDP

nh_use_broadcast

Возможные значения	включено, выключено
Описание	Использовать широковещательный механизм UDP.
Значение по умолчанию	включено

Общие ключевые слова раздела [NH_TCPIP]

nh_session

Возможные значения	<число>
Описание	Определяет максимальный период времени, в течение которого защищаемое приложение пытается установить связь с HASP LM.
Значение по умолчанию	2 секунды

nh_send_rcv

Возможные значения	<число>
Описание	Устанавливает для HASP LM максимальное время получения или отправки пакета.
Значение по умолчанию	1 секунда

Мониторинг лицензий HASP4 Net

Утилита Aladdin Monitor позволяет осуществлять централизованное администрирование приложения HASP LM и ключей HASP4 Net.

Доступны следующие возможности:

- Проверка свойств HASP LM.
- Проверка ключей HASP4 Net.
- Запуск и прерывание сервиса HASP LM.

Aladdin Monitor предназначена для работы в следующих средах: Windows 98/ME, Windows NT/2000/XP. Программой поддерживаются протоколы TCP/IP и IPX.

Передача Aladdin Monitor

Утилиту Aladdin Monitor, снабженную файлом со справочной информацией, следует предоставлять клиентам.

Настройка Aladdin Monitor может осуществляться при помощи файла конфигурации HASP4 Net (см. раздел «Настройка клиентов HASP4 Net» на стр. 263).

Установка Aladdin Monitor

Установка утилиты Aladdin Monitor может осуществляться на любой станции в пределах сети. Устанавливать на этой же станции HASP LM не требуется.

Для установки Aladdin Monitor следует использовать установочную утилиту *aksmon.exe*, следуя инструкциями мастера установки.

Настройка Aladdin Monitor

Изменяться могут следующие программные настройки:

- Используемый язык (немецкий или английский).
- Частота обновления диалогового окна (значение по умолчанию – 2 секунды).
- Частота запросов (значение по умолчанию – каждые три минуты).
- Использование режима Hardlock, режима HASP или и того и другого.
- Использование файла конфигурации *nethasp.ini*.

Для изменения настроек следует выбрать пункт **Settings** в меню **File**. Изменения вступают в силу после следующего запуска приложения.

Проверка свойств HASP License Manager

В левой части окна выберите HASP LM, информацию о подключении которого вы хотите проверить.

Если HASP LM не отображается, следует выбрать папку HASP LM или обновить отображаемую информацию, выбрав File/Rescan.

Информация о HASP LM отображается в правой части окна.

HASP LM, которые работают только с протоколом NetBIOS, не распознаются утилитой Aladdin Monitor.

О выбранном HASP LM отображается следующая информация:

- Общая информация (табл. 21.1)
- Информация об управляемых ключах HASP4 Net (табл. 21.2)

Таблица 21.1. Информация о HASP License Manager

Поле	Значение
Name	Имя компьютера, на котором запущен HASP LM.
Version	Версия HASP LM.
IP	IP-адрес компьютера.
IPX	IPX-адрес компьютера.
LM Type	Версия HASP LM.
TCP/IP, IPX	Используемый протокол.

Таблица 21.2. Информация о ключах HASP

Поле	Значение
HASP #	Кумулятивный номер ключа HASP.
HASP Model	Максимально возможное число лицензий.
Current Station	Подключенные в настоящий момент станции.

Проверка ключей HASP

В левой части окна следует выбрать ключ HASP, информацию о котором необходимо проверить. Информация о ключе HASP может быть проверена только в том случае, если было осуществлено подключение.

Если ключ не отображается, следует дважды щелкнуть мышью по HASP LM, который работает с данным ключом, либо обновить информацию в окне, выбрав File/Rescan.

О выбранном ключе HASP в правой части окна отображается следующая информация:

- Общая информация о ключе HASP (табл. 21.3).
- Обзор программы (табл. 21.4).
- Обзор подключений отдельных программ (табл. 21.5).

Таблица 21.3. Информация о ключе HASP

Поле	Значение
HASP #	Кумулятивный номер ключа HASP.

Таблица 21.4. Таблица программ

Поле	Значение
Program No.	Номер, присвоенный защищаемой программе.
Current Stations	Подключившиеся в настоящий момент станции.
Maximum Stations	Максимально возможное число станций.
Activations	Максимально возможное число активаций.

Таблица 21.5. Таблица подключений

Поле	Значение
No.	Кумулятивное число подключений.
Login ID	Адрес подключившейся станции.
Protocol	Используемый протокол.
Timeout	Период времени, после которого станция будет удалена из списка подключений (в секундах).

Запуск и прекращение работы HASP License Manager в виде сервиса

Сервис HASP LM позволяет осуществлять администрирование ключей HASP4 Net на рабочей станции NT. Для запуска или прекращения работы сервиса HASP LM на локальной машине можно использовать утилиту Aladdin Monitor.

Запуск сервиса

Выберите **Start HASP LM Service** в меню **HASP LM Service** или в меню **Service/HASP**. Кроме этих двух способов, вы можете запустить сервис, используя символ трафика. После этого сервис будет запущен, что сделает присоединенные к сетевым станциям ключи HASP4 Net доступными.

Вы также можете запустить сервис при помощи контекстного меню. Для этого вам следует щелкнуть правой кнопкой мышки на папке **HASP LM** и выбрать **Start HASP LM**.

Прекращение работы сервиса

Выберите **Stop HASP LM Service** в меню **HASP LM Service** или в меню **Service/HASP**. Кроме этих двух способов, вы можете прекратить работу сервиса, используя символ трафика.

После этого сервис прекратит работу, а отображаемая информация обновится, что может потребовать определенного времени, поскольку для этого необходим опрос сети.

Вы также можете прекратить работу сервиса при помощи контекстного меню. Для этого вам следует щелкнуть правой кнопкой мышки на папке **HASP LM** и выбрать **Stop HASP LM**.

Адаптация HASP Net к сети

Определение области станций при использовании протокола IPX

В данном разделе рассматриваются дополнительные ключи HASP LM и ключевые слова для файла *nethasp.ini*, которые могут использоваться для адаптации системы HASP4 Net к сетевой среде.

При использовании протокола IPX можно позволить сетевым станциям, принадлежащим различным сегментам сети, получать доступ к HASP LM.

1. Для этого следует:

   Загрузить HASP LM с ключом –ipxnosap. Это обеспечит доступ к HASP LM через файл *newhaddr.dat*.

2. Внесите следующие изменения в файл *nethasp.ini*:

   - В разделе [NH_COMMON] укажите NH_IPX=Enabled
   - В разделе [NH_IPX] укажите NH_USE_BROADCAST=Disabled
   - В разделе [NH_IPX] укажите NH_USE_BINDERY=Disabled

   Такие настройки обеспечат то, что защищаемое приложение будет осуществлять поиск адресного файла и считывать адрес HASP LM.

3. Скопируйте защищаемое приложение и файл *nethasp.ini* в одну директорию. Убедитесь в том, что каждая станция, которая входит в заданную область, загружает приложение из этой директории.

Определение области станций при использовании протокола TCP/IP

При использовании протокола TCP/IP существует два способа определения области станций. Вы можете задать область станций, которую будет обслуживать HASP LM, либо вы можете определить, что данная область станций будет искать определенный HASP LM.

Определение области с использованием *nhsrv.ini*

HASP LM для Windows и Win32 могут читать файл *nhsrv.ini*. Для определения области станций, которую будет обслуживать HASP LM, в этот файл следует внести следующие изменения:

- В разделе [NHS_SERVER] следует указать NHS_IP_LIMIT=<ipaddr> [,<ipaddr...]

Примеры форматов <ipaddr>

При определении области станций при помощи файла *nhsrv.ini* можно использовать следующие форматы:

- 10.1.2.3

  При этом HASP LM будет обслуживать только станцию с этим IP-адресом.

- 10.1.2.*

  HASP LM будет обслуживать станции, отвечающие указанной маске IP-адресов, т.е. с 10.1.2.0 до 10.1.2.255.

- 10.1.*.*

  HASP LM будет обслуживать станции, отвечающие указанной маске IP-адресов, т.е. с 10.1.0.0 до 10.1.255.255.

Вы можете исключить определенные IP-адреса при помощи знака «!». Например, вы можете указать !10.1.2.7.

Для того чтобы разрешить доступ к HASP4 Net только нескольким станциям:

1. Отредактируйте файл *nhsrv.ini* и укажите в нем область станций.
2. Скопируйте файл *nhsrv.ini* в такое место, где к нему может получить доступ HASP LM.

Определение области с использованием *nethasp.ini*

Файл конфигурации HASP4 Net можно отредактировать таким образом, чтобы станции из заданной области осуществляли поиск HASP LM (по соответствующему адресу).

Для этого следует:

1. Отредактируйте файл *nethasp.ini*: В разделе [NH_TCPIP] укажите NH_SERVER_ADDRESS=<адрес HASP LM>
2. Скопируйте файл *nethasp.ini* в место, доступ к которому могут получить только станции, принадлежащие к заданной области.

Определение области станций при использовании протокола NetBIOS

Для того чтобы разрешить доступ к ключу только некоторых сетевых станций:

1. Загрузите HASP LM с ключом –**nbname** и указанным по выбору именем (до восьми символов, символы не зависят от регистра). При этом будет определено имя NetBIOS для HASP LM.

   Например, чтобы загрузить *haspserv.exe* и определить имя firstserv, следует ввести **haspserv –nbname=firstserv**

2. Внесите следующие изменения в файл *nethasp.ini*:

   - В разделе [NH_COMMON] укажите NH_NETBIOS=Enabled
   - В разделе [NH_NETBIOS] укажите NH_NBNAME=firstsrv

   Это свяжет станцию с именем NetBIOS.

3. Скопируйте защищаемое приложение и файл *nethasp.ini* в одну директорию. Убедитесь в том, что каждая станция, которая входит в заданную область, загружает приложение из этой директории.

Нельзя задать одинаковое имя NetBIOS двум станциям. Если вы хотите загрузить HASP LM более чем на одной станции, следует присвоить различные имена NetBIOS каждому HASP LM.

Настройка времени ожидания

HASP LM не может обрабатывать более одного запроса одновременно. Время ожидания задает период времени, в течение которого защищаемое приложение будет пытаться получить доступ к HASP LM.

Практически во всех сетях значения времени ожидания примерно равны, поэтому изменять этот параметр следует только в том случае, если HASP4 Net присоединен к медленной или загруженной сетевой станции.

Для определения времени ожидания:

В соответствующем разделе файла *nethasp.ini* укажите:

```
NH_SESSION=<m>
NH_SEND_RCV=<n>
```

где m и n измеряются в секундах. По умолчанию m=2, n=1.

Определение числа обслуживаемых защищаемых приложений

HASP LM позволяет изменить число обслуживаемых защищаемых приложений. По умолчанию HASP LM обслуживает 250 (NLM) или 1000 (Win32, Mac) защищаемых приложений.

От максимального числа обслуживаемых приложений зависит загрузка памяти. При необходимости память можно освободить, изменив значение данного параметра по умолчанию.

Для изменения числа обслуживаемых приложений:

Загрузите HASP LM с соответствующим ключом:

```
nhsrvw32 –userlist=n
```

где n – число обслуживаемых защищаемых приложений. Ключ **userlist** может использоваться только с приложениями Win32.

Часть 5
Использование системы дистанционного перепрограммирования

В данной части содержится информация по системе дистанционного перепрограммирования (Remote Update System – RUS). Эта система позволяет обновлять память тех ключей, которые установлены у ваших клиентов. Глава «Система дистанционного перепрограммирования» на стр. 289 описывает концепции RUS и соответствующие инструменты.

Глава «Интерфейс Win32 API для системы дистанционного перепрограммирования» на стр. 311 описывает API, который можно использовать для написания соответствующих приложений.

Система дистанционного перепрограммирования

Система дистанционного перепрограммирования (Remote Update System – RUS) HASP – это специальная утилита для безопасного удаленного обновления памяти ключей ваших клиентов.

RUS позволяет вам обновлять память тех ключей HASP, которые уже переданы вашим клиентам. Вы можете обновить информацию (зашифровать дополнительные данные, внести другие изменения) и отослать эту информацию вашим клиентам по электронной почте, факсу или телефону.

Таким образом, вы можете включать дополнительные защищенные модули, модифицировать уже распространенное программное обеспечение и выполнять другие сходные операции. Например, вы можете отослать новые параметры вашим клиентам, чтобы перевести ваше приложение из демонстрационного режима в полнофункциональный.

С помощью RUS вы можете менять параметры ключей HASP4 M1, M4, HASPNet и HASP4 Time.

Применение RUS

Применение RUS включает в себя две стадии:

- Создание утилит RUS
- Обновление данных ключей HASP ваших клиентов

Утилиты RUS

С помощью RUS вы можете создать две утилиты:

- Утилиту производителя
- Утилиту клиента

Утилита производителя предназначена для использования в вашей компании. Утилиту клиента необходимо переслать покупателям защищаемого приложения.

Чтобы обновить HASP, вы и ваш клиент должны каждый использовать свою утилиту RUS.

Процедура обновления

Чтобы обновить память HASP:

1. Клиент использует утилиту клиента, чтобы получить ID-номер своего ключа HASP, который необходим для обновления информации, и отправить этот номер вам.
2. Введите полученный ID-номер и обновленные данные в утилиту производителя.
3. Сгенерируйте пароли RUS в утилите производителя и сообщите их клиенту.
4. Ваш клиент должен ввести пароли RUS в утилите клиента и обновить данные ключа HASP.

Операция по обновлению данных, содержащихся в ключах клиента, абсолютно надежна. Все данные, передаваемые вами клиенту, зашифрованы. Кроме того, пароли RUS зависят от номера ключа, для которого они были сгенерированы, поэтому никто не сможет использовать эти данные для обновления другого ключа.

Стадии RUS

Иллюстрации, приведенные ниже, показывают две стадии удаленного перепрограммирования ключей HASP:

Рис. 24.1. Стадия 1: Создание утилит RUS

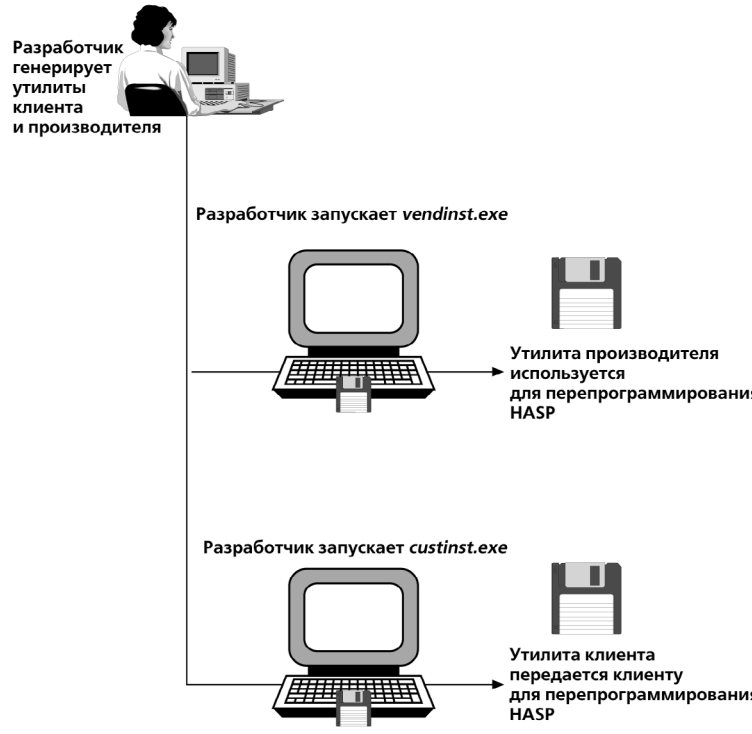
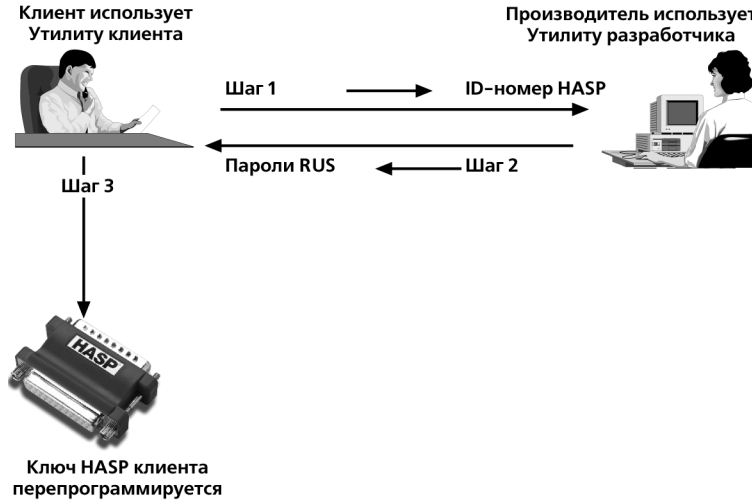


Рис. 24.2. Стадия 2: Обновление данных HASP



Создание утилит RUS

При создании двух утилит RUS (одна для вас, а вторая для клиента) вы должны использовать две инсталляционные программы RUS: *vendinst.exe* и *custinst.exe*.

Программа *vendinst.exe* генерирует утилиту производителя, *vendor.exe*, а программа *custinst.exe* – утилиту клиента, *customer.exe*.

Программы *vendinst.exe* и *custinst.exe* располагаются в каталоге RUS. Этот каталог содержит два подкаталога. Каталог IBM содержит бинарные файлы, необходимые для создания утилит RUS для IBM-совместимых компьютеров. Каталог NEC содержит бинарные файлы, необходимые для создания утилит RUS на компьютерах NEC. Перед созданием утилит, скопируйте необходимые программы в каталог RUS.

Вы должны будете отправить файлы *customer.exe* и *customer.hlp* вашим клиентам.

Генерирование утилит RUS

При генерировании утилит RUS убедитесь, что файлы *vendinst.exe, vendor.bin, custinst.exe* и *customer.bin* находятся в одном и том же каталоге:

- В случае если вы хотите создать утилиты RUS для IBM-совместимых компьютеров, скопируйте содержимое каталога IBM в корневой каталог RUS.
- В случае, когда вам необходимо создать утилиты RUS для NEC, скопируйте содержимое каталога NEC в корневой каталог RUS.

Чтобы сгенерировать утилиты RUS:

1. Подключите ключ приложения HASP к вашей машине.
2. Введите следующую команду из командной строки DOS, чтобы сгенерировать утилиту производителя:

   ```
   vendinst <password1> <password2> [target pc]
   ```

3. Введите следующую команду из командной строки DOS, чтобы сгенерировать утилиту клиента:

   ```
   custinst <password1> <password2> [target pc]
   ```

Вы сгенерировали утилиты.

Так как утилиты были сгенерированы с использованием паролей RUS, уникальных для каждого ключа (или нескольких ключей с одинаковым кодом разработчика), они работают лишь для обновления данных конкретного ключа (или нескольких ключей с одинаковым кодом разработчика).

Параметры инсталляции RUS

При создании утилит RUS вы задаете следующие параметры:

password1, password2

Эти пароли принадлежат ключу HASP, для которого создаются утилиты.

target pc

Необязательный ключ, который указывает тип компьютера, на котором должны работать утилиты RUS:

Ключ	Описание
-ibm	тип компьютера, на котором должны работать утилиты RUS - IBM PC. Это значение по умолчанию.
-nec	тип компьютера, на котором должны работать утилиты RUS – NEC.

Утилита производителя

Используйте утилиту производителя, чтобы запрограммировать обновления ключей HASP клиента. Эти данные передаются в форме паролей RUS, которые вы генерируете и передаете своему клиенту.

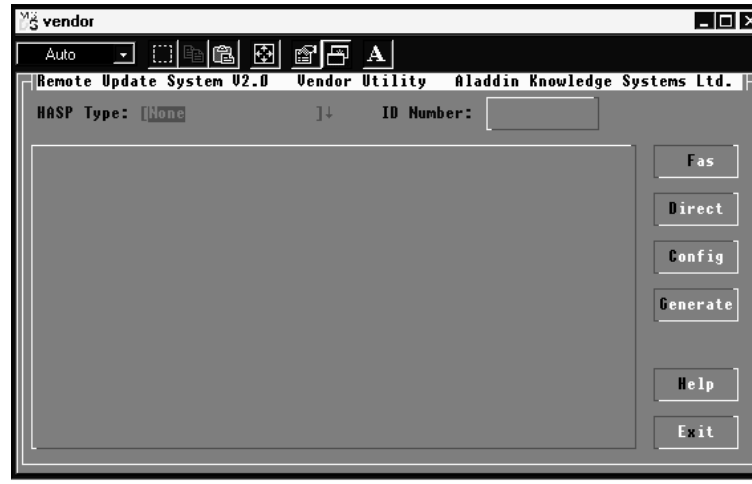
Для получения информации о том, каким образом клиент обновляет данные своего ключа, используя эти пароли, смотрите раздел «Утилита клиента» на стр. 304.

Генерирование паролей RUS

Чтобы сгенерировать пароли RUS:

1. Запустите утилиту производителя (напечатайте в командной строке DOS *vendor.exe*, чтобы запустить приложение).

   На экране появится главное окно утилиты:



2. Подведите курсор мыши к правой стороне поля **HASP Type.** В появившемся списке выберите модель ключа HASP, которая установлена у вашего клиента.
3. Введите ID-номер ключа вашего клиента в поле **ID Number**.

Этот номер вам должен сообщить клиент, который использует утилиту клиента. Для получения более полной информации следует обратиться к информации раздела «Утилита клиента» на стр. 304.

4. Введите обновленные данные. Есть два способа ввода этих данных: режим FAS и режим прямого ввода.

   - Нажмите **FAS**, чтобы ввести параметры FAS. На экране появятся параметры FAS для той модели HASP, которую вы выбрали ранее.
   - Нажмите **Direct**, чтобы отредактировать память HASP.

   Во время одной сессии RUS вы можете выбрать один из методов обновления данных, но не оба.

5. Нажмите **Config**, если хотите использовать файл конфигурации. В открывшемся окне вы сможете:

   - Сохранить установленные параметры в файле конфигурации.

   Введите имя файла или выберите файл, нажав **Browse**, затем нажмите **Save**.

   - Загрузить предварительно сохраненные параметры из файла конфигурации.

   Введите имя файла или выберите файл, нажав **Browse**, затем нажмите **Load**.

6. Нажмите **Generate**, чтобы сгенерировать пароли RUS, затем нажмите **OK**, чтобы отобразить пароли RUS на экране. Может быть создано до 11 паролей RUS.

   Вы можете сохранять пароли RUS в файле, отметив поле **Save RUS passwords to a File**, введя имя файла, а затем нажав **OK**. Этот файл может быть в дальнейшем использован утилитой клиента, чтобы загружать пароли RUS автоматически.

   Теперь вам необходимо сообщить пароли RUS клиенту. Клиент использует эти пароли для обновления памяти HASP.

Ввод данных в режиме FAS

Используя параметры FAS, вы сможете установить дополнительные параметры защиты вашего приложения (приложений).

В утилите RUS можно установить следующие параметры FAS:

Номер программы

Каждая защищаемая программа имеет свой уникальный номер. В случае обновления защитных параметров программы укажите номер, который ей был присвоен изначально. В случае, если вы добавляете новую защищаемую программу, присвойте ей новый номер.

Вы можете устанавливать параметры FAS только для одного приложения за одну сессию RUS. Пароли RUS для каждой из защищаемых программ генерируются отдельно для каждой программы.

Не вводите значения большие, чем максимально допустимое количество защищаемых программ для конкретного ключа HASP. Ниже приведены корректные интервалы значений для различных ключей:

- 1-16 для HASP4 M1
- 1-112 для HASP4 M4
- 1-8 для HASP4 Time
- 1-112 для HASP4 Net

Количество авторизованных запусков

Введите в это поле максимальное количество разрешенных запусков приложения. Если вы не хотите ограничивать этот параметр, выберите **U**. Этот параметр доступен для ключей HASP4 M1, M4 и HASP4 Net.

Дата окончания лицензии

Этот параметр используется только для ключа HASP4 Time. Он означает дату, после которой программа не будет запускаться. Введите двухцифровые числа для дня, месяца и года (дата окончания). Введите «00-00-00» если вы не хотите ограничивать использование приложения по времени.

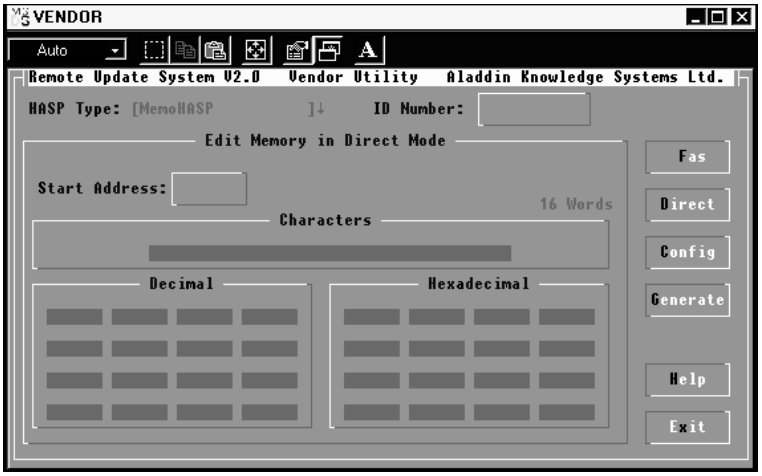
Количество авторизованных станций

Этот параметр используется только для ключа HASP4 Net. Он указывает то количество станций, на котором разрешен одновременный запуск приложения. Присвойте этому параметру значение, не превосходящее количество

поддерживаемых станций выбранной моделью ключа HASP (например, 5 для HASP4 Net5, 10 для HASP4 Net10 и т.д.). Если вы используете ключ HASP4 NetU, вы можете ввести **U,** чтобы разрешить одновременный запуск с неограниченного количества станций.

Ввод данных в режиме прямого ввода

В этом режиме вы сможете редактировать до 16 последовательно расположенных ячеек памяти. Окно редактирования памяти в режиме прямого доступа выглядит следующим образом:



Введите данные в следующие поля:

Start Address

Вы начнете редактирование памяти HASP с этого адреса. Введите десятичное значение. Не оставляйте это поле пустым!

Data Cells

16 ячеек памяти отображаются в трех режимах (Characters, Decimal и Hexadecimal), каждый в отдельной панели. Введите данные в любую из трех панелей, используя формат этой панели. Ввод данных в одной панели автоматически вызовет заполнение ячеек отображения в других панелях с соответствующими значениями.

Чтобы перемещаться между панелями, щелкните по любой ячейке в выбранной панели.

Вы можете ввести данные не во все 16 ячеек, а только в часть их. Тем не менее, не оставляйте незаполненные поля между полями с введенными значениями.

Активация утилиты производителя

Вы можете запускать утилиту производителя из командной строки DOS, используя ключи командной строки. Эта возможность поможет вам автоматизировать процесс запуска утилиты с помощью командного файла либо сэкономить время выполнения задания.

Ключи командной строки

В таблице ниже приведены возможные ключи командной строки утилиты и краткое описание каждого из них. При использовании командной строки вы можете вводить как полное название ключа, так и его аббревиатуру из заглавных букв. Например, вы можете использовать либо **-HT** либо – **HaspType.**

Таблица 24.1. Ключи командной строки утилиты производителя

Ключ	Описание
-Help	Выводит список ключей и их краткое описание.
-HaspType <HASP type>	Задает тип ключа HASP. Возможные значения: memo для HASP4 M1, M4 time для HASP4 Time net для HASP4 Net
-IDnum <HASP ID number>	Шестнадцатеричное значение ID-номера ключа HASP клиента.
-CfgFile <configuration filename>	Полный путь к файлу конфигурации, в котором содержится часть или все параметры, установленные и сохраненные в утилите производителя ранее.
-DiRect	Указывает, что данные редактируются в режиме прямого доступа.
-Fas	Указывает, что данные редактируются в режиме FAS.
-GenPassFile <file-name>	Имя файла паролей RUS.
-PrgNum <program number>	Номер, присвоенный защищаемой программе.

Ключ	Описание
-STations <number of stations>	Максимальное количество станций, на котором разрешен одновременный запуск приложения. Этот параметр используется только для ключей HASP4 Net
-ACtivations <number of activations>	Максимальное количество активаций приложения. Введите U, чтобы не ограничивать этот параметр. Этот параметр используется только для ключей HASP4 M1, M4 и HASP4 Net.
-ExpDate <mm-dd-yy>	Дата окончания лицензии. Введите 00-00-00, чтобы не ограничивать этот параметр. Этот параметр используется только для ключей HASP4 Time.
-StartAddr <start address>	Стартовый адрес для редактирования памяти в режиме прямого доступа.
-DataD <dataD1>..<dataD16>	До 16 десятичных значений, которые вы хотите записать в память HASP
-DataH <dataH1>..<dataH16>	До 16 шестнадцатеричных значений, которые вы хотите записать в память HASP
-DataS <"string" >	Количество символов (до 32 для HASP4 M1, M4 и HASP4 Net, до 16 для HASP4 Time), которые вы хотите записать в память HASP. Вводите строку, используя символы апострофа.
-BatcH	Запускает утилиту в фоновом режиме без отображения окна утилиты.

Пример использования ключей командной строки

Пример, приведенный здесь, демонстрирует запуск утилиты производителя из командной строки:

```
vendor -bh -ht memo -id 7a37381e -f -pn 5 -ac
u –gpf ruspass.txt
```

Запускает утилиту производителя (vendor) в фоновом режиме (-bh). Тип ключа клиента - HASP4 M1, M4 (-ht memo), ID-номер этого ключа - 7a37381e (-id 7a37381e). Использовать FAS для обновления (-f), обновления касаются программы с номером 5 (-pn 5), количество запусков не ограничено (-ac u). Генерирует пароли RUS, сохраняя их в файле *ruspass.txt* (-gpf ruspass.txt)

Файл *ruspass.txt* может быть использован утилитой клиента, чтобы загружать пароли RUS автоматически.

```
vendor -ht net -dr -dd 12 15 25
```

Запускает утилиту производителя (vendor). Тип ключа клиента - HASP4 Net (-ht net). Режим прямого доступа памяти (-dr). Первые три ячейки памяти (-dd) получают десятичные значения 12, 15 и 25.

Коды ошибок утилиты производителя.

Утилита производителя возвращает коды ошибки уровня DOS. В таблице приведены краткие описания кодов ошибок:

Таблица 24.2 Коды ошибок утилиты производителя

Код	Описание
2, 5	Неправильный (или отсутствует) ID-номер.
3, 4	Неправильный (или отсутствует) тип ключа HASP.
7, 8	Неправильный (или отсутствует) номер программы для FAS.
9, 10	Неправильное (или отсутствует) количество активаций приложения.
11, 12	Неправильное (или отсутствует) количество авторизованных станций.
13, 14, 16	Неправильный (или отсутствует) стартовый адрес.
15, 20	Неправильный (или отсутствует) аргумент данных.
17, 18	Аргумент данных за пределами допустимых значений.
22	Нет строки данных.
23, 24	Строка данных слишком длинная.
25	Допускается только один ключ типа данных (DataD, DataH или DataS).
26, 27	Неправильная (или отсутствует) дата окончания действия лицензии.
29	Невозможно использовать ключ CfgFile внутри файла конфигурации.

Код	Описание
30	Отсутствует имя файла конфигурации.
31, 35	Ошибка открытия/чтения файла конфигурации.
32, 39	Невозможно создать/отсутствует файл паролей RUS.
33	Не хватает памяти для чтения файла.
34	Файл не является файлом конфигурации утилиты производителя.
36	Отсутствует ключ типа ввода данных (FAS/Direct).
40	Слишком много слов данных.
41	Неизвестный ключ.
52	Неправильный ID-номер.
55, 56	Невозможно использовать ключи FAS в режиме прямого доступа (и наоборот).
59	Допустимые значения количества активаций от 0 до 65534 либо U для нелимитированного количества.
60	Допустимые значения количества станций от 1 до 65534 либо U для нелимитированного количества.

Утилита клиента

Для того чтобы ваши клиенты могли обновлять данные своих ключей HASP, им необходимо предоставить утилиту клиента (*customer.exe*). Инструкции, приведенные ниже, иллюстрируют процесс использования этой утилиты клиентом.

Вы должны поставлять вашим клиентам эти инструкции наряду с файлом справки *customer.hlp*.

Использование утилиты клиента

Использование данной утилиты включает в себя две стадии:

- Получение ID-номера HASP
- Обновление данных на ключе HASP

Получение ID-номера HASP

На этом этапе вы проверяете ID-номер вашего ключа HASP и предоставляете информацию о номере своему производителю.

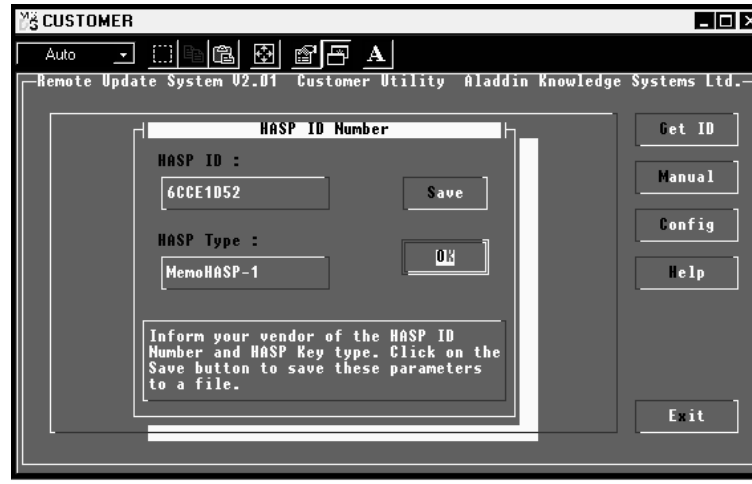
Получение ID–номера HASP:

1. Подключите ключ HASP к вашему компьютеру.
2. В командной строке DOS введите команду:

   `customer`

   На экране появится окно утилиты клиента.
3. Нажмите **Get ID**, чтобы получить ID-номер HASP (и увидеть тип ключа HASP).

   На экране появится диалог с ID-номером и типом ключа:



4. Если вы хотите сохранить эти два параметра в файл конфигурации, нажмите **Save** и введите имя файла, либо нажмите **Browse** и выберите файл на диске. Затем нажмите **OK**.

Передайте полученную информацию (ID-номер вашего ключа HASP и его тип) своему производителю напрямую либо перешлите ему свой конфигурационный файл. С помощью этих данных производитель сможет сгенерировать пароли RUS, которые необходимы вам для обновления вашего ключа HASP.

Обновление вашего ключа HASP с помощью паролей RUS

На этом этапе вы уже должны получить пароли RUS от своего производителя. Эти пароли содержат данные, которые будут сохранены в память вашего ключа HASP.

Чтобы обновить ключ HASP с помощью паролей RUS:

1. Подключите ключ HASP к вашему компьютеру.
2. В командной строке DOS введите команду:

   customer

   На экране появится окно утилиты клиента.

3. Введите пароли RUS, которые вы получили от своего производителя, в утилиту клиента одним из следующих способов:

   • Нажав **Manual** и введя пароли вручную в появившемся диалоге:

   • Нажав **Config**, введя имя файла конфигурации, содержащего пароли RUS (или нажав **Browse**, чтобы найти его), затем нажав **OK**. Эта процедура загружает пароли автоматически из конфигурационного файла, предоставленного вам вашим производителем.

   На экране появятся пароли RUS.

4. Нажмите **Update** после ввода паролей, чтобы обновить данные вашего ключа HASP.

Запуск утилиты клиента с помощью ключей командной строки

С помощью ключей командной строки вы можете сэкономить время выполнения процедуры обновления. Эти ключи дают вам возможность исполнять утилиту клиента из командных файлов или из ваших собственных приложений, поскольку отпадает необходимость проходить все экраны утилиты.

Ключи командной строки утилиты клиента

В таблице ниже приведены возможные ключи командной строки утилиты и краткое описание каждого из них. При вводе вы можете использовать как полное название ключа, так и его аббревиатуру из заглавных букв. Например, вы можете использовать либо **–UP**, либо **–UPdate.**

Таблица 24.3. Ключи командной строки утилиты клиента

Ключ	Описание
-Help	Выводит список ключей и их краткое описание
-GetID	Получить ID-номер ключа HASP
-VenFile <filename>	Имя файла, в который будут сохранены данные об ID-номере ключа HASP
-UPdate <pass1>..<pass11>	Загружает пароли в память ключа. Одновременно могут быть загружены до 11 паролей
-CfgFile <configuration filename>	Полный путь и имя файла конфигурации, содержащего пароли RUS
-BatcH	Запускает утилиту в фоновом режиме без отображения окна утилиты
- NoBatch	Отменяет режим BatcH (например, во время загрузки паролей RUS из файла конфигурации)

Примеры использования ключей командной строки

Ниже приводятся примеры использования утилиты клиента с помощью ключей командной строки:

```
customer -bh -cf ruspass.txt
```

Запускает утилиту клиента (customer) в фоновом режиме (-bh) без показа экранов утилиты. Обновляет ключ HASP, записывая в него пароли, содержащиеся в файле конфигурации *ruspass.txt* (-cf ruspass.txt).

```
customer -cf ruspass.txt –nb
```

Запускает утилиту клиента (customer), входит в режим показа экранов (-bh). Обновляет ключ HASP, записывая в него пароли, содержащиеся в файле конфигурации *ruspass.txt* (-cf ruspass.txt).

Коды ошибок утилиты клиента.

Утилита клиента возвращает коды ошибки уровня DOS. В таблице приведены краткие описания кодов ошибок:

Таблица 24.4 Коды ошибок утилиты клиента

Код	Описание
1, 2	Неправильные (или отсутствуют) пароли RUS.
4	Отсутствует имя файла конфигурации.
6	Ошибка открытия файла.
7	Неверный файл конфигурации.
8	Переполнение памяти в момент чтения файла конфигурации.
12	Неверный ключ HASP.
13	Невозможно сгенерировать файл без ID-номера.
14	Отсутствует имя файла для записи ID-номера.
18	Ошибка открытия или записи файла для записи ID-номера.
19	Неизвестный ключ.

Код	Описание
101	Не найден ключ HASP.
104	Неправильный ID-номер HASP.
106	Неправильный стартовый адрес.
107	Слишком много слов данных.
108	Неправильные пароли RUS.
109	Ошибка чтения памяти HASP.

Интерфейс Win32 API системы дистанционного перепрограммирования

С помощью RUS Win32 API вы можете удаленно обновлять память ключей HASP4 M1, M4, HASP4 Time и HASP4 Net.

Такое обновление генерирует защищенный и зашифрованный набор строк, которые пересылаются пользователю и корректно интерпретируются только конкретным ключом HASP.

Каталог RUS API на диске HASP CD содержит следующие файлы:

haspdev.dll

Библиотека для разработчика (производителя). Эта библиотека предлагает две функции для создания строк обновления памяти ключей HASP: одна функция для общего обновления памяти, а другая для обновления памяти FAS.

haspclnt.dll

Библиотека разработчика (производителя). Эта библиотека предлагает две функции для создания строк обновления памяти ключей HASP: одна функция для интегрирования обновлений памяти в ключи HASP клиента, другая – для получения ID-номеров ключей HASP.

Утилита настройки

Утилита настройки (confdll.exe) является приложением командной строки для интегрирования паролей HASP в код динамической библиотеки клиента, а также для создания HASP Envelope.

Примеры приложений

Remote Update System Win32 API содержит примеры приложений, чтобы помочь вам в разработке собственного кода для интеграции RUS в ваше приложение.

Следующие разделы описывают методы реализации Win32 API в ваших приложениях, а также методы обновления памяти с помощью Win32 API

Реализация RUS

Перед обновлением памяти ключей с использованием Win32 RUS API вы должны ознакомиться с концепциями RUS. Для этого вам необходимо ознакомиться с примерами приложений.

Реализация RUS включает в себя следующие стадии:

Стадия 1: Подготовка вашего приложения для RUS

1. Создайте экземпляр *haspclnt.dll* с вашими паролями HASP. Эта библиотека будет поставляться совместно с вашим приложением (см. раздел «Утилита настройки» на стр. 321).
2. Создайте клиентские процедуры обновления RUS в вашем приложении (см. примеры приложений на стр. 312).

Стадия 2: Осуществление перепрограммирования

1. Используйте *haspclnt.dll* (DLL клиента) чтобы получить ID-номер того ключа, который будет обновляться.
2. Используйте *haspdev.dll* из среды разработки для создания необходимых строк обновления.

   Пошлите обновленные данные вашему клиенту.

Функции, предоставляемые библиотекой производителя

Библиотека *hasdevdll* предоставляет две функции:

- signed int RUS_CreateUpdateDirect для общего обновления памяти и
- signed int RUS_CreateUpdateFAS для обновлений памяти FAS.

Обе функции создают строки обновления, которые должны быть пересланы клиенту для обновления памяти его ключа HASP.

RUS_CreateUpdateDirect

Описание

При выполнении этой функции не нужен ключ HASP.

Синтаксис

```
signed int RUS_CreateUpdateDirect (
DWORD IdNum,
int Password1,
int Password2,
int KeyType,
int Address,
int NumOfBytesToUpdate,
char *MemoryImage,
int CodeBufferSize,
char *Code,
void *sKey)
```

Используемые параметры:

IdNum	ID-номер ключа HASP. Введите уникальный ID-номер ключа вашего клиента. Если вы хотите генерировать данные обновления без проверки ID-номера, введите 0. С этим значением будут обновляться все ключи, пароли которых совпадают с указанными ниже.
Password1, Password2	Пароли обновляемого ключа HASP.

KeyType	Тип ключа HASP. Возможные типы - HASP4 M1, M4, HASP4 Net или HASP4 Time.
Address	Начальный адрес сегмента обновляемой памяти HASP.
NumOfBytesToUpdate	Количество обновляемых байтов.
MemoryImage	Указатель на данные, которые вы хотите занести в ключ HASP.
CodeBufferSize	Длина буфера.
Code	Указатель на область данных, содержащих ASCIIZ-строку (ASCII строку, оканчивающуюся 0) сгенерированных кодов.
sKey	Параметр зарезервирован. Должен быть всегда равен NULL.
Возвращаемые значения	См. раздел «Возвращаемые значения» на стр. 319

RUS_CreateUpdateFAS

Описание

При выполнении этой функции не нужен ключ HASP. Эта функция доступна для ключей HASP4 Net, HASP4 Time и HASP4 M1, M4.

Синтаксис

```
signed int RUS_CreateUpdateFAS (
DWORD Idnum,
int Password1,
int Password2,
int KeyType,
int ProgramNumber,
int Activations,
int Year,
int Month,
int Day,
int Stations,
int CodeBufferSize,
char *Code,
void *sKey)
```

Используемые параметры:

IdNum	ID-номер ключа HASP. Введите уникальный ID-номер ключа вашего клиента. Если вы хотите генерировать данные обновления без проверки ID-номера, введите 0. С этим значением будут обновляться все ключи, пароли которых совпадают с указанными ниже.
Password1, Password2	Пароли обновляемого ключа HASP.
KeyType	Тип ключа HASP. Возможные типы - HASP4 M1, M4, HASP4 Net или HASP4 Time.

ProgramNumber	Каждая программа должна иметь свой уникальный номер. В случае если вы обновляете параметры защиты конкретной программы, укажите ее номер. Если вы добавляете новую программу для защиты, присвойте ей новый номер.
Activations	Максимальное количество авторизованных запусков программы. Используется для обновления данных ключей HASP4 Net и HASP4 M1, M4. Установите значение 65535, если вы хотите предоставить возможность не лимитированного количества запусков программы.
Year, Month, Day	Дата окончания срока действия лицензии. Используется только для HASP4 Time. Значение года (year) должно быть от 1992 до 2091. Установите все параметры равными 0, если не хотите использовать это ограничение.
Stations	Максимальное количество станций для одновременного запуска приложения. Используется только для HASP4 Net. Установите этот параметр равным 0, если не хотите использовать ограничение.
CodeBufferSize	Длина буфера.
Code	Указатель на область данных, содержащих ASCIIZ-строку (ASCII строку, оканчивающуюся 0) сгенерированных кодов.
sKey	Параметр зарезервирован. Должен быть всегда равен NULL.
Возвращаемые значения	См. главу «Возвращаемые значения» на стр. 319.

Функции, предоставляемые библиотекой клиента

Библиотека *HASPCLNT.DLL* предоставляет функции, которые должны исполняться на компьютере клиента с подключенным ключом, подлежащим обновлению.

Эта библиотека предоставляет две функции:

- signed int RUS_ PerformUpdate для обновления памяти клиента HASP.
- signed int Get_KeyID для получения ID-номера подключенного на момент исполнения функции ключа клиента.

Перед вызовом одной из этих функций DLL должна быть модифицирована с помощью утилиты настройки (см. раздел «Утилита настройки» на стр. 321).

RUS_PerformUpdate

Описание

Эта функция обновляет память одного или нескольких ключей HASP. В случае если обновление зависит от ID-номера, проверяются на корректность значения ID-номера, тип ключа и пароли. В противном случае проверяются только тип ключа и пароли.

Синтаксис

```
signed int RUS_PerformUpdate (char *Code)
```

Используемые параметры:

Code — Строка данных для обновления в формате ASCIIZ (сгенерированная с использованием функциональности библиотеки производителя).

Возвращаемые значения

См. раздел «Возвращаемые значения» на стр. 319.

Get_KeyID

Описание:	Эта функция позволяет получить значение ID-номера подключенного ключа. Она очень похожа на соответствующую функцию из HASP API и добавлена в эту библиотеку из соображений целостности продукта.
Синтаксис	`signed int Get_KeyID (unsigned int *KeyID)`
Используемые параметры:	
KeyID	Возвращает значение ID-номера подключенного ключа.
Возвращаемые значения	См. раздел «Возвращаемые значения» на стр. 319.

Возвращаемые значения

Общие

SUCCESS

Выполнение прошло успешно.

HASP_ERROR_IN_LOW_WORD

Произошла непредвиденная ошибка. Код ошибки HASP API возвращается в нижнем слове.

OPERATION_FAILED

Операция не была завершена корректно.

DLL_NOT_CUSTOMIZED

Библиотека не была модифицирована.

RUS ID

HASP_NOT_FOUND

Ключ не найден.

UNDEFINED_HASP

Ключ HASP не определен

HASP_3_DETECTED

Найден ключ версии HASP 3.

BATTERY_DEAD_OR_MEMORY_CORRUPT

Не работает батарея ключа, либо память ключа повреждена.

RUS Update

INVALID_CODE

Неправильная строка указана в параметре Code.

INVALID_KEY_ID

Обновление зависит от ID-номера; ID-номер подключенного ключа неправильный.

KEY_NOT_FOUND

Не найден ключ HASP, отвечающий заданному паролю.

KEY_TYPE_MISMATCH

Подсоединенный ключ имеет отличный от указанного в процедуре вызова тип.

Memory Update

BUFFER_TOO_SMALL

Длина буфера, указанного в параметре CodeBufferSize, слишком мала.

ILLEGAL_MODULE_NUMBER

Номер программы не соответствует допустимому диапазону значений для указанного ключа.

INVALID_PARAMETER

По крайней мере один параметр неверен. Например, не указаны пароли.

ILLEGAL_KEYTYPE

Указан неизвестный тип ключа.

DETECTED_TAMPERING

Замечены несанкционированные манипуляции с лицензией

HASH_MEMORY_OVERFLOW

Буфер хэш-памяти слишком мал.

Утилита настройки

Каждый производитель генерирует свои уникальные библиотеки с необходимыми функциями дистанционного перепрограммирования. Такая библиотека содержит пароли HASP и защищена от декомпиляции с помощью HASP Win32 Envelope. Чтобы создать собственную модифицированную библиотеку, вы должны использовать простую утилиту командной строки (*confdll.exe*):

Синтаксис

```
confdll <DLL name> <Password 1> <Password2>
<Envelope>
```

Пример

```
confdll.exe C:\demo\haspclnt.dll 15417 9632
D:\demo\instw32.exe
```

В приведенном примере библиотека будет модифицирована для использования с паролями демонстрационного ключа HASP и защищена с помощью метода Win32 HASP Envelope

(файл *instw32.exe* на HASP CD).

Приложение А

Выявление неисправностей

Первая часть этого приложения предлагает перечень действий, который может помочь вам разрешить некоторые наиболее часто встречающиеся проблемы, с которыми вы можете столкнуться при использовании HASP. Вторая часть представляет собой список специфических проблем, с которыми можете столкнуться вы или ваши клиенты, а также возможные решения подобных проблем.

Линейка продуктов HASP отвечает самым высоким требованиям надежности и контроля качества. Тем не менее, как и любое другое периферийное устройство, HASP может не заработать на конкретном компьютере из-за проблем с аппаратным либо программным обеспечением. Это приложение может помочь вам разрешить подобные ситуации. Чтобы избежать трудностей, удостоверьтесь, что вы пользуетесь самыми новыми драйверами и приложениями HASP. Обращайтесь к вашему поставщику HASP за обновлениями.

В случае если вы не можете справиться с проблемой сами, сначала проверьте, как работают тестовые приложения, примеры программ и утилиты диагностики. Если это не поможет вам отыскать решение, обратитесь к вашему представителю HASP, имея на руках результаты работы этих приложений.

Перечень действий

В случае, когда один из ваших клиентов сообщает о проблеме, проверьте:

- Устраняется ли проблема заменой ключа HASP?

  Если да, то замените неработающий ключ новым.

- Подключен ли ключ HASP к порту?
- Подключен ли принтер к ключу HASP?

  Если да, и принтер работает нормально, то отключите его и проверьте, работает ли HASP теперь.

  Если да, то проверьте кабель принтера и удостоверьтесь, что он отвечает стандартам IEEE. Часто эта информация нанесена прямо на кабель.

  Если кабель отвечает стандартам, но HASP не работает, попробуйте использовать для HASP второй параллельный порт во избежание любых проблем, которые могут быть связаны с принтером.

  Если проблема не разрешается и теперь, попробуйте другой принтер или подключите принтер к другому порту.

- Есть ли проблемы с печатью?

  Если да, то попробуйте другой принтер на этом же компьютере, чтобы определить, зависит ли проблема от принтера. Запустите *hinstall* с ключами **-i -cnt=yes**.

- Нет ли вируса на компьютере?
- Повторяется ли проблема, если приложение запустить на другом компьютере такой же конфигурации?

Проблемы и их решения

Проблема: Ключ HASP подсоединен, но приложение не может найти его.

Решение: Хотя мы и приложили все усилия для создания очень надежного сообщения между компьютером и HASP, иногда вызов процедуры hasp() может оказаться неуспешным (либо сообщение может быть утеряно). Мы рекомендуем вызывать процедуру hasp() несколько раз перед тем как решить, что HASP не подключен.

Проблема: При попытке распечатать из защищенного приложения Windows, вы получаете сообщения об ошибке принтера.

Решение: Эта ситуация – результат конфликта доступа к принтеру и доступа к HASP. Чтобы избежать подобных конфликтов, установите в систему драйвер устройства HASP.

Проблема: HASP подсоединен к принтеру. При этом Windows 3.x выводит предупреждение о том, что принтер недоступен или происходит конфликт устройств.

Решение:

1. Выберите **Main** из Program Manager.
2. Выберите **Control Panel**.
3. Выберите **386 Enhanced**.
4. Выберите **LPT1** в фрейме Device Contention и нажмите **Never Warn**.
5. Повторите шаг 4 для LPT2 и LPT3.
6. Нажмите **OK**.

Проблема: Вы пытаетесь использовать *hinstall.exe* для инсталляции драйвера dHASP устройства HASP под Windows NT, но получаете ошибку 9121.

Решение: Вы получите эту ошибку в том случае, если попытаетесь запустить *hinstall.exe*, не обладая полномочиями администратора. Убедитесь, что вы обладаете полномочиями администратора системы.

Проблема: Вы пытаетесь запустить приложение, защищенное с помощью HASP4 M1/M4, под Windows NT/2000/XP и Windows 95/98/ME, но приложение не находит ключа HASP.

Решение: Убедитесь, что драйвер устройства HASP установлен. Если проблема не исчезает, запустите:

hinstall –info

Утилита Hinstall покажет номер версии, дату инсталляции и тип компьютера. Сообщите вашему представителю HASP эти параметры наряду с описанием проблемы.

Проблема: Ваше 16-разрядное приложение защищено с помощью HASP4 Net и HASP Envelope. Несмотря на то, что количество станций, запускающих приложение, меньше, чем максимально возможное количество лицензий, вы получаете ошибку «too many users» (слишком много пользователей).

Решение: 16-разрядное приложение, защищенное с помощью HASP4 Net и HASP Envelope, не выполняет процедуру HASP4 Net logout. Таким образом, при окончании работы приложения запись о нем остается в таблице лицензий ключа. Таким образом, одна и та же станция упоминается в таблице много раз.

Переделайте защиту, используя HASP API так, чтобы выполнять процедуру HASP4 Net logout. Таким образом, запись о приложении будет корректно удаляться из таблицы, а лицензии будут высвобождаться.

Проблема: Вы защитили приложение и для работы на локальной машине, и для работы в сети с использованием ключей HASP4 M1/M4 и HASP4 Net. После этого вы обнаружили, что максимальное количество станций, на которых может быть запущено приложение, превосходит количество лицензированных станций на единицу.

Решение: Эта ситуация возникает потому, что приложение пытается вначале получить доступ к локальному ключу HASP4 M1/M4. В случае если коды разработчика ключей HASP4 M1/M4 и HASP4 Net одинаковы, приложение, запущенное на станции с установленным ключом HASP4 Net, находит ключ HASP4 Net локально. Приложение считает данный ключ локальным и не записывается в таблицу HASP4 Net. Таким образом, данное приложение не использует сетевую лицензию.

Чтобы избежать подобной ситуации, сделайте так, чтобы ключи HASP4 M1/M4 и HASP4 Net имели разные коды разработчика.

Проблема: Ваше приложение, защищенное с помощью HASP4 Net для Windows, возвращает ошибку HASP4 Net LastStatus Error 21.

Решение: Приложения для надстроек DOS и Windows требуют 8кб памяти DOS. HASP4 Net API требует еще 1 Кб памяти DOS.

Ошибка HASP4 Net LastStatus Error 21 появляется тогда, когда доступно менее 1Кб памяти DOS, чего недостаточно для системы HASP4 Net. В этом случае не запускается не только приложение, защищенное с помощью HASP4 Net для Windows, но и другие приложения для Windows.

Чтобы разрешить эту проблему, либо удалите программы-резиденты, либо закройте некоторые приложения Windows. Это решение применяется для любого приложения, которое сообщает о недостаточности памяти DOS.

Проблема: Ваше приложение работает на станции, на которой не установлены драйверы сети. Компьютер зависает после того, как приложение выполняет процедуру HASP4 Net login.

Решение: Это происходит в тот момент, когда файл конфигурации HASP4 Net загружает указанный сетевой протокол. Система HASP4 Net пытается использовать указанный протокол без проверки на его присутствие в системе. В случае если запрашиваемый протокол не установлен на станции, станция перестает отвечать на запросы пользователя.

Для решения этой проблемы удалите файл конфигурации HASP4 Net. В случае если этот файл вам необходим, загрузите соответствующие драйверы протокола.

Проблема: После попытки доступа к параллельному порту, компьютер зависает.

Решение: Параллельные порты компьютеров IBM PC и совместимых присвоены одному из портов ввода/вывода: 3BCh, 378h или 278h.

Сетевые карты обычно нуждаются в 10h или 20h портах ввода/вывода одновременно, начиная с их базового адреса.

В том случае, если значения портов ввода/вывода сетевой платы пересекаются со значениями портов параллельного порта, попытка доступа к параллельному порту может вызвать зависание компьютера. Например: печать, доступ к ключам и доступ к периферийным устройствам, подсоединенным к параллельному порту, может вызвать зависание.

Поэтому важно избегать пересечения адресов портов ввода/вывода. Для этого можно изменить базовый адрес порта ввода/вывода сетевой платы.

Существуют два способа изменения адреса:

- Некоторые сетевые карты позволяют вам присваивать адреса портов ввода/вывода джамперами.

  Для получения более полной информации по этой возможности, обратитесь к документации вашей сетевой платы.

- Многие новые сетевые платы позволяют изменять адреса с помощью программного обеспечения, поставляемого с этими платами.

Проблема: Ваше приложение, защищенное с помощью HASP4 Net для Windows, работает в ОС Windows for Workgroups в сети Novell (протокол IPX) и возвращает код ошибки 3.

Решение: Типы фреймов, описанные в файле net.cfg и установки сети Windows, не идентичны друг другу. Проверьте тип фреймов в файле net.cfg и установите такой же тип фрейма в Windows.

Чтобы установить типы фрейма в Windows:

- Выберите **Network Setup** в Control panel.
- Щелкните по пиктограмме **IPX/SPX Compatible Transport with NetBIOS**.
- Выберите **Frame Type**.
- Выберите тип фрейма в списке **Value**, затем нажмите **Set**.
- Нажмите **OK**.
- Перезагрузите компьютер.

Проблема: Ваше приложение очень долго ищет ключ HASP4 Net в большой сети Novell.

Решение: В этом случае мы рекомендуем модифицировать механизм поиска. Используйте файл конфигурации HASP4 Net, чтобы отключить механизмы поиска broadcast и bindery. После того, как вы проделаете это, клиент HASP4 Net ищет HASP LM, используя механизм адресных файлов, что значительно быстрее.

Проблема: Ваше приложение очень долго ищет ключ HASP4 Net в большой сети TCP/IP.

Решение: В этом случае мы рекомендуем модифицировать механизм поиска. Используйте файл конфигурации HASP4 Net, чтобы указать метод поиска UDP или TCP и указать IP-адрес HASP LM. После того, как вы проделаете это, клиент HASP4 Net ищет HASP LM по конкретному адресу, что значительно быстрее.

Проблема: При использовании HASP4 Net вы получаете ошибку 8.

Решение: Ошибка 8 означает, что запрос достиг HASP LM, но клиент HASP4 Net не получил ответного сообщения. Чтобы решить эту проблему, попытайтесь увеличить время, в течение которого клиент HASP4 Net ожидает ответа. Чтобы сделать это, внесите соответствующие изменения в файл конфигурации HASP4 Net.

Проблема: Вы получаете ошибку 15 при использовании HASP4 Net под TCP/IP или IPX?

Решение: Ошибка 15 под TCP/IP/IPX происходит только в том случае, если вы используете механизм поиска broadcast. Эта ошибка означает, что сообщение было ретранслировано клиентом HASP4 Net, но HASP LM не был найден.

Увеличьте время ожидания в файле *nethasp.ini* до 8 секунд. Если это не помогает, причиной этой ошибки может быть:

- HASP LM не загружен.
- В случае использования протокола TCP/IP, HASP LM находится в другой подсети.
- В случае использования протокола IPX, SAP не поддерживается.
- Если вы получаете ошибку 15 многократно, попробуйте использовать другую процедуру поиска.

Проблема: Вы используете HASP4 Net5 и разрешили 5 подключений, но приложение может быть использовано одновременно только тремя пользователями.

Решение: Вначале с помощью HaspEdit убедитесь, что ваше приложение запрограммировано на пять лицензий HASP4 Net. Если это так, возможно, что вы не заметили того, что уже работают все пять лицензий. Используйте Aladdin Monitor, чтобы убедиться, какие станции в действительности используют лицензии.

Приложение B

Пароли ключей HASP Demo

При заказе ключей HASP им присваивается уникальный код разработчика и пароль. Пароли заказчик ключей получает вместе с ключами HASP.

В приведенных ниже таблицах приведен список паролей, которые присваиваются демонстрационным ключам HASP, входящим в Комплект разработчика HASP.

Таблица B.1. Пароли ключей HASP Demo Memory

Код разработчика	Первый пароль	Второй пароль
DEMOMA	15417	9632
DEMOMB	29875	28774
DEMOMC	29313	23912

Таблица B.2. Пароли ключей HASP4 Std Demo

Код разработчика	Первый пароль	Второй пароль
DEMO3A	5932	25657
DEMO3B	20580	22012
DEMO3C	10038	15697

Приложение C

Технические спецификации

Таблица C.1. Общие спецификации для всех ключей HASP

Материал корпуса	ABS
Рабочая температура	0°-55°C (32°-131°F)
Температура хранения	-25°-70°C (-13°-158°F)
Влажность	0-100% (без конденсации)
Стандарт UL-6C61 1950	94-V0
Рабочее напряжение чипа ASIC	1.8-5.5V
Гирляндное соединение (за исключением ключей HASP4 USB)	до 10 ключей
Число циклов перезаписи одной ячейки памяти	не менее 1000000
Срок хранения информации в памяти	не менее 10 лет

Таблица C.2. Спецификации HASP4 Std, HASP4 M1, HASP4 M4, HASP4 Net

Размеры	39x53x17мм

Вес	~33гр
Разъем	DB25
Используемые линии	D0-D7, INIT, ATFDXT, PE
Размер памяти на чтение/запись:	
HASP4 без памяти	нет
HASP4 M1	112 байт
HASP4 M4	496 байт
HASP4 Net	496 байт
Батарейки/внешние источники питания	нет

Таблица C.3. Спецификации HASP4 Time

Размеры	52x53x16мм
Вес	~50гр
Разъем	DB25
Используемые линии	D0-D7, INIT, ATFDXT, PE
Размер памяти на чтение/запись	512 байт
Показания часов	час, минута, секунда, год, месяц, день
Точность часов	погрешность - 2 часа в год
Срок службы батарейки	более четырех лет

Таблица C.4. Спецификации моделей для порта USB

Размеры	57x16x8мм
Вес	~7гр
Разъем	USB Type A
Используемые линии	Питание, земля, две линии данных
Размер памяти на чтение/запись:	
HASP4 USB	нет
HASP4 USB M1	112 байт
HASP4 USB M4	496 байт
HASP4 USB Net	496 байт
Батарейки/внешние источники питания	нет

Таблица C.5. HASP PC-Card

Размеры	Type II
Вес	~25гр
Рабочая температура	0°-70°C (32°-158°F)
Влажность	Относительная влажность 20-80%
Энергопотребление	<100 mA (обычно 50 mA)
Рабочее напряжение	5V
Батарейки/внешние источники питания	нет
Технология ASIC	CMOS 2μФ с ячейками E2
Число циклов программирования	>100000
Срок хранения информации в памяти	более 10 лет

Таблица C.6. AladdinCARD ISA

Размеры	113x100мм
Вес	~105гр (включая кабели)
Разъем	8-битный слот шины ISA внешний DB25 (розеточный)
Адреса ввода/вывода	278h, 278h, 3BCh
IRQ	нет, IRQ5, IRQ7
Рабочее напряжение	4.5V .. 5.5V

Таблица C.7. AladdinCARD PCI

Размеры	180x124мм
Вес	~105гр (включая кабели)
Разъем	разъем PCI 32-bit/33MHz/5V внутренний DB25 (розеточный)
Адреса ввода/вывода	назначаются динамически
IRQ	назначаются динамически
Рабочее напряжение	5V

Глоссарий

Автоматический поиск HASP4 Net	Метод, с помощью которого защищаемое приложение осуществляет поиск HASP LM.
Активации	Разрешенное число запусков защищаемого приложения.
Время простоя	Период времени, по истечению которого станция считается неактивной.
Дата окончания срока	Дата, после которой защищаемое приложение больше не будет работать.
Динамическая загрузка драйвера устройства HASP	Свойство драйвера устройства HASP, позволяющее драйверу загружаться без перезагрузки системы.
Драйвер устройства HASP	Драйвер, выступающий в роли интерфейса между ключом HASP и защищаемым приложением.
Каскадное соединение	Последовательное соединение ключей HASP один за другим.
Клиент HASP4 Net	Станция, осуществляющая запуск защищаемого приложения.
Код разработчика	Уникальный код, присваиваемый каждому разработчику программных продуктов, хранящийся на ключах HASP.
Комплект для начала работы с HASP	Набор средств, необходимых для оценки и немедленного обеспечения защиты программного обеспечения.

Комплект разработчика HASP	Набор средств, позволяющих оценить систему защиты HASP.
Локальный ключ HASP	Ключ HASP, предназначенный для использования на отдельно стоящем компьютере.
Механизм поиска с помощью адресного файла	Механизм поиска, предусматривающий доступ клиента HASP4 Net к файлу с целью получения адреса HASP Net License Manager.
Модели HASP4 Net	Серии ключей HASP4 Net, которые различаются максимальным числом лицензий.
Память HASP	Внутренняя память HASP, доступная на чтение и запись, размер которой зависит от модели ключа.
Пароли HASP	Два уникальных пароля, присваиваемых каждому разработчику программного обеспечения для получения доступа к HASP.
Процедура hasp()	Процедура HASP API, которая обеспечивает доступ к HASP.
Система полного управления доступом (FAS)	Функция, позволяющая защищать несколько приложений с помощью одного ключа, а также определять условия использования каждого приложения.
Регистрационная таблица HASP4 Net	Список защищаемых приложений и станций, на которых они запущены, подключившихся к ключу HASP4 Net.
Тестовые утилиты HASP	Утилиты, предназначенные для тестирования операций HASP API и HASP4 Net.
Файл конфигурации HASP4 Net	Файлы, содержащие настройки клиента HASP4 Net и HASP LM.
Фоновые проверки HASP	Проверки наличия ключа HASP, осуществляемые HASP Envelope во время работы защищаемого приложения.
Число активаций	Определенное заранее число возможных запусков защищаемого приложения.
Число лицензий	Число станций, на которых можно одновременно использовать защищаемое приложение.

Широковещательный механизм поиска	Механизм поиска, в рамках которого клиент HASP4 Net использует широковещательную передачу SAP для поиска HASP Net License Manager.
Aladdin Diagnostic	Инструмент, позволяющий получать информацию о системе и HASP.
ASIC	Application Specific Integrated Circuit, интегральная схема специального назначения для ключей HASP; заказная схема, реализующая уникальный алгоритм.
DataHASP	Функция утилиты HASP Envelope, которая используется для защиты файлов данных.
DemoMA	Код разработчика демонстрационных ключей HASP.
HASP	Аппаратная система защиты программного обеспечения
HASP API	Программный интерфейс ключей HASP для работы с объектными файлами и DLL, позволяющий вставлять вызовы HASP в программный код защищаемого приложения.
HASP Demo	Демонстрационный ключ HASP, который может использоваться для оценки системы защиты HASP. Пароли – 15147, 9632, код разработчика – DEMOMA.
HASP Edit	Утилита, используемая для получения доступа к HASP и редактирования памяти ключей HASP.
HASP Envelope	Утилита, помещающая программу в защитную оболочку; следит за тем, чтобы программа не работала без наличия соответствующего ключа HASP.
HASP Envelope Wizard	Пошаговое руководство по обеспечению защиты с использованием HASP Envelope.
HASP4 Net Login	Процесс, с помощью которого защищаемое приложение запрашивает разрешение на запуск у HASP LM.
HASP4 Net Logout	Процесс, с помощью которого защищаемое приложение информирует HASP LM о том, что оно больше не использует лицензию.
HASP Memory	Ключ HASP с памятью, которая доступна на чтение и запись. Размер памяти зависит от модели ключа. Память содержат следующие модели HASP: HASP M1 (112 байт), HASP4 M4 (496 байт), HASP4 Time (496+16 байт), HASP4 Net (496 байт).

HASP4 Net Monitor	Инструмент, используемый для отслеживания использования защищаемых приложений в сети.
HASP Net License Manager	Программа, выступающая в роли интерфейса между защищаемым приложением и ключом HASP4 Net.
HASP4 Time	Локальный ключ с чипом ASIC, встроенными часами и, в зависимости от модели, внутренней памятью размером до 512 байт.
Hinstall	Приложение, которое устанавливает драйвер устройства HASP.
ID–номер HASP	Уникальный номер, присваиваемый ключу HASP с памятью в процессе производства.
nethasp.ini	Файл конфигурации HASP4 Net для защищаемого приложения. В нем содержатся настройки, определяющие поведение HASP4 Net в сети.
nhsrv.ini	Файл конфигурации для HASP Net LM. В нем содержатся настройки HASP LM.
PC–CardHASP	Современная карта для защиты программного обеспечения, которая вставляется в слот PCMCIA ноутбука.

Индекс

В

Д

З

К

Л

М

Н

О

П

Р

С

Т

У

Ш

Aladdin

SECURING THE GLOBAL VILLAGE

For further info: www.eAladdin.com/HASP

International	T: +972-3-636-2222 F: +972-3-537-5796, HASP@eAladdin.com
North America	T: 1-800-562-2543, 1-847-818-3800, F: 1-847-818-3810, HASP.us@eAladdin.com
UK	T: +44-1753-622266 F: +44-1753-622262, HASP.uk@eAladdin.com
Germany	T: +49-89-89-42-21-0 F: +49-89-89-42-21-40, HASP.de@eAladdin.com
Benelux	T: +31-30-688-0800 F: +31-30-688-0700, HASP.nl@eAladdin.com
France	T: +33-1-41-37-70-30 F: +33-1-41-37-70-39, HASP.fr@eAladdin.com
Israel	T: +972-3-636-2222 F: +972-3-537-5796, HASP.il@eAladdin.com
Brazil	T: +55-21-235-2499 F: +55-21-236-0768, HASP.br@eAladdin.com
Japan	T: +81-426-60-7191 F: +81-426-60-7194, HASP.jp@eAladdin.com
Russia	T: +7-095-231-3113 F: +7-095-928-6781, HASP@Aladdin.ru
Spain	T: +34-91-375-99-00 F: +34-91-754-26-71, HASP.es@eAladdin.com
China	T: +86-10-6526-9920 F: +86-10-6526-9921, HASP.cn@eAladdin.com