Guardant®

Система защиты от компьютерного пиратства

Эффективная
защита приложений

Урок 1.2
Знакомство с утилитой программирования электронных ключей

Содержание

Общее описание утилиты программирования

Полноценная организация процессов лицензирования приложений возможна только с использованием утилиты программирования ключей Guardant – **GrdUtil.exe**.

Необходимость в программировании ключа самим разработчиком защищаемого приложения возникает при усложнении лицензионной политики (к примеру, когда нужно по отдельности проводить процесс лицензирования различных функциональных модулей защищаемого приложения, или в случае записи нескольких независимых лицензий в один электронный ключ).

Используемые термины и обозначения

Разработчик – создатель программного продукта; программист, использующий электронные ключи Guardant для защиты и лицензирования своего продукта.

Конечный пользователь – клиент разработчика, покупатель программного продукта, защищенного ключами Guardant.

Маска ключа (образ памяти ключа) – образ памяти ключа, сохраненный в базе данных или файле формата .nsd. Совокупность полей памяти, их структуры и значений, представленная в удобной для восприятия форме.

Аппаратный алгоритм – алгоритм шифрования, выполняющийся в электронном ключе. Неизменяемая часть алгоритма, собственно реализующая функцию шифрования, содержится в микропрограмме ключа. Доступная для изменения часть алгоритма (дескриптор) содержится в EEPROM. Дескриптор состоит из набора флагов и определителя (секретного ключа), которые служат для формирования конкретного вида алгоритма и его свойств.[*] Дескриптор алгоритма является разновидностью защищенной ячейки.

Защищенная ячейка – тип поля, защищенного аппаратными запретами на чтение/запись и содержащего набор структурированных для той или иной цели данных. К разновидностям защищенных ячеек относятся дескрипторы аппаратных алгоритмов, таблица лицензий (для сетевых ключей), собственно защищенные ячейки (для хранения произвольных данных) и загружаемый код (только для ключей Guardant Code).

К достоинствам ячеек относятся: их защищенность от снятия дампа, наличие сервисов активации/деактивации, доступ к содержимому по паролю, упрощенная адресация по номеру ячейки.

Определитель алгоритма – секретный ключ алгоритма.

Возможности утилиты программирования ключей GrdUtil

Утилита предоставляет широкий спектр возможностей для редактирования памяти ключа и работы с лицензионной информацией:

- Работа с маской (внешним образом памяти) ключа
- Работа с электронными ключами:
- Работа со встроенной базой данных:
- Подготовка данных для защиты приложений

Утилита программирования ключей **GrdUtil** имеет удобный графический интерфейс, однако в случае необходимости автоматизации тиражирования лицензий существует возможность ее использования через **интерфейс командной строки**.

[*] В большинстве ситуаций, описанных в документации, под аппаратным алгоритмом подразумевается его дескриптор. Это сделано для простоты понимания, т. к. дескриптор – единственная часть аппаратного алгоритма, которой может управлять разработчик.

Программирование ключей

Лицензия, записываемая в ключ при защите приложения (речь о которой шла ранее) представляет собой, по сути, совокупность данных маски ключа. Под **программированием ключа** понимается запись сформированной маски в ключ, содержащей информацию о лицензионных ограничениях использования защищенного приложения.

Реализация лицензионных ограничений происходит путем задания соответствующих параметров отдельных алгоритмов в процессе формировании маски ключа (таких как, к примеру, активация/деактивация алгоритма по времени или ограничение общего числа обращений к алгоритму/защищенной ячейке).

Поля памяти ключа

Для хранения маски в ключах Guardant имеется 4 КБ энергонезависимой памяти (более подробно об устройстве и принципах работы ключа речь пойдет в одном из следующих уроков).

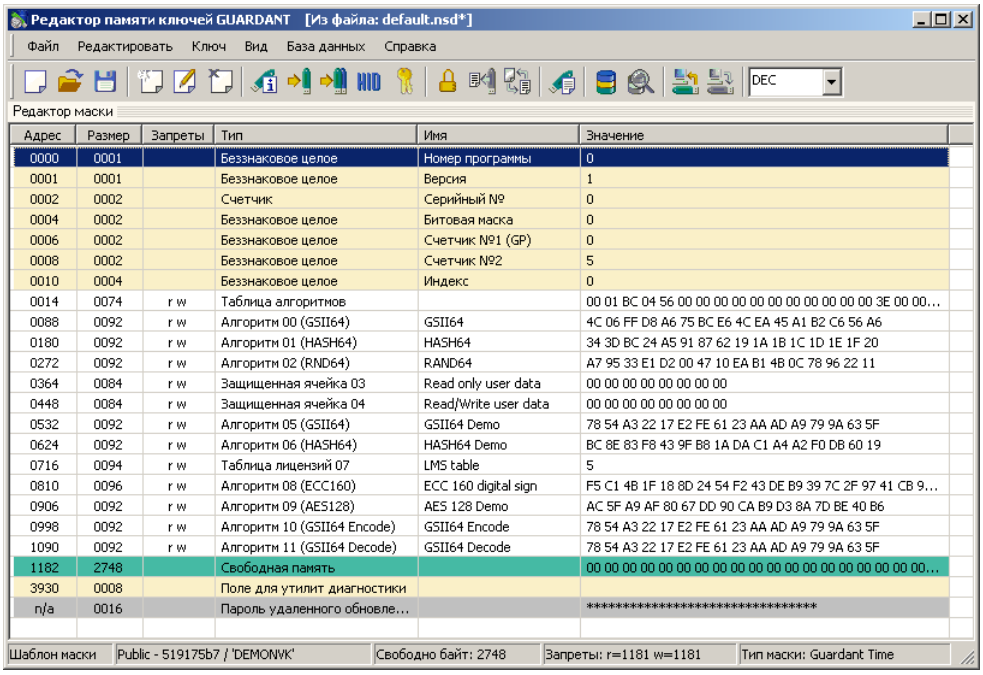
Всю память ключа можно функционально разделить на две последовательно расположенные в памяти области:

- Поля общего назначения
- Поля свободного назначения

Поля общего назначения имеют строго жесткую структуру и используются утилитами автозащиты и функциями API для хранения информации о самом электронном ключе. Они доступны для чтения/записи вне зависимости от установленных запретов чтения/записи памяти ключа.

Поля свободного назначения позволяют хранить любую структуру данных и могут быть защищены от чтения/записи посредством установки соответствующих запретов.

Ниже приведен снимок окна **редактора памяти GrdUtil**:



Маска ключа

Маска ключа представляет совокупность структур данных, описывающих содержимое памяти ключа, записываемых в область памяти свободного назначения.

К данным маски также можно отнести содержимое некоторых полей памяти общего назначения, задаваемых в процессе записи маски в ключ (среди них информация о запретах чтения/записи памяти ключа, а также значения некоторых флагов).

Маска может храниться в **базе данных GrdUtil** или в специальном файле (**файле маски** с расширением *.nsd).

Маски различных поколений ключей различных типов имеют определенные отличия. Важно следить, чтобы в ключ записывалась подходящая по типу маска.

Существует возможность добавления в структуру следующих типов полей:

- Аппаратный алгоритм
- Защищенная ячейка
- Таблица сетевых лицензий
- Целое число
- Строка
- Счетчик
- Дамп памяти

Под **аппаратным алгоритмом** понимается структура данных маски, содержащая информацию об использовании того или иного криптографического алгоритма, поддерживаемого ключом, включая определитель алгоритма.

При обращении к полю такого типа преобразование данных происходит внутри ключа, секретный ключ (**определитель алгоритма**) при этом не покидает электронного ключа.

Аппаратные алгоритмы предназначены для выполнения криптографических преобразований информации, используемых в процессе защиты и исполнения защищенного приложения. Они реализуются микропрограммой ключа, записанной в микроконтроллер. Микропрограмма Guardant защищена от считывания и модификации.

Забегая вперед, необходимо отметить, что наиболее эффективная защита может быть реализована с использованием основных типов – аппаратных алгоритмов и защищенных ячеек

Редактор маски

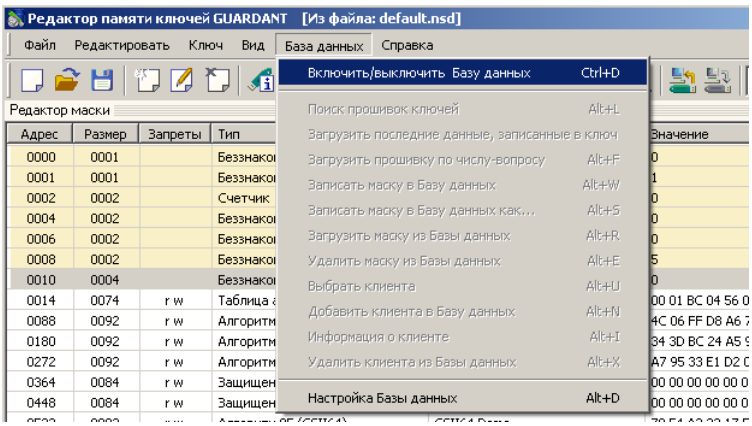
Редактор маски расположен в главном окне GrdUtil и предоставляет основную часть функциональности утилиты программирования ключей. Обобщенный порядок работы с редактором маски выглядит следующим образом:

- Создание маски или загрузка из базы данных / файла маски
- Редактирование структуры и содержимого маски
- Выполнение локального или удаленного обновления памяти ключа
- Сохранение текущей маски в базе данных / файле маски

Программирование ключей

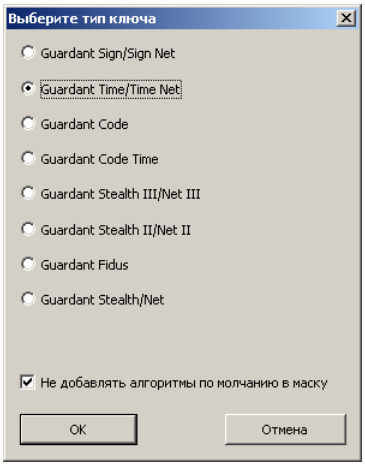
Шаг 0. Подготовка к работе

При первом запуске **GrdUtil** будет предложено включить базу данных и выполнить ее экспресс-настройку. С целью упрощения процесса изучения работы с утилитой программирования рекомендуется отклонить предложение (или отключить базу данных, если она была включена до этого). Порядок работы с ней будет подробно рассмотрен позже.



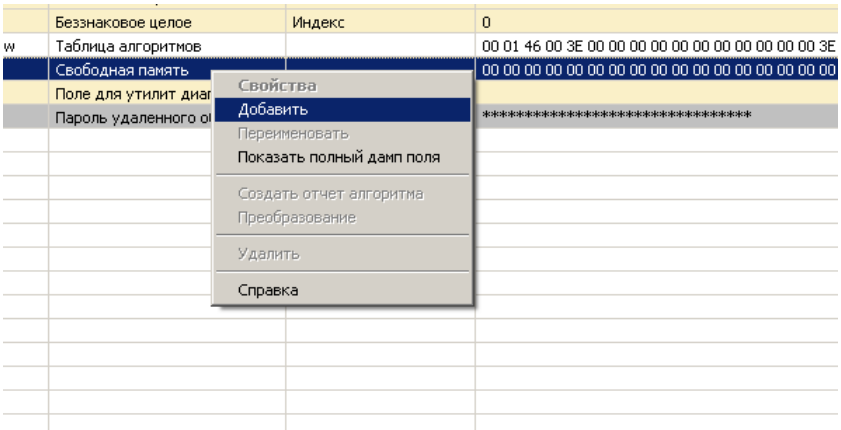
Шаг 1. Создание пустой маски

Маска содержит информацию о лицензиях защищаемого приложения. Для каждого приложения она будет содержать свой набор алгоритмов. Поэтому начинать процесс программирования ключа рекомендуется с пустой маски. Создадим маску без алгоритмов для интересующего нас типа ключа (предположим, что это **Guardant Time**).

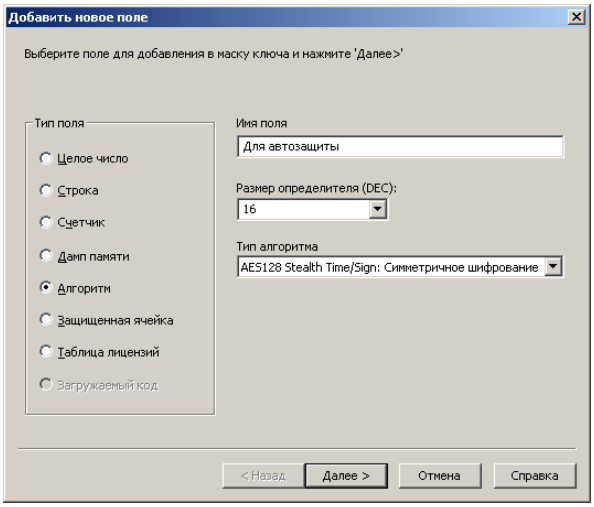


Шаг 2. Добавление элементов в маску

Приступим к формированию маски. Для корректной работы автозащиты необходимо наличие хотя бы одного симметричного алгоритма в маске. Добавим его в свободную область памяти.



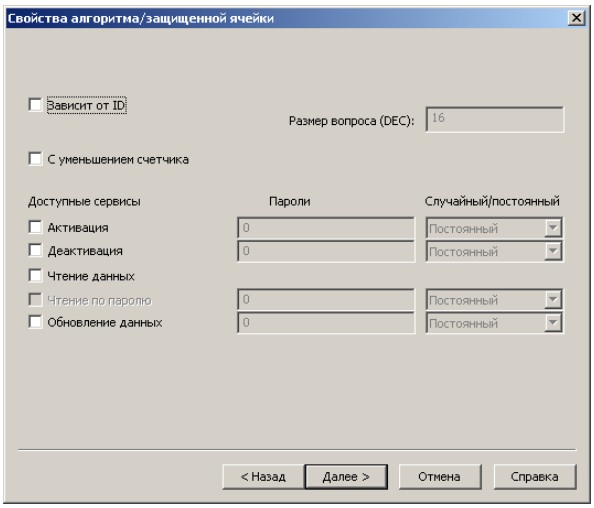
Из симметричных криптографических алгоритмов доступны два: GSII64 и AES128. Выберем AES128. Он поддерживается всеми современными ключами Guardant. Размер определителя для указанного алгоритма фиксирован и равен 16 байт (128 бит).



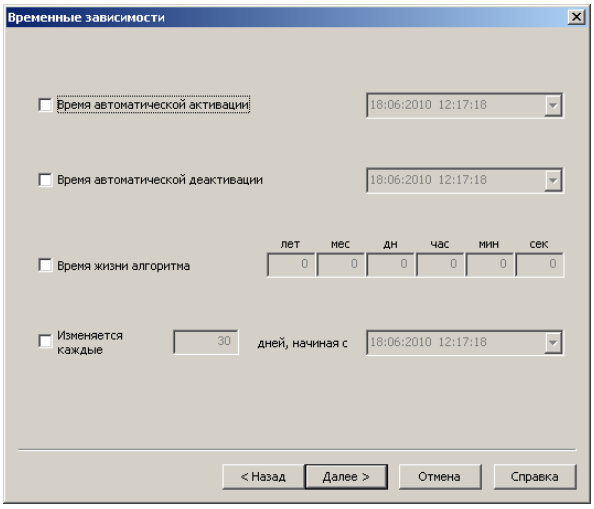
В следующем окне диалога необходимо установить ряд параметров создаваемого алгоритма.

Из них следует отметить возможность установки **зависимости от уникального идентификатора** электронного ключа (ID). Такая зависимость позволяет впоследствии привязывать защищаемое приложение к одному единственному ключу так, как будто в нем записан уникальный секретный ключ симметричного алгоритма шифрования, без необходимости хранить различные ключи для каждой копии защищенного приложения. Это может быть использовано при необходимости персонализации каждой копии защищаемого приложения.

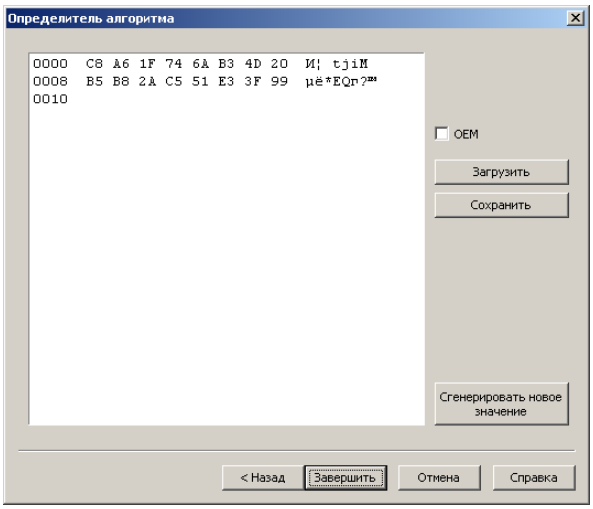
Сервисы работы с определителем позволяют производить удаленное блокирование/разблокирование алгоритма, а также чтение и обновление его определителя. Для алгоритма автозащиты их рекомендуется не устанавливать.



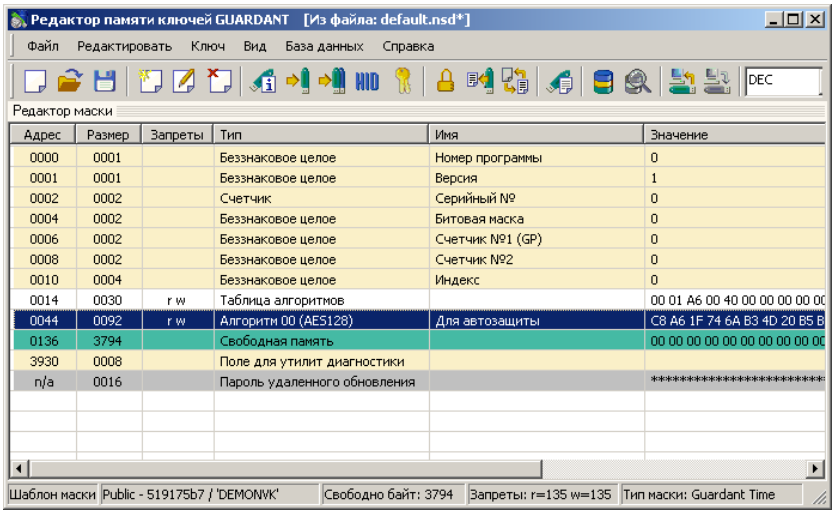
Диалог установки параметров временных зависимостей алгоритма будет доступен только при работе маской ключей с часами реального времени **Guardant Time / Time Net / Code Time**.



Заключительный этап добавления аппаратного алгоритма в маску ключа представляет задание его определителя. По умолчанию он содержит псевдослучайную последовательность с высокими вероятностными характеристиками, которую можно перегенерировать, задать вручную или загрузить из бинарного файла.



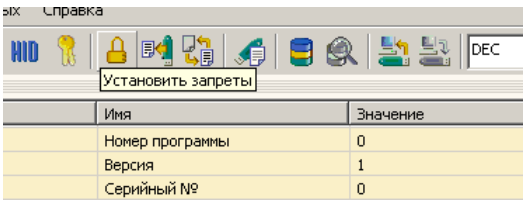
После окончания работы Мастера в структуру маски будет добавлен созданный алгоритм:



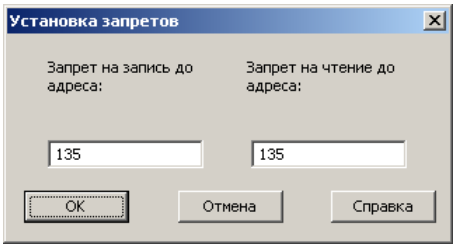
Аналогичным образом может быть создана **защищенная ячейка** и любое другое поле. Отличие состоит лишь в том, что временные ограничения могут быть заданы только для аппаратных алгоритмов.

Шаг 3. Установка параметров записи маски и режима работы ключа

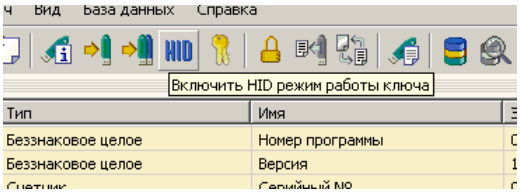
На данном этапе необходимо определить, какие поля маски будут доступны для чтения/записи. Для установки **запретов на доступ к памяти** выберем соответствующий элемент панели инструментов.



По умолчанию запретами накрываются все алгоритмы и защищенные ячейки маски. Однако вручную можно сдвинуть **границы запретов** чтения/записи как в меньшую, так и в большую сторону.



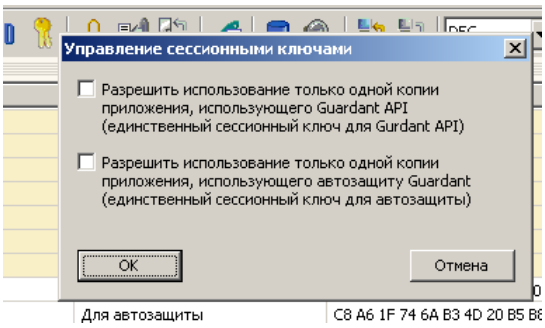
Также при необходимости, на данном этапе можно установить **HID-режим** работы ключа (см. рисунок). В этом случае во время следующего подключения ключа к порту ключ будет работать без использования драйвера Guardant.



Примечание

Работа электронных ключей с использованием **драйвера Guardant** является более стабильной и защищенной от внешних воздействий и анализа.

Выбрав элемент панели инструментов **Управление сессионными ключами** можно ограничить число одновременно открытых сессий работы с ключом, установив, таким образом, защиту от использования защищенного приложения в терминальном режиме.



Шаг 4. Запись маски в ключ

Теперь можно записать маску в ключ. По мере усовершенствования защиты приложения в нее могут добавляться новые алгоритмы и ячейки памяти. При этом, данные могут быть записаны в ключ пользователя удаленно, как полное либо частичное обновление памяти ключа. Речь об этом пойдет в одном из следующих уроков.

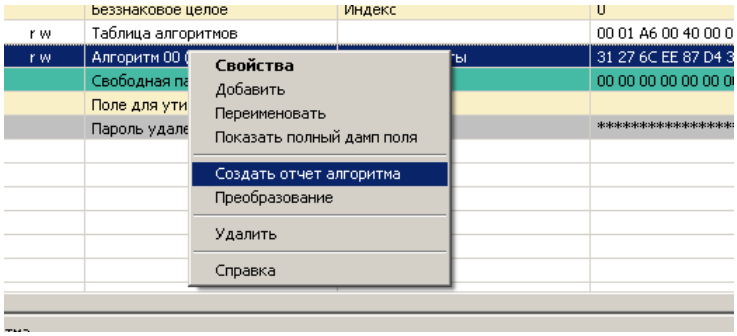
Работа с аппаратными алгоритмами

После того, как маска записана в ключ, существует возможность протестировать записанные алгоритмы одним из следующих способов:

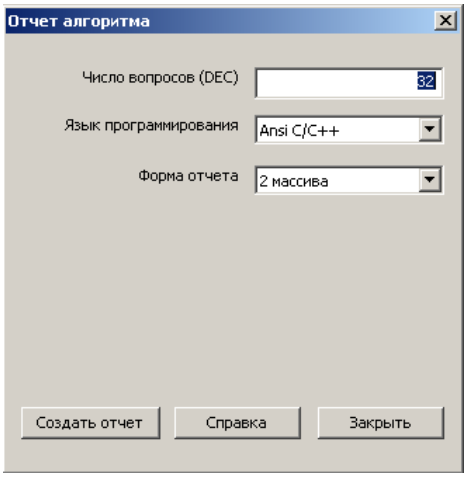
- Создать таблицу вопросов-ответов при помощи аппаратного алгоритма
- Преобразовать данные при помощи аппаратного алгоритма
- Выполнить обращение к алгоритму/защищенной ячейке из Guardant API при помощи графической утилиты вызова функций Guardant API

Создание отчета аппаратного алгоритма

Под генерацией отчета аппаратного алгоритма понимается создание таблицы вопросов-ответов заданного алгоритма на необходимом языке программирования. В качестве входных данных при этом используются случайные последовательности.



Из доступных параметров можно задать общее количество пар вопросов-ответов и формат отчета: язык программирования и способ взаимного расположения вопросов и ответов в отчете.



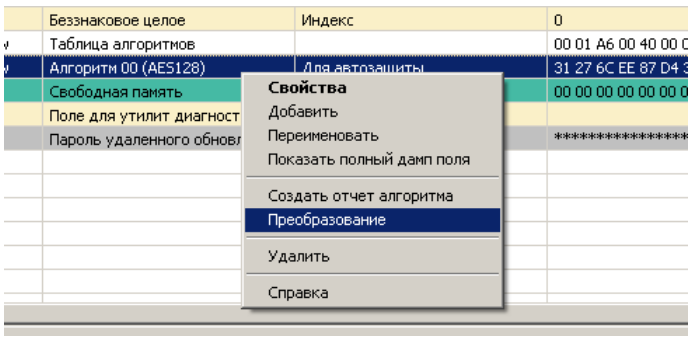
Отчет по алгоритму AES128, записанному в маске ключа Guardant Time под номером «0», для языка C/C++ будет выглядеть примерно следующим образом:

```
/* Guardant Time, Operation - Transform */
/* 32  questions and answers, Algorithm No 0 */
/* Question/answer size 16  bytes            */

#define ns_Size 32
#define ns_Blen 16

unsigned char ns_Question[ns_Size*ns_Blen] = {
 0x33, 0x09, 0xea, 0xb1, 0x63, 0xa4, 0x71, 0x9d, 0xC
 0x68, 0x12, 0x9b, 0x46, 0xbe, 0x2a, 0x99, 0xec, 0xe
 0x1f, 0xa6, 0x4b, 0x17, 0x32, 0x38, 0x40, 0x4e, 0x2
 0x9c, 0x64, 0x71, 0x1b, 0xf7, 0x52, 0x50, 0x7d, 0x5
 0x86, 0xa5, 0xc1, 0xb4, 0xda, 0x5b, 0x14, 0x7d, 0x5
 0x83  0x2c  0xe6  0x4b  0x3b  0x87  0xfc  0x50  0xb
```

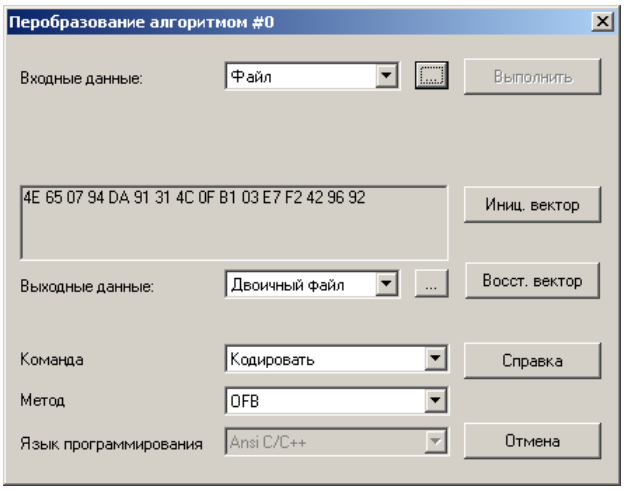
Полученные таким образом данные могут быть использованы в процессе защиты.

Преобразование данных при помощи аппаратного алгоритма

Преобразование данных отличается от процедуры создания отчета тем, что на вход аппаратному алгоритму подается заранее определенная последовательность байт (заданная в виде строки ASCII символов или файла данных).



В случае преобразования данных необходимо вручную выбрать режим шифрования (**метод**), а также способ представления выходных данных.



В результате преобразования, помимо файла вывода, будет предложено сохранить файл с информацией о производимом преобразовании с именем «**report.rep**».

Выполнение функций Guardant API

Обращение к алгоритму можно также выполнить при помощи функций Guardant API, в частности, при их вызове из утилиты GrdUtil (пункт **Вызов библиотечных функций** панели инструментов). Подробнее порядок работы с этим инструментом будет рассмотрен в одном из уроков по работе с **Guardant API**.

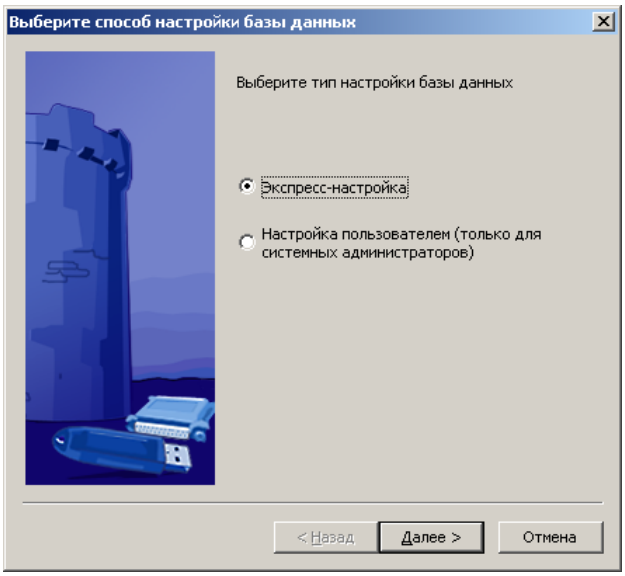
Работа с базой данных

Одной из ключевых возможностей утилиты GrdUtil, является возможность ведения базы данных масок записанных в ключ, а также сопоставления информации о фактах записи маски и конечных пользователях защищенного приложения. Это даёт возможность вести учет лицензий передаваемых пользователям, модифицировать их при необходимости и даже удаленно проводить частичное или полное обновление памяти ключа (речь о котором пойдет в одном из следующих уроков). Последняя возможность особенно актуальна, так как позволяет при добавлении новых функций защиты, сохранять алгоритмы, необходимые для корректной работы предыдущей версии.

В начале данного урока база данных была временно отключена. Соответственно, информация обо всех фактах прошивки ключей за этот период не сохранялась. Настало время ее включить.

Настройка базы данных

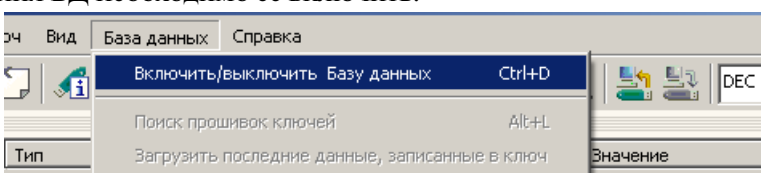
В неактивном состоянии может быть выполнена настройка базы данных.



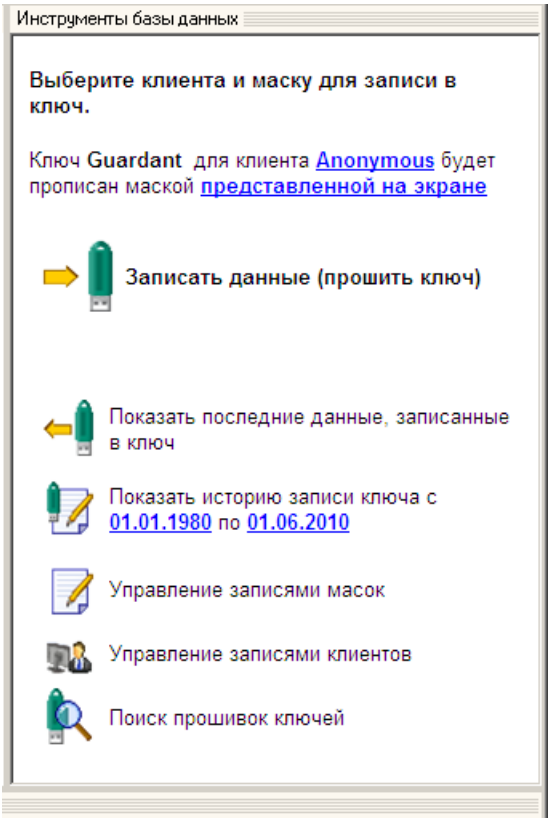
В директории Комплекта разработчика расположены два файла баз данных: **Sample.mdb** (содержащий пример ведения БД) и **Grdutil.mdb** (используемый для хранения рабочей БД по умолчанию). Выберем второй.

Включение базы данных

После выбора файла расположения БД необходимо ее включить.

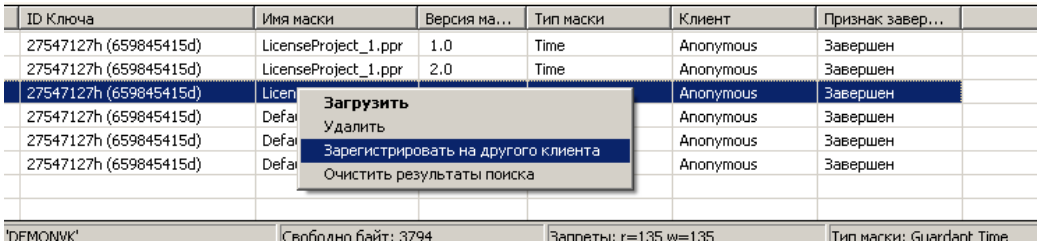


В открывшемся окне предлагаются следующая функциональность по работе с базой данных:



Работа с базой данных

Перед началом записи масок в ключи для определенного клиента (группы клиентов) необходимо создать его в диалоге **Управление записями клиентов** и/или **Выбрать клиента** для сохранения информации о следующих фактах записи маски на его имя. В случае необходимости существует возможность **Зарегистрировать на другого клиента** прошивку ключа, сохраненную в БД:



Необходимо отметить, что маска ключа, записываемая в процессе автозащиты приложения (при выполнении автозащиты с записью лицензии в ключ), по умолчанию регистрируется в БД на пользователя с именем **Anonymous** вне зависимости от текущего выбранного клиента.

Остальной функционал GrdUtil работы с Базой данных должен быть интуитивно понятен. В случае необходимости, можно обратиться к первому тому документации Guardant.

Заключение

В процессе выполнения данного урока был рассмотрен обобщенный порядок действий, необходимый для самостоятельного формирования и записи в ключ лицензии (маски с лицензионными ограничениями). Структура маски формируется в процессе проектирования защиты. Защита приложения происходит на основе созданной маски и, соответственно, после записи ее в ключ.

Возможность ведения базы данных записанных в ключи масок призвана упростить итеративный процесс разработки, реализации и сопровождения защиты приложения.

Рекомендуется не отключать базу данных в процессе работы с ключами. Хранимая в ней информация способна значительно облегчить процедуры учета записанных и переданных конечным пользователям лицензий, в особенности, когда лицензионная политика подразумевает помещение уникальных данных в ключ при каждой итерации записи маски.

Дополнительные источники информации

При возникновении вопросов, на которые вам не удалось найти ответа в этом пособии, рекомендуем обратиться к следующим дополнительным источникам информации:

WWW: http://www.guardant.ru

Web-сайт разработчика содержит большой объем справочной информации об электронных ключах Guardant.

Служба технической поддержки:

e-mail: hotline@guardant.ru

тел. +7(495)925-77-90