Guardant®

Система защиты от компьютерного пиратства

Эффективная
защита приложений

Урок 1.3: автозащита
с самостоятельным программированием ключа

Содержание

Общее описание процесса защиты

В предыдущих уроках был рассмотрен порядок работы с **Мастером автозащиты** в варианте создания и записи лицензии в ключ самим Мастером (урок 1.1), а также затронуты основы работы с **утилитой программирования** ключей (урок 1.2).

Очередным шагом на пути к созданию защиты приложения является использование Мастера автозащиты в варианте, когда этапы «навешивания» защиты и программирования ключа разделены.

В рамках данного урока будет рассмотрен порядок работы с Мастером автозащиты, который используется при создании **схемы защиты** приложения, совмещающей автоматическую защиту и защиту при помощи **Guardant API**.

Этапы построения защиты

Работа по подготовке защищаемого приложения и электронных ключей к передаче конечным пользователям может быть разделена на несколько этапов.

Первый этап – **проектирование защиты** – включает в себя формирование маски ключа в соответствии с требованиями к возможностям установки лицензионных ограничений. Это итерационный процесс, в ходе которого может происходить разработка и реализация отдельных механизмов лицензирования приложения, а также выбор алгоритмов приложения, подлежащих защите.

На втором этапе происходит запись в ключ спроектированной маски и реализация защиты приложения при помощи Guardant API, а также автоматическая защита. Его завершением является проведение контрольных испытаний защищенного приложения.

Заключительным этапом является **тиражирование лицензий** с записью соответствующих лицензионных ограничений в ключи и передача их конечным пользователям.

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, сохраненный в базе данных или файле формата .nsd. Совокупность полей памяти, их структуры и значений, представленная в удобной для восприятия форме.

Создание маски ключа

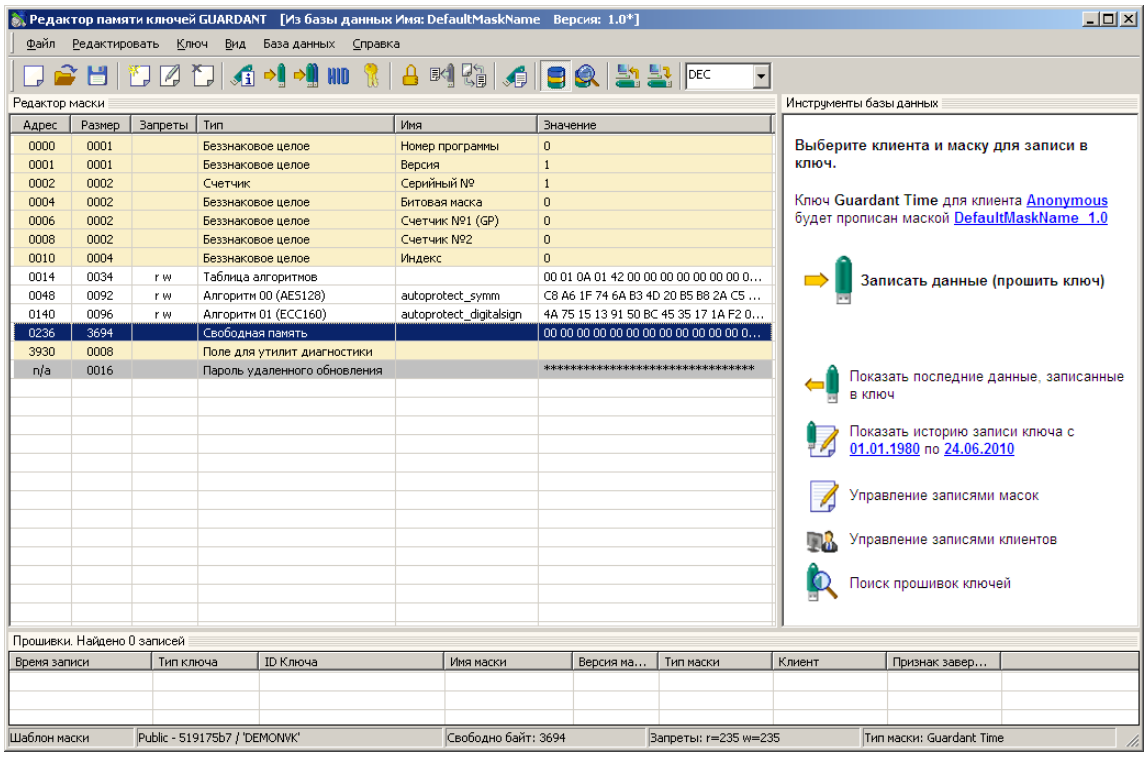
Маска ключа создается на этапе проектирования защиты. Лицензионные ограничения на использование отдельных функциональных модулей приложения реализуются при помощи установки ограничений на обращение к элементам маски ключа (аппаратным алгоритмам).

Приступим к созданию маски. В нее необходимо добавить набор алгоритмов достаточный для полноценной организации лицензионной политики.

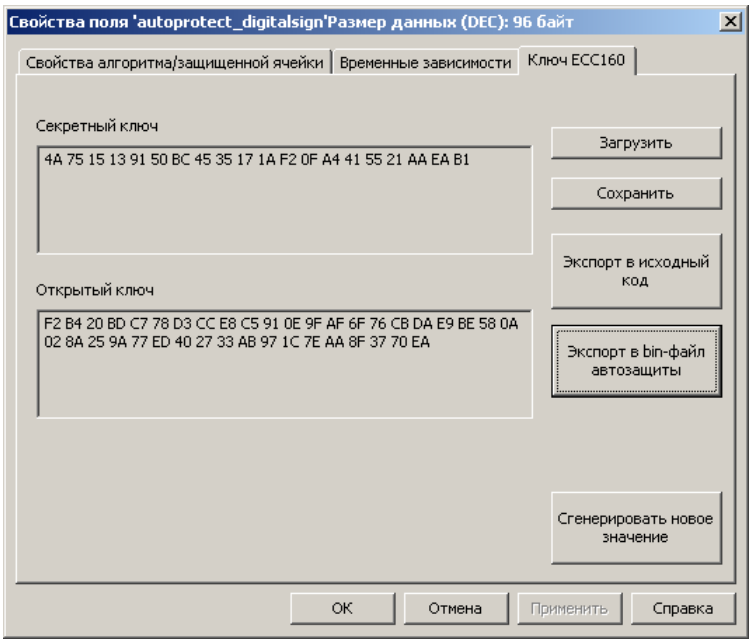
Добавление симметричного алгоритма автозащиты

Для успешного наложения автозащиты в маске ключа, к которому привязывается приложение, должен содержаться любой симметричный алгоритм (GSII64 или AES128). Кроме того, некоторые опции автозащиты требуют наличия алгоритма ЭЦП (ECC160).

Создадим маску и добавим в нее два алгоритма: AES128, как это было показано в уроке 1.1, и ECC160 с аналогичными параметрами:



Так как маска создается вручную, то для последующего использования алгоритма ЭЦП при наложении автозащиты, нужно указать Мастеру автозащиты открытый ключ для созданного алгоритма цифровой подписи. Для этого необходимо экспортировать его из маски в бинарный файл (**Свойства алгоритма ECC160** --> **Ключ ECC160** ---> **Экспорт в bin-файл автозащиты**).



Добавление пользовательских алгоритмов и защищенных ячеек

На данном этапе необходимо добавить в маску набор алгоритмов и защищенных ячеек, достаточный для задания всех лицензионных ограничений.

Одним из возможных подходов является постановка в соответствие отдельных функциональных модулей приложения аппаратным алгоритмам.

Наибольшая эффективность защиты может быть получена именно при использовании **аппаратных алгоритмов** и **защищенных ячеек**. Другие типы элементов маски рекомендуется использовать только в информационных целях.

В отличие от аппаратных алгоритмов, необходимых для работы автозащиты, параметры пользовательских алгоритмов могут быть любыми. Существуют следующие возможности

- Задания временных ограничений на обращение к алгоритму
  - по количеству обращений
  - по астрономическому времени (функционал Guardant Time)
- Привязки определителя алгоритма к уникальному ID ключа
- Установки сервисов чтения/обновления определителя по паролю
- Установки сервисов активации/деактивации алгоритма по паролю
- Ручного задания определителя
- Генерации случайного значения определителя

Установка защиты от терминальных сессий

Использование технологии защищенного обмена между электронным ключом и защищенным приложением на основе сеансового ключа (более подробно речь об этом пойдет в одном из следующих уроков), помимо своей основной функции, позволяет ограничить запуск защищенного приложения **одной копией** (меню **Ключ**-->**Управление сессионными ключами**).

Ключ удаленного обновления

При создании маски автоматически происходит генерация ключа удаленного обновления. Именно его наличие определяет возможность проведения **безопасного удаленного обновления (TRU)** маски ключа.

В рамках процедуры TRU информация для обновления содержимого ключа может передаваться по открытому каналу. Обновление при этом будет доступно лишь тому пользователю, которому оно предназначено.

Запись маски

Подготовка маски ключа закончена, теперь можно прошить ее в ключ.

Основным назначением электронного ключа, в который на данном этапе записывается маска, является использование его для реализации и тестирования защиты при помощи Guardant API, а также наложения автоматической защиты и проведения контрольных испытаний защищенного приложения.

В процессе записи маски в ключ будет автоматически предложено сохранить маску в базу данных. Сохранение произойдет в соответствии с выбранным клиентом. Ввиду указанного назначения ключа, рекомендуется зарегистрировать факт записи маски клиента **Anonymous**, или другого, специально созданного для этой цели.

При необходимости, в процессе подготовки ключа к передаче конечному пользователю, маску можно будет записать повторно для конкретного клиента (или перерегистрировать факт записи маски на него).

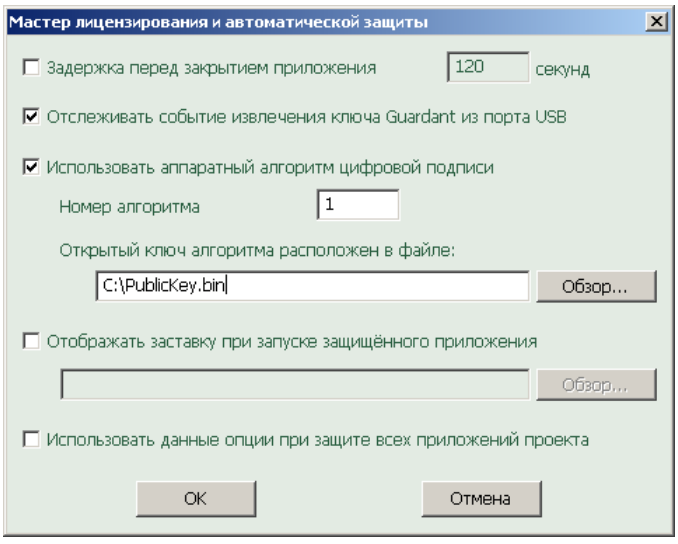
Выбор текущего клиента, для которого производится запись маски в ключ, осуществляется в диалоге **Управление записями клиентов**.

Создание проекта автозащиты

После окончания этапа формирования и записи маски подразумевается реализация защиты при помощи **Guardant API** (подробнее этот этап будет рассмотрен в одном из следующих уроков). Затем следует установка автоматической защиты.

Для создания проекта автозащиты воспользуемся последним пунктом Мастера автозащиты

Практически все действия по настройке проекта автозащиты аналогичны действиям в случае записи маски самим Мастером и описаны в уроке 1.1. Основное отличие составляет необходимость указания открытого ключа цифровой подписи в **Дополнительных параметрах** (в случае использования алгоритма ЭЦП) на заключительном этапе работы Мастера:



В результате будет получен защищенный исполняемый файл, готовый для передачи конечному пользователю.

Контрольные испытания

После окончания процесса защиты исполняемого файла, до передачи продукта конечному пользователю, необходимо провести ряд контрольных испытаний защищенного приложения, в ходе которых проверить

- Корректность работы приложения в целом
- Корректность работы защищенного функционала приложения
- Корректность механизмов лицензирования во всех режимах работы приложения, включая ситуации добавления/изменения/отзыва прав на использование функционала защищенного приложения
- Устойчивость работы защищенного функционала в различных средах и при различных нагрузках

По окончании контрольных испытаний будет получен готовый для передачи конечным пользователям программный продукт.

Следующий этап состоит в тиражировании лицензий и включает в себя запись созданной маски с необходимыми лицензионными ограничениями и регистрации факта записи маски ключа на соответствующего клиента разработчика программного продукта.

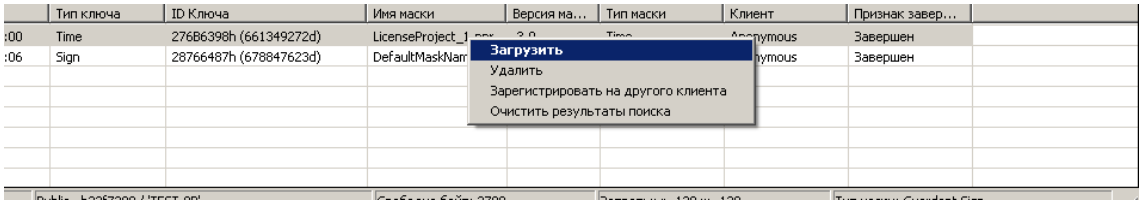
Тиражирование лицензий

Тиражирование лицензий ключа, в общем случае, состоит из двух этапов:

- Выбор/создание клиента (записи о клиенте в БД)
- Запись маски в ключ для выбранного клиента

Функционал ведения базы данных предоставляет утилита программирования ключей GrdUtil, рассмотренная в уроке 1.2. Сохранение информации о фактах записи масок в базу данных производится автоматически. При этом база данных должна быть подключена (Ctrl + D), а в диалоге **Управление записями клиентов** выбрана необходимая запись.

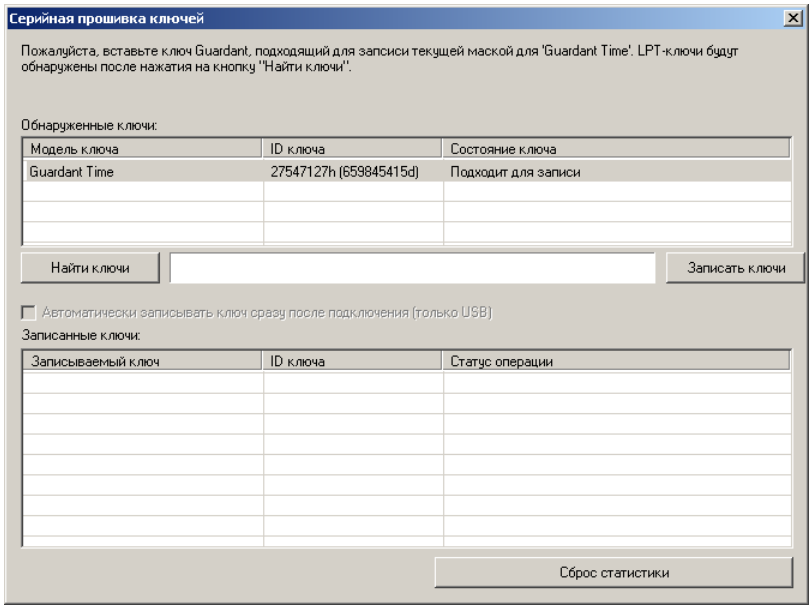
Для записи в ключ очередной копии лицензии необходимо **найти** в базе данных маску ключа, соответствующую проекту защиты приложения, и **загрузить** ее в редактор:



Внести необходимые изменения в маску (задав соответствующие проданной лицензии ограничения в свойствах аппаратных алгоритмов).

Пакетный режим программирования ключей

В целях упрощения процесса предпродажной подготовки рекомендуется воспользоваться **пакетным режимом** программирования ключей (выбрав соответствующий элемент панели инструментов).



В списке обнаруженных ключей необходимо выбрать подлежащие записи, и нажать кнопку **Записать ключи** (или установить отметку **Автоматически записывать маску** при отсутствии ключей в порту USB).

Заключение

В данном уроке был представлен обобщенный порядок действий при подготовке защищаемого приложения и электронных ключей к передаче конечным пользователям.

В результате выполнения уроков серии **1.x** были рассмотрены базовые принципы работы с основными утилитами комплекта разработчика Guardant. Эти навыки необходимы для успешной реализации собственной уникальной системы защиты приложения на основе Guardant API.

Следующие уроки будут посвящены более глубокому изучению приемов, необходимых для создания собственной системы защиты приложения. В частности, в них будут рассмотрены вопросы выбора функционала приложения, подлежащего защите и некоторые особенности реализации защиты при помощи Guardant API.

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