Guardant®

Система защиты от компьютерного пиратства

Эффективная
защита приложений

Урок 2: устройство
электронных ключей Guardant

Содержание

Общее описание электронных ключей

Электронный ключ, в общем случае, представляет собой защищенный от несанкционированного считывания носитель информации. Он имеет широкий спектр применения: начиная от хранения на нем криптографической информации и заканчивая аутентификацией пользователей и лицензированием программного обеспечения.

Функциональность последних поколений электронных ключей значительно повысилась. Помимо стандартных возможностей защищенного хранения данных появилась возможность проведения некоторых математических преобразований внутри ключа с использованием записанных в него данных, а также реализация собственных алгоритмов обработки данных в электронном ключе.

В рамках данного урока электронные ключи будут рассмотрены как с точки зрения разработчика, так и конечного пользователя продукта. Урок является информационным, однако необходим для понимания технологий Guardant и принципов работы с ними.

Используемые термины и обозначения

EEPROM – электрически стираемое программируемое ПЗУ, используемое для хранения данных ключа.

Симметричная криптосистема — криптосистема, в которой для шифрования и расшифрования применяется один и тот же криптографический ключ.

Асимметричная криптосистема (криптосистема с открытым ключом) — криптосистема шифрования или цифровой подписи, в которой для проведения прямого и обратного преобразований используются различные криптографические ключи, один из которых является общедоступным и предназначен для передачи по открытому каналу. Открытый ключ используется для проверки ЭЦП и шифрования сообщений. Для генерации ЭЦП и для расшифрования сообщения используется секретный ключ.

AES128 (AES256) — открытый симметричный алгоритм блочного шифрования (длина ключа 128 (256) бит, длинна блока 128 бит).

ECC160 – криптосистема с открытым ключом, основанная на преобразованиях в группе точек эллиптических кривых (длина секретного ключа 160 бит, длина открытого ключа 320 бит, длина подписываемого сообщения 160 бит, длина вырабатываемой цифровой подписи 320 бит).

GSII64 — закрытый симметричный алгоритм блочного шифрования (длина ключа 128/256 бит, длина блока 64 бита), разработанный специалистами компании «Актив».

HASH64 — алгоритм вычисления хэш-функции на основе GSII64 (длина ключа 128/256 бит). Алгоритм используется для вычисления надежных контрольных сумм, проверки целостности и аутентификации данных.

RND64 — алгоритм генерации псевдослучайных последовательностей длиной 64 бита на основе GSII64 (длина ключа 128/256 бит).

SHA256 – алгоритм вычисления хеш-функции (длина выхода 256 бит).

Устройство ключей Guardant

Внешне электронный ключ Guardant представляет собой USB-брелок, на корпусе которого указана модель ключа (или название организации разработчика / наименование программы продукта, в случае заказных ключей).




С точки зрения конечного пользователя, ключ Guardant – это устройство, необходимое для корректной работы защищенного приложения.

С точки зрения разработчика программного продукта, электронный ключ является носителем секретной информации, способным выполнять криптографические преобразования, на основе чего и производится защита приложения.

Микроконтроллер

Основным элементом электронных ключей Guardant является микроконтроллер. Программа, записанная в нем, осуществляет обработку информации и реализует протокол обмена с драйвером.



Не рекомендуется делать попытки разобрать ключ. Наличие следов вскрытия и других физических повреждений приводит к потере гарантии.

Идентификационный номер ключа

Каждый ключ Guardant содержит уникальный идентификационный номер (ID), который прошивается на этапе производства ключа и не может быть продублирован или изменен. Маркировка ID наносится также на корпус ключа. Она позволяет вести учет электронных ключей, передаваемых клиентам и партнерам, что упрощает техническую поддержку.

Светодиод

Электронные ключи Guardant оборудованы светодиодом, по которому можно судить о корректности работы ключа, а также диагностировать некоторые проблемы, возникающие в процессе его работы (по способу мигания ключа).

Коды доступа

Коды доступа являются одновременно идентификатором разработчика (общий код), по которому защищенное приложение может найти нужный ему ключ, а также паролями для выполнения различных операций Guardant API с ключом:

Код доступа	Назначение
Public Общий код	Показывает принадлежность ключа тому или иному разработчику
Private Read Секретный код чтения	Чтение памяти ключа, выполнение аппаратных алгоритмов, обращение к защищенным ячейкам
Private Write Секретный код записи	Запись данных в память ключа
Private Master Секретный мастер-код	Выполнение специальных операций с ключом (инициализация памяти, установка запретов)

Уникальный набор кодов доступа присваивается разработчику при первом заказе ключей Guardant и прошивается в каждый ключ. Разработчик вводит эти коды в процессе инсталляции комплекта разработчика и использует их при вызове функций Guardant API.

Электронные ключи с определенными кодами доступа может приобрести только лицо/организация-владелец этих кодов.

Организация памяти ключа

Все ключи Guardant поколения Sign/Time имеют 4 кб энергонезависимой памяти (EEPROM) для хранения данных, которая обеспечивает их сохранность в течение порядка 100 лет. Поддерживается неограниченное число сеансов чтения и до 1000 000 сеансов записи в память ключа. Для удобства работы память разбита на логические поля.

Обобщенно эти поля можно разделить на следующие группы:

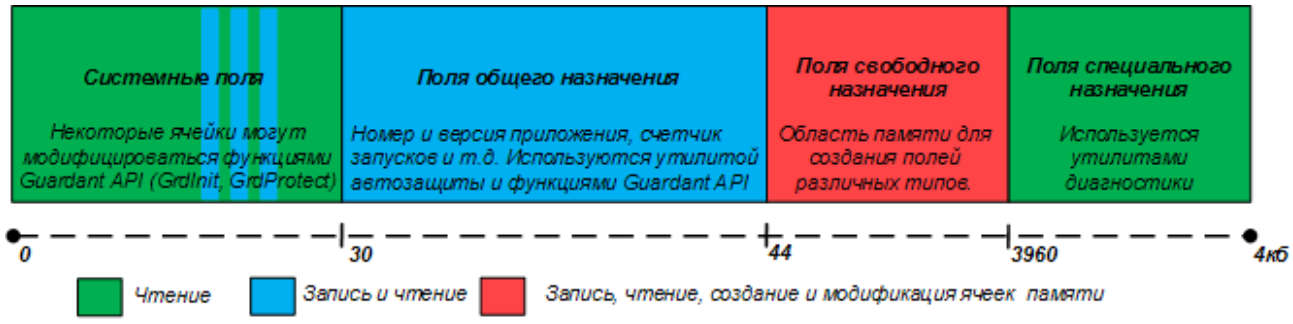


Схема №1.

Для нас интересны в первую очередь **поля общего** и **свободного назначения**. В них располагается информация, используемая в процессе исполнения защищенного приложения. Значения указанных полей можно изменять при помощи утилиты программирования ключей и Guardant API. Обращение к записанным в ключ алгоритмам может выполняться только после наложения запретов на чтение/запись полей памяти (автоматически производится утилитой программирования после записи маски). Структура маски впоследствии может быть изменена только при полном обнулении содержимого полей общего/свободного назначения.

В ключах с функциональностью **Guardant Code** дополнительно содержится защищенная от считывания Flash память для хранения реализуемых разработчиком алгоритмов, а также некоторое количество RAM, необходимой для их исполнения.

Сводная таблица характеристик ключей Guardant

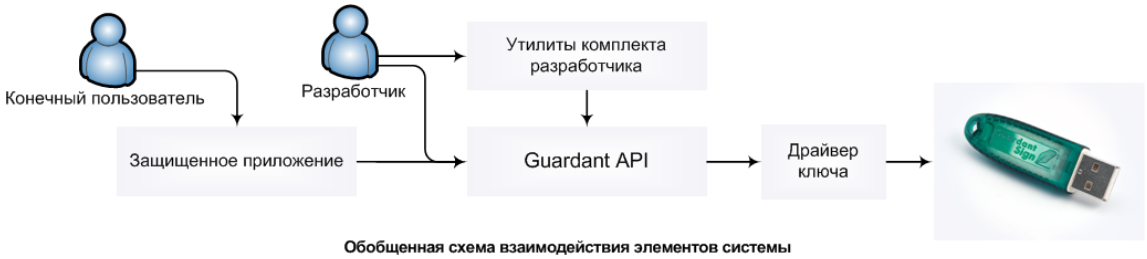
Модель ключа	Sign	Time	Code	Code Time	Sign Net	Time Net
Тип ключа	Локальный				Сетевой	
Платформы	Windows, Linux				Windows, Linux*	
Объем памяти	EEPROM: 4 кб		EEPROM: 4 кб Flash: 128 кб ОЗУ 20 кб		EEPROM: 4 кб	
Процессор	32-bit ARM7 60МГЦ					
Аппаратные алгоритмы	ECC160, AES128, GSII64, SHA256, HASH64, RND64		ECC160, AES128, SHA256		ECC160, AES128, GSII64, SHA256, HASH64, RND64	
Загружаемый код	Нет		Да		Нет	
Защищенные ячейки	Да					
HID-режим	Да					
TRU	Да					
Часы RTC	Нет	Да	Нет	Да	Нет	Да
Интерфейс	USB 2.0/1.1					

Примечание

(*) Под Linux сервер Guardant Net может работать только в среде коммерческой сборки Wine@etersoft.

Схема взаимодействия элементов системы

Основными субъектами взаимодействия являются разработчик и конечный пользователь защищенного приложения. Обобщенная схема взаимодействия субъектов может быть представлена следующим образом:



Обобщенная схема взаимодействия элементов системы

Одной из основных задач, решаемых в случае комплексного подхода к защите приложений при помощи электронных ключей, является организация безопасного обмена данными между всеми субъектами взаимодействия на каждом этапе реализации, эксплуатации и сопровождения защиты (защищенного приложения).

Схема защищенного обмена с приложением

Для организации безопасного обмена данными между ключом и защищенным приложением используется следующая схема:

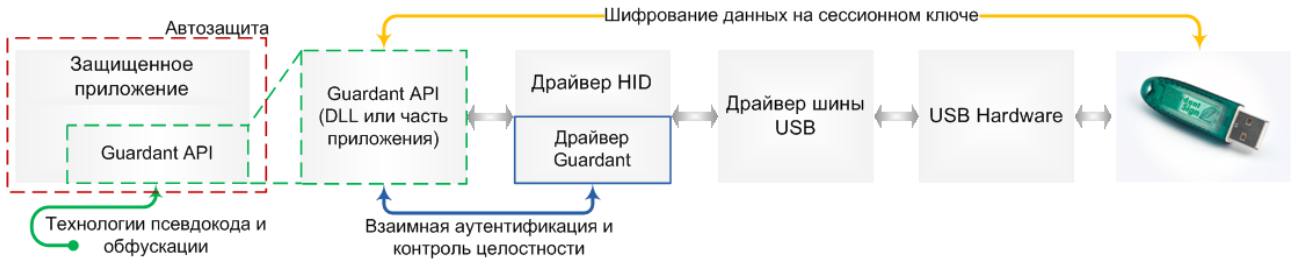


Схема безопасного обмена данными

Данные, передаваемые между Guardant API и электронным ключом, шифруются на **сессионном ключе**, вырабатываемом при инициализации сессии взаимодействия. Таким образом, перехват данных в неконтролируемых каналах (драйвер HID, драйвер шины USB и аппаратная часть USB) не даст нарушителю дополнительной информации для анализа защиты, а также не позволит построить эмулятор ключа на указанном уровне.

Внутренняя логика **Guardant API** защищена современными технологиями псевдокода и обфускации, что значительно затрудняет ее анализ нарушителем.

В качестве дополнительной ступени защиты может выступать **драйвер Guardant**. В нем реализованы алгоритмы взаимной аутентификации и контроля целостности с Guardant API. Именно поэтому рекомендуется по возможности использовать ключ в режиме работы с драйвером Guardant.

Легко видеть, что **наиболее уязвимым местом** в защите приложения при помощи электронного ключа является код самого защищаемого приложения, в частности, точки вызова функций Guardant API. Вследствие этого настоятельно рекомендуется использовать **Автозащиту Guardant** в совокупности с собственными механизмами защиты. В силу динамичности развития автозащиты, существует возможность обновления защиты приложения путем наложения новой версии автозащиты, без необходимости внесения значительных изменений в общую схему защиты и вызовы Guardant API.

Алгоритм выработки сеансового ключа

Традиционная схема защиты приложений с помощью электронного ключа может быть скомпрометирована посредством построения табличного эмулятора. В силу того, что защищенное приложение зачастую содержит конечное число обращений к ключу, при накоплении определённой статистики возможно построение эмулятора USB драйвера, который на каждый вопрос будет посылать известный ответ из таблицы подстановки.

Для противодействия этому специалистами компании «Актив» была разработана уникальная схема генерации сессионных ключей для симметричного блочного шифра, позволяющая сделать уникальным весь трафик, передаваемый в рамках каждой сессии между приложением и ключом. Указанный метод делает бесполезным встраивание логгеров в цепочку обмена, что значительно уменьшает число потенциальных атак со стороны злоумышленников.

Использование закрытых криптографических алгоритмов

При использовании ключей поколения Sign/Time существует возможность использования как открытых криптоалгоритмов (AES128, SHA256, ECC160), так и закрытого криптографического алгоритма.

Для ключей **Guardant Sign** и **Guardant Time** закрытым алгоритмом является GSII64. Это симметричный алгоритм с длиной ключа 128 или 256 бит. Особенность его использования в том, что для потенциального нарушителя он является черным ящиком. Это значит, что атаку на алгоритм методом полного перебора можно осуществить только при непосредственном обмене с самим ключом. В этом случае время на реализацию атаки на алгоритм значительно возрастает за счет ограничения канала передачи и вычислительной способности микропроцессора ключа.

В случае использования ключей поколения **Guardant Code**, разработчиком защиты может быть реализован собственный алгоритм обработки данных, а также загружен в ключ любой из известных существующих криптографических алгоритмов. В обоих случаях вероятность успешной атаки на алгоритм в электронном ключе становится пренебрежимо малой.

Технологии псевдокода и обфускации

Защита функционала Guardant API основана на технологии псевдокода (выполнение происходит в среде специальной виртуальной машины), а также технологии генерации полиморфного кода (запутывание исполняемого кода). Более подробная информация расположена на корпоративном сайте компании: **часть1** и **часть2**.

Указанные технологии значительно усложняют использование средств отладки (в том числе, специализированных) и, соответственно, анализ защищенного приложения.

Взаимная аутентификация и контроль целостности

Драйвер Guardant работает в нулевом кольце защиты (Ring 0). При работе осуществляется взаимная проверка целостности и аутентичности библиотеки функций Guardant API и драйвера Guardant.

Указанные меры значительно затрудняют отладку библиотеки функций Guardant API и делают практически невозможной подмену отдельных ее модулей.

Вследствие этого рекомендуется по возможности использовать электронные ключи в режиме работы с драйвером Guardant.

Заключение

В данном уроке представлена основная информация о технологиях Guardant. Она необходима для понимания общих принципов комплексного подхода к защите приложений при помощи электронных ключей. Эта информация призвана указать те особенности, на которые должен обратить внимание разработчик при реализации собственной защиты.

При вскрытии защиты программного обеспечения большинство атак производится не на технологии, использованные для защиты, а непосредственно на реализацию защиты.

Разработчики компании «Актив» попытались сделать все для обеспечения возможности создания действительно надежной защиты приложений на основе электронных ключей.

О том, как наиболее грамотно реализовать множество собственных механизмов защиты, речь пойдет в следующих уроках.

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