Общее описание

Аппаратными алгоритмами принято называть алгоритмы, реализующие преобразования данных (кодирование и декодирование, вычисление хэш-функции и т.п.) внутри микроконтроллера ключа. При такой реализации алгоритмов ни сам метод преобразования, ни ключи шифрования не покидают памяти микроконтроллера, что осложняет анализ этих алгоритмов извне. По сути, такая реализация алгоритмов делает электронный ключ аппаратным «черным ящиком», анализ которого возможен только по входящим и исходящим данным.
Важная информация
Реализация возможности обработки данных вне процессора и памяти компьютера аппаратными алгоритмами является основной защитной функцией электронных ключей Guardant.

Безопасность памяти ключа

Все данные, которые хранятся внутри ключа (Flash, RAM, EEPROM) защищены от чтения несанкционированного изменения аппаратно архитектурой самого микроконтроллером. Получить доступ к этой памяти без уничтожения ее содержимого невозможно.

Аппаратные запреты

Часть энергонезависимой памяти ключей Guardant недоступна ни для чтения, ни для записи, часть доступна только для чтения. Остальная память доступна полностью и для чтения, и для записи.
Естественно, хакеры проявляют повышенный интерес к содержимому памяти любых ключей. Ведь считав информацию из ключа, можно создать его эмулятор или полную аппаратную копию. EEPROM-память расположена внутри микроконтроллера, а значит защищена так же хорошо, как и сама микропрограмма электронного ключа. С помощью Guardant API, Guardant Code API или GrdUtil к ней можно получать доступ.
Энергонезависимая память, используемая в ключах Guardant, позволяет устанавливать аппаратные запреты чтения/записи ее содержимого. Скопировать содержимое области памяти, на которую наложен запрет, программным способом невозможно – для этого попросту нет никаких средств. Ключ просто не отвечает на запрос на чтение/запись защищенных областей его памяти.
Установка аппаратных запретов производится на нижнем, аппаратном уровне – это гарантирует невозможность их обхода программными средствами.
Аппаратные запреты можно устанавливать на любую область доступной памяти ключей, снимать запреты, расширять или сужать границы защищенной памяти. На дескрипторы создаваемых в ключах Guardant аппаратных алгоритмов и защищенных ячеек аппаратные запреты чтения и записи утилитой программирования устанавливаются по умолчанию – в этом заключается гарантия того, что аппаратные алгоритмы ключа Guardant не могут быть считаны или продублированы. При программировании ключей утилитами собственной разработки необходимо заботиться о правильной установке аппаратных запретов.

Аппаратная блокировка терминальных сеансов

Наличие механизма шифрования трафика между электронным ключом и Guardant API дает возможность на аппаратном уровне ограничить использование локального ключа одновременно из нескольких терминальных сеансов.
В обычном режиме для каждого сеанса взаимодействия с ключом, ограниченного функциями GrdLogin() и GrdLogout(), вырабатывается отдельный сессионный ключ шифрования трафика. Количество сессионных ключей определяется количеством открытых сеансов.
Соответственно, электронный ключ может одновременно использоваться большим количеством одновременно работающих копий защищенного приложения, в том числе из терминальных сеансов. Такая ситуация нарушает принцип «один ключ – одна лицензия», поэтому возникает естественное желание не разрешать запуск более одной копии приложения с одним ключом. Как правило, эта задача решается программными методами, такими как создание глобальных объектов в памяти при запуске приложения и проверки их существования при запуске дополнительных копий. Также существуют методы отслеживания лишних запущенных копий, использующие запись случайных чисел в память ключа и проверку их наличия в процессе работы.
Технология, реализованная в ключах Guardant, позволяет обеспечивать работоспособность только одной копии запущенного приложения. Для этого электронный ключ при предпродажном программировании должен быть переведен в один из режимов ограничения количества сеансовых ключей:

- Один сессионный ключ для автозащиты
- Один сессионный ключ для API
- Два сессионных ключа (по одному для автозащиты и API)

Схема работы в режимах ограничения следующая:
При запуске 1-й копии приложения и выполнении функции **GrdLogin()** генерируется один или два сессионных ключа, которые хранятся в оперативной памяти электронного ключа на протяжении сеанса.
Если происходит запуск 2-й копии приложения, использующего тот же электронный ключ, при выполнении функции **GrdLogin()** происходит выработка нового сессионного ключа. При этом новый ключ заместит собой тот, что был сгенерирован при запуске 1-й копии. Первая копия приложения не сможет обмениваться данными с электронным ключом, поскольку будет использовать устаревший сессионный ключ, и станет неработоспособной.