Алгоритмы защиты и лицензирования

Аппаратные алгоритмы – это математические функции преобразования данных, выполняющиеся в самом ключе, без использования ресурсов компьютера.

Аппаратные алгоритмы служат для кодирования информации, необходимой для работы защищенного приложения. При правильной организации системы защиты использование аппаратных алгоритмов делает бессмысленным удаление из тела приложения вызовы функций API: в этом случае не произойдет декодирование нужных приложению данных. Кроме того, сам факт наличия аппаратных алгоритмов сильно усложняет логику работы электронных ключей Guardant.

Использование аппаратных алгоритмов – это основной путь создания качественной и эффективной защиты приложения.

Аппаратные алгоритмы реализуются микропрограммой ключа, записанной в микроконтроллер, в сочетании с дескрипторами, которые хранятся в памяти электронного ключа. Микропрограмма ключа Guardant является неизменяемой частью алгоритма, ее невозможно ни считать, ни модифицировать. Для формирования конкретного вида алгоритма и его параметров, а также для управления его «поведением», служит дескрипт ор – набор данных, хранящихся в памяти ключа, доступной разработчику защиты.

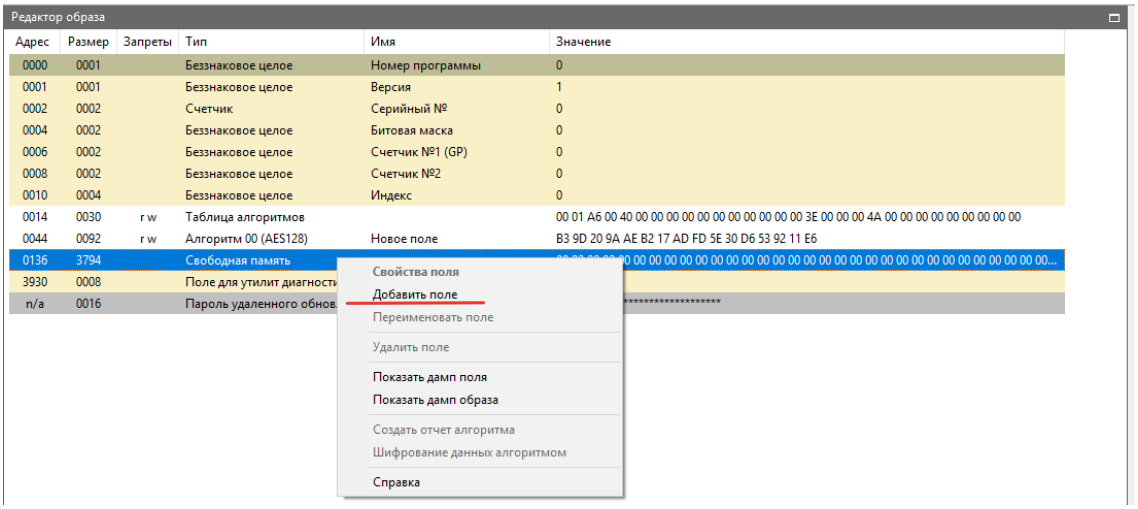
GrdUtil.exe позволяет создавать, редактировать и удалять дескрипторы аппаратных алгоритмов.

Диалог создания алгоритма *(при работе с аппаратными алгоритмами подразумевается, что все действия выполняются с их дескрипторами)* выполнен в виде мастера, состоящего из нескольких страниц:

- Добавить алгоритм (см. ниже)
- Свойства алгоритма
- Временные зависимости (только для ключей типа Time)
- Определитель алгоритма (кроме алгоритма ECC160)
- Цифровая подпись ECC160 (только для алгоритма ECC160)

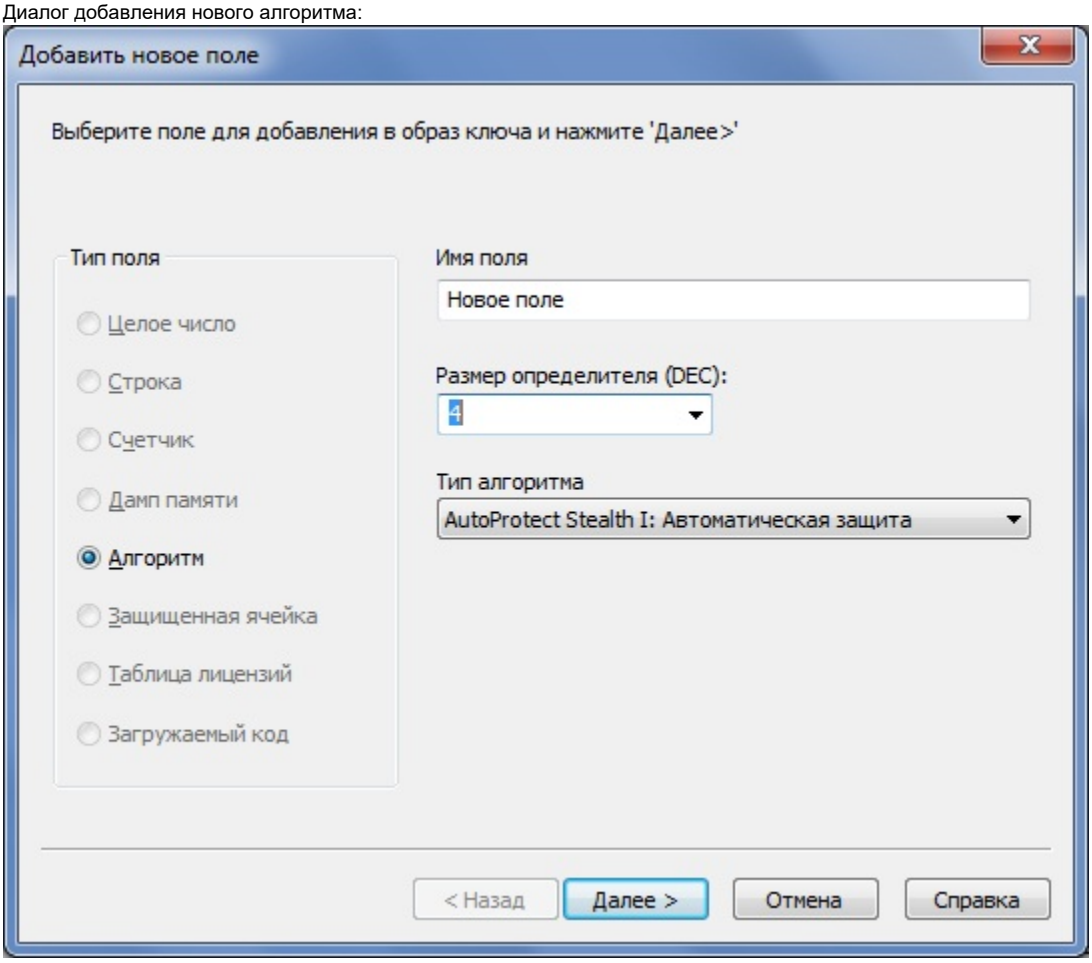
Переход к следующей странице происходит при помощи нажатия кнопки **[Далее]** после выполнения текущего диалога.

Чтобы создать аппаратный алгоритм выберите в ***Редакторе образа*** запись, далее выберите команду **Образ ключа | (Поле) Добавить поле** либо нажмите на запись правой клавишей мыши и в выпадающем меню выберите команду **Добавить поле.**



В появившемся диалоге **Добавить новое поле** выберите тип поля **Алгоритм**, задайте имя и тип нового алгоритма, а также размер его определителя.

Диалог добавления нового алгоритма:



Тип алгоритма

Выберите тип алгоритма при помощи разворачивающегося списка.

Тип создаваемого алгоритма зависит от образа ключа:

Тип образа	Тип алгоритма
Guardant Sign/ Sign Net Guardant Time/ Time Net	1. Симметричное шифрование: GSII64, в т. ч. особые режимы алгоритма только для кодирования и декодирования 2. Симметричное шифрование: AES128, в т. ч. особые режимы алгоритма только для кодирования и декодирования 3. Выработка ЭЦП: ECC160 4. Выработка хэш-функции: HASH64 5. Выработка хэш-функции: SHA256 6. Генерация случайных чисел: RND64
Guardant Code/ Code Time	1. Симметричное шифрование: AES128, в т. ч. особые режимы алгоритма только для кодирования и декодирования 2. Выработка ЭЦП: ECC160 3. Выработка хэш-функции: SHA256

Guardant Stealth III / Net III	1. Симметричное шифрование: GSII64 2. Выработка хэш-функции: HASH64 3. Генерация случайных чисел: RND64
Guardant Stealth II / Net II	1. Симметричное шифрование: GSII64 2. Однонаправленный алгоритм Stealth и его разновидности: Fast, Random, AutoProtect
Guardant Stealth / Net	Однонаправленный алгоритм Stealth и его разновидности: Fast, Random, AutoProtect
Guardant Fidus	-

Размер определителя

Определитель – основная составляющая дескриптора аппаратного алгоритма, которая задает конкретный вид функции преобразования. Определители алгоритмов в современных ключах Guardant имеют фиксированную четную длину, зависящую от типа алгоритма. С помощью GrdUtil.exe можно задавать размер определителя и редактировать его вид.

Чтобы выбрать (или задать – для однонаправленного алгоритма Stealth) размер определителя, воспользуйтесь одноименным комбинированным полем-списком, слева от которого указывается система счисления.

Размер определителя зависит от типа алгоритма:

Тип алгоритма	**Размер определителя, байтов**
Симметричное шифрование: AES128	16
Выработка ЭЦП: ECC160	20
Симметричное шифрование: GSII64	16 или 32
Выработка хэш-функции: SHA256	-
Выработка хэш-функции: HASH64	16 или 32
Генерация случайных чисел: RND64	16 или 32
Однонаправленный Stealth – устаревш.	4 – 200 (оптимально - 32)