GrdTRU_SetKey

Функция(метод) **GrdTRU_SetKey** инициализирует память ключа и записывает 128-битовый секретный ключ для алгоритма **GSII64,** который будет использоваться для удаленного программирования. Для использования этой функции(метода) в защищенное хранилище функцией(методом) Grd SetAccessCodes должен быть записан **Private master code**.

C

```
int GRD_API GrdTRU_SetKey(
  HANDLE hGrd,
  void *pGSII64_Key128
);
```

hGrd	хэндл, через который будет выполнена данная операция
pGSII64_Key128	указатель на буфер, содержащий 128-битовый уникальный секретный ключ шифрования алгоритма **GSII64**

Отсутствует.
Функция **GrdTRU_SetKey** инициализирует память ключа, т. е. сбрасывает запреты на чтение и запись, обнуляет количество аппаратных алгоритмов и удаляет все данные из области данных пользователя (т. е. все дескрипторы алгоритмов, защищенные ячейки и другие данные, хранящиеся в этой области). Функция используется при реорганизации памяти ключей «вручную» (без помощи программы GRDUTIL) в тех случаях, когда планируется использовать технологию Trusted Remote Update для ключей Guardant Stealth/Net III.

Не рекомендуется без особой необходимости пользоваться этой функцией, т.к. неаккуратное использование может принести немало вреда (например, можно «случайно» стереть важные данные или дескрипторы алгоритмов). Для реорганизации памяти ключа лучше использовать программу GRDUTIL.

128-битовый секретный ключ обеспечивает защищенность обмена данными при удаленном программировании. Этот ключ хранится в недоступной для пользователя и разработчика области памяти электронного ключа и не может быть прочитан. Он должен быть сгенерирован и записан в ключ на этапе предпродажной подготовки. Копия этого секретного ключа обязательно должна оставаться у разработчика. В противном случае удаленное программирование по технологии TRU будет невозможно. База данных, содержащая ID ключей и соответствующие им секретные ключи удаленного программирования обязательно должна быть защищена от доступа неавторизованного персонала.

При программировании ключей для которых будет использоваться TRU с помощью GRDUTIL есть два варианта:

1. С использованием базы данных секретные ключи генерируются утилитой автоматически для каждого электронного ключа и записываются в его память, а также сохраняются в базе данных, в привязке к соответствующему ID, при каждой прошивке ключа.
2. Без использования базы данных утилита генерирует секретный ключ один раз, при создании маски и он записывается во все электронные ключи, в которые программируется данная маска.

С точки зрения повышения защищенности, первый вариант (с использованием базы данных) предпочтительней, так как в каждом ключе оказывается уникальный ключ шифрования TRU и компрометация одного из них никак не скажется на остальных. Чего не скажешь о втором варианте, в котором компрометация одного секретного ключа TRU даст возможность злоумышленнику возможность перепрограммировать любой другой ключ, прошитый по этой маске.

Однако, даже в случае невозможности применения базы данных, безопасность обновления через TRU все равно гораздо выше, так как на недоверенном удаленном компьютере, секретный ключ шифрования никогда не покидает электронный ключ и записываемые в ключ данные передаются туда шифрованными. Т.е., даже если клиент сам хакер, или пригласил хакера для того чтобы он, любом доступным им способом, «подсмотрел» или изменил в что-либо в записываемых в электронный ключ данных, все рано смотреть там будет нечего – все зашифровано, изменить ничего нельзя - все подписано. Применить число-ответ в другой ключ тоже нельзя – все контролируется.

C#

```
public static GrdE GrdTRU_SetKey(Handle grdHandle, byte[] gsii64_Key128)
```

grdHandle [in]

Тип: Handle

Нэндл, через который будет выполнена данная операция.

gsii64_Key128 [in]

Тип: byte [ ]

Указатель на буфер, который содержит уникальный секретный ключ шифрования алгоритма GSII64 размером 128 бит.

Отсутствует.
Метод **GrdTRU_SetKey** инициализирует память ключа, т. е. сбрасывает запреты на чтение и запись, обнуляет количество аппаратных алгоритмов и удаляет все данные из области данных пользователя (т. е. все дескрипторы алгоритмов, защищенные ячейки и другие данные, хранящиеся в этой области). Метод используется при реорганизации памяти ключей «вручную» (без помощи программы GRDUTIL) в тех случаях, когда планируется использовать технологию Trusted Remote Update для ключей Guardant Stealth/Net III.

Не рекомендуется без особой необходимости пользоваться этим методом, т.к. неаккуратное его использование может принести немало вреда (например, можно «случайно» стереть важные данные или дескрипторы алгоритмов). Для реорганизации памяти ключа лучше использовать программу GRDUTIL.

128-битовый секретный ключ обеспечивает защищенность обмена данными при удаленном программировании. Этот ключ хранится в недоступной для пользователя и разработчика области памяти электронного ключа и не может быть прочитан. Он должен быть сгенерирован и записан в ключ на этапе предпродажной подготовки. Копия этого секретного ключа обязательно должна оставаться у разработчика. В противном случае удаленное программирование по технологии TRU будет невозможно. База данных, содержащая ID ключей и соответствующие им секретные ключи удаленного программирования обязательно должна быть защищена от доступа неавторизованного персонала.

При программировании ключей для которых будет использоваться TRU с помощью GRDUTIL есть два варианта:

1. С использованием базы данных секретные ключи генерируются утилитой автоматически для каждого электронного ключа и записываются в его память, а также сохраняются в базе данных, в привязке к соответствующему ID, при каждой прошивке ключа.
2. Без использования базы данных утилита генерирует секретный ключ один раз, при создании маски и он записывается во все электронные ключи, в которые программируется данная маска.

С точки зрения повышения защищенности, первый вариант (с использованием базы данных) предпочтительней, так как в каждом ключе оказывается уникальный ключ шифрования TRU и компрометация одного из них никак не скажется на остальных. Чего не скажешь о втором варианте, в котором компрометация одного секретного ключа TRU даст возможность злоумышленнику возможность перепрограммировать любой другой ключ, прошитый по этой маске.

Однако, даже в случае невозможности применения базы данных, безопасность обновления через TRU все равно гораздо выше, так как на недоверенном удаленном компьютере, секретный ключ шифрования никогда не покидает электронный ключ и записываемые в ключ данные передаются туда шифрованными. Т.е., даже если клиент сам хакер, или пригласил хакера для того чтобы он, любом доступным им способом, «подсмотрел» или изменил в что-либо в записываемых в электронный ключ данных, все рано смотреть там будет нечего – все зашифровано, изменить ничего нельзя - все подписано. Применить число-ответ в другой ключ тоже нельзя – все контролируется.

Java

```
public static GrdE GrdTRU_SetKey(Handle grdHandle, byte[] gsii64_Key128)
```

grdHandle [in]

Тип: Handle

Нэндл, через который будет выполнена данная операция.

gsii64_Key128 [in]

Тип: byte [ ]

Указатель на буфер, который содержит уникальный секретный ключ шифрования алгоритма GSII64 размером 128 бит.
Отсутствует.
Метод **GrdTRU_SetKey** инициализирует память ключа, т. е. сбрасывает запреты на чтение и запись, обнуляет количество аппаратных алгоритмов и удаляет все данные из области данных пользователя (т. е. все дескрипторы алгоритмов, защищенные ячейки и другие данные, хранящиеся в этой области). Метод используется при реорганизации памяти ключей «вручную» (без помощи программы GRDUTIL) в тех случаях, когда планируется использовать технологию Trusted Remote Update для ключей Guardant Stealth/Net III.

Не рекомендуется без особой необходимости пользоваться этим методом, т.к. неаккуратное его использование может принести немало вреда (например, можно «случайно» стереть важные данные или дескрипторы алгоритмов). Для реорганизации памяти ключа лучше использовать программу GRDUTIL.

128-битовый секретный ключ обеспечивает защищенность обмена данными при удаленном программировании. Этот ключ хранится в недоступной для пользователя и разработчика области памяти электронного ключа и не может быть прочитан. Он должен быть сгенерирован и записан в ключ на этапе предпродажной подготовки. Копия этого секретного ключа обязательно должна оставаться у разработчика. В противном случае удаленное программирование по технологии TRU будет невозможно. База данных, содержащая ID ключей и соответствующие им секретные ключи удаленного программирования обязательно должна быть защищена от доступа неавторизованного персонала.

При программировании ключей для которых будет использоваться TRU с помощью GRDUTIL есть два варианта:

1. С использованием базы данных секретные ключи генерируются утилитой автоматически для каждого электронного ключа и записываются в его память, а также сохраняются в базе данных, в привязке к соответствующему ID, при каждой прошивке ключа.

2. Без использования базы данных утилита генерирует секретный ключ один раз, при создании маски и он записывается во все электронные ключи, в которые программируется данная маска.

С точки зрения повышения защищенности, первый вариант (с использованием базы данных) предпочтительней, так как в каждом ключе оказывается уникальный ключ шифрования TRU и компрометация одного из них никак не скажется на остальных. Чего не скажешь о втором варианте, в котором компрометация одного секретного ключа TRU даст возможность злоумышленнику возможность перепрограммировать любой другой ключ, прошитый по этой маске.

Однако, даже в случае невозможности применения базы данных, безопасность обновления через TRU все равно гораздо выше, так как на недоверенном удаленном компьютере, секретный ключ шифрования никогда не покидает электронный ключ и записываемые в ключ данные передаются туда шифрованными. Т.е., даже если клиент сам хакер, или пригласил хакера для того чтобы он, любом доступным им способом, «подсмотрел» или изменил в что-либо в записываемых в электронный ключ данных, все рано смотреть там будет нечего – все зашифровано, изменить ничего нельзя - все подписано. Применить число-ответ в другой ключ тоже нельзя – все контролируется.