

GrdTRU_DecryptQuestionTime

Функция(метод) **GrdTRU_DecryptQuestionTime** расшифровывает число-вопрос и проверяет подлинность его и остальных присланных с удаленного компьютера параметров. Функция(метод) является аналогом [GrdTRU_DecryptQuestion](#) для ключа Guardant Time.

C

```
int GRD_API GrdTRU_DecryptQuestionTime(
    HANDLE hGrd,
    DWORD dwAlgoNum_GSII64,
    DWORD dwAlgoNum_HashS3,
    void *pQuestion,
    DWORD dwID,
    DWORD dwPublic,
    QWORD *pqwDongleTime,
    QWORD *pqwDeadTimes,
    DWORD dwDeadTimesNumbers,
    void *pHash
);
```

<i>hGrd</i>	хэндл, через который будет выполнена данная операция
<i>dwAlgoNum_GSII64</i>	номер аппаратного алгоритма типа GSII64 , который будет использоваться для расшифровывания числа-вопроса
<i>dwAlgoNum_Hash64</i>	номер аппаратного алгоритма Hash64 , который будет использоваться для проверки подлинности числа-вопроса на основании MAC
<i>pQuestion</i>	буфер, содержащий присланное удаленным пользователем число-вопрос. Длина буфера 8 байт
<i>dwID</i>	ID ключа удаленного пользователя, для которого будет произведена операция
<i>dwPublic</i>	численное значение Public Code ключа удаленного пользователя, для которого будет произведена операция
<i>pqwDongleTime</i>	зашифрованное значение времени ключа. 8 байт.
<i>pqwDeadTimes</i>	зашифрованные значения времён жизни алгоритмов. По 8 байт
<i>dwDeadTimesNumbers</i>	количество 8 байтовых элементов в <i>pqwDeadTimes</i>
<i>pHash</i>	буфер, содержащий MAC, вычисленный на ключе удаленного пользователя. Длина буфера 8 байт
<i>pQuestion</i>	после выполнения функции в этот буфер возвращается расшифрованное число-вопрос. Длина буфера 8 байт

Возможные ошибки

GrdE_SystemDataCorrupted	Системные данные TRU повреждены. (Секретный ключ удаленного программирования отсутствует)
GrdE_NoQuestion	Число-вопрос не было сгенерировано или было регенерировано до записи числа ответа
GrdE_InvalidData	Неверный формат данных для удаленного программирования
GrdE_QuestionOK	Число-вопрос уже было сгенерировано, ключ ожидает данных для удаленного программирования
GrdE_UpdateNotComplete	Ошибка при записи данных удаленного программирования. Операция не была завершена
GrdE_InvalidHash	Неверное значение MAC (Message Authentication Code)
	Набор ошибок Guardant API

Использование функции **GrdTRU_DecryptQuestionTime** позволяет получить число-вопрос в расшифрованном виде и убедиться в том, что оно действительно было сгенерировано на ключе с ID равным *dwID* и Public Code равным *dwPublic*. Функция является аналогом [GrdTRU_DecryptQuestion](#) для ключа Guardant Time и используется в случаях, когда необходимо продлевать время работы защищенного приложения.

Если число-вопрос расшифровано правильно и проверка подлинности прошла успешно, функция возвращает [GrdE_OK](#).

Расшифрованное 8-байтовое число-вопрос помещается в тот же буфер *pQuestion*, в котором находилось зашифрованное число-вопрос. Расшифрованное число-вопрос необходимо для генерации ответа, поэтому его нужно сохранить для дальнейшего использования.

Расшифрование числа-вопроса производится аппаратным алгоритмом типа **GSII64** с номером, задаваемым *dwAlgoNum_GSII64*. На момент расшифровывания этот алгоритм должен быть создан в ключе, находящемся у разработчика. В качестве определителя должен использоваться секретный 128-битовый ключ, который был сгенерирован и прошит в ключ удаленного пользователя при предпродажной подготовке функцией [GrdTRU_SetKey](#) для ключа с ID, равным *dwID*. При использовании GRDUTIL этот ключ берется из базы данных.

Проверка подлинности числа-вопроса производится аппаратным алгоритмом типа **Hash64** с номером *dwAlgoNum_Hash64*. На момент проверки этот алгоритм должен быть создан в ключе, находящемся у разработчика. В качестве определителя должен использоваться секретный 128-битовый ключ, который был сгенерирован и прошит в ключ удаленного пользователя при предпродажной подготовке функцией [GrdTRU_SetKey](#) для ключа с ID равным *dwID*. При использовании GRDUTIL этот ключ берется из базы данных.

Рабочий ключ, находящийся у разработчика, инициализировать функцией [GrdTRU_SetKey](#) с секретным ключом, таким же как у удаленного пользователя, необязательно. Все преобразования делаются на заранее запрограммированных алгоритмах, номера которых указываются в параметрах указываемых в параметрах *dwAlgoNum_GSII64* и *dwAlgoNum_Hash64*.

После расшифрования в *pqwDongleTime* содержится расшифрованное время ключа из микросхемы таймера, а в *pqwDeadTimes* расшифрованное время деактивации алгоритмов.

Два старших байта параметра *pqwDongleTime* - нули. 2 старших байта элементов массива *pqwDeadTimes* - числовое имя алгоритма. Остальные 6 байт - время жизни соответствующего алгоритма.

Время жизни (8 байт) имеет следующий формат:

1, 2 байты	числовое имя алгоритма (либо нули, если время из микросхемы таймера);
3 байт	год от 2000 (08 для 2008 года);
4 байт	месяц года (01 - январь, 02 - февраль, ...);
5 байт	день месяца (1 - 31);
6 байт	часы (0 - 23);
7 байт	минуты (0 - 59);
8 байт	секунды (0 - 59)

C#

```
public static GrdE GrdTRU_DecryptQuestionTime(Handle grdHandle, GrdAlgNum algNum_GSII64, GrdAlgNum
algNum_HashS3, byte[] question,
        uint id, uint publicCode, ref ulong dongleTime, ulong[] deadTimes, int deadTimesNumbers, byte[] hash)
```

grdHandle [in]

Тип: [Handle](#)

Нэндл, через который будет выполнена данная операция.

algNum_GSII64 [in]

Тип: [GrdAlgNum](#)

Номер аппаратного алгоритма типа **GSII64**, который будет использоваться для расшифровывания числа-вопроса.

algNum_HashS3 [in]

Тип: [GrdAlgNum](#)

Номер аппаратного алгоритма **Hash64**, который будет использоваться для проверки подлинности числа-вопроса на основании MAC.

question [in]

Тип: byte []

Буфер, содержащий присланное удаленным пользователем число-вопрос.

id [in]

Тип: uint

ID ключа удаленного пользователя, для которого будет произведена операция.

publicCode [in]

Тип: uint

Численное значение Public Code ключа удаленного пользователя, для которого будет произведена операция.

dongleTime [in]

Тип: `ulong`

Зашифрованное значение времени ключа.

deadTimes [in]

Тип: `ulong [ ]`

Зашифрованное значение времен жизни алгоритмов.

deadTimesNumbers [in]

Тип: `int`

Количество элементов, возвращаемых в *deadTimes*.

hash [in]

Тип: `byte [ ]`

Буфер, содержащий MAC, вычисленный на ключе удаленного пользователя.

pQuestion	после выполнения функции в этот буфер возвращается расшифрованное число-вопрос. Длина буфера 8 байт
-----------	---

Возможные ошибки

GrdE.SystemDataCorrupted	Системные данные TRU повреждены. (Секретный ключ удаленного программирования отсутствует)
GrdE.NoQuestion	Число-вопрос не было сгенерировано или было регенерировано до записи числа ответа
GrdE.InvalidData	Неверный формат данных для удаленного программирования
GrdE.QuestionOK	Число-вопрос уже было сгенерировано, ключ ожидает данных для удаленного программирования
GrdE.UpdateNotComplete	Ошибка при записи данных удаленного программирования. Операция не была завершена
GrdE.InvalidHash	Неверное значение MAC (Message Authentication Code)
	Набор ошибок Guardant API

Использование метода **GrdTRU_DecryptQuestionTime** позволяет получить число-вопрос в расшифрованном виде и убедиться в том, что оно действительно было сгенерировано на ключе с ID равным *id* и Public Code равным *publicCode*. Метод является аналогом [GrdTRU_DecryptQuestion](#) для ключа Guardant Time и используется в случаях, когда необходимо продлевать время работы защищенного приложения.

Если число-вопрос расшифровано правильно и проверка подлинности прошла успешно, функция возвращает [GrdE.OK](#).

Расшифрованное 8-байтовое число-вопрос помещается в тот же буфер *question*, в котором находилось зашифрованное число-вопрос. Расшифрованное число-вопрос необходимо для генерации ответа, поэтому его нужно сохранить для дальнейшего использования.

Расшифровка числа-вопроса производится аппаратным алгоритмом типа **GSII64** с номером, задаваемым [GrdAN.GSII64](#). На момент расшифровки этот алгоритм должен быть создан в ключе, находящемся у разработчика. В качестве определителя должен использоваться секретный 128-битовый ключ, который был сгенерирован и прошит в ключ удаленного пользователя при предпродажной подготовке методом [GrdTRU_SetKey](#) для ключа с ID, равным *id*. При использовании GRDUTIL этот ключ берется из базы данных.

Проверка подлинности числа-вопроса производится аппаратным алгоритмом типа **Hash64** с номером [GrdAN.Hash64](#). На момент проверки этот алгоритм должен быть создан в ключе, находящемся у разработчика. В качестве определителя должен использоваться секретный 128-битовый ключ, который был сгенерирован и прошит в ключ удаленного пользователя при предпродажной подготовке методом [GrdTRU_SetKey](#) для ключа с ID равным *id*. При использовании GRDUTIL этот ключ берется из базы данных.

Рабочий ключ, находящийся у разработчика, инициализировать методом [GrdTRU_SetKey](#) с секретным ключом, таким же как у удаленного пользователя, необязательно. Все преобразования делаются на заранее запрограммированных алгоритмах, номера которых указываются в параметрах указываемых в параметрах [GrdAN.GSII64](#) и [GrdAN.Hash64](#).

После расшифровки в *dongleTime* содержится расшифрованное время ключа из микросхемы таймера, а в *deadTimes* расшифрованное время деактивации алгоритмов.

Два старших байта параметра *dongleTime* - нули. 2 старших байта элементов массива *deadTimes* - числовое имя алгоритма. Остальные 6 байт - время жизни соответствующего алгоритма.

Время жизни (8 байт) имеет следующий формат:

1, 2 байты	числовое имя алгоритма (либо нули, если время из микросхемы таймера);
3 байт	год от 2000 (08 для 2008 года);
4 байт	месяц года (01 - январь, 02 - февраль, ...);
5 байт	день месяца (1 - 31);
6 байт	часы (0 - 23);
7 байт	минуты (0 - 59);
8 байт	секунды (0 - 59)

Java

```
public static GrdE GrdTRU_DecryptQuestionTime(Handle grdHandle, int algoNum_GSII64, int algoNum_HashS3,
        byte[] question, int id, int publicCode, long[] dongleTime, long[] deadTimes, int deadTimesNumbers, byte
[] hash)
```

grdHandle [in]

Тип: [Handle](#)

Нэндл, через который будет выполнена данная операция.

algoNum_GSII64 [in]

Тип: int

Номер аппаратного алгоритма типа **GSII64**, который будет использоваться для расшифровывания числа-вопроса.

algoNum_HashS3 [in]

Тип: int

Номер аппаратного алгоритма **Hash64**, который будет использоваться для проверки подлинности числа-вопроса на основании MAC.

question [in]

Тип: byte []

Буфер, содержащий присланное удаленным пользователем число-вопрос.

id [in]

Тип: int

ID ключа удаленного пользователя, для которого будет произведена операция.

publicCode [in]

Тип: int

Численное значение Public Code ключа удаленного пользователя, для которого будет произведена операция.

dongleTime [in]

Тип: long []

Зашифрованное значение времени ключа.

deadTimes [in]

Тип: long []

Зашифрованное значение времен жизни алгоритмов.

deadTimesNumbers [in]

Тип: int

Количество элементов, возвращаемых в deadTimes.

hash [in]

Тип: byte []

Буфер, содержащий MAC, вычисленный на ключе удаленного пользователя.

pQuestion	после выполнения функции в этот буфер возвращается расшифрованное число-вопрос. Длина буфера 8 байт
-----------	---

Возможные ошибки

GrdE.SystemDataCorrupted	Системные данные TRU повреждены. (Секретный ключ удаленного программирования отсутствует)
GrdE.NoQuestion	Число-вопрос не было сгенерировано или было регенерировано до записи числа ответа
GrdE.InvalidData	Неверный формат данных для удаленного программирования
GrdE.QuestionOK	Число-вопрос уже было сгенерировано, ключ ожидает данных для удаленного программирования
GrdE.UpdateNotComplete	Ошибка при записи данных удаленного программирования. Операция не была завершена
GrdE.InvalidHash	Неверное значение MAC (Message Authentication Code)
	Набор ошибок Guardant API

Использование метода **GrdTRU_DecryptQuestionTime** позволяет получить число-вопрос в расшифрованном виде и убедиться в том, что оно действительно было сгенерировано на ключе с ID равным *id* и Public Code равным *publicCode*. Метод является аналогом **GrdTRU_DecryptQuestion** для ключа Guardant Time и используется в случаях, когда необходимо продлевать время работы защищенного приложения.

Если число-вопрос расшифровано правильно и проверка подлинности прошла успешно, функция возвращает **GrdE.OK**.

Расшифрованное 8-байтовое число-вопрос помещается в тот же буфер *question*, в котором находилось зашифрованное число-вопрос. Расшифрованное число-вопрос необходимо для генерации ответа, поэтому его нужно сохранить для дальнейшего использования.

Расшифровка числа-вопроса производится аппаратным алгоритмом типа **GSII64** с номером, задаваемым **GrdAN.GSII64**. На момент расшифровки этот алгоритм должен быть создан в ключе, находящемся у разработчика. В качестве определителя должен использоваться секретный 128-битовый ключ, который был сгенерирован и прошит в ключ удаленного пользователя при предпродажной подготовке методом **GrdTRU_SetKey** для ключа с ID, равным *id*. При использовании GRDUTIL этот ключ берется из базы данных.

Проверка подлинности числа-вопроса производится аппаратным алгоритмом типа **Hash64** с номером **GrdAN.Hash64**. На момент проверки этот алгоритм должен быть создан в ключе, находящемся у разработчика. В качестве определителя должен использоваться секретный 128-битовый ключ, который был сгенерирован и прошит в ключ удаленного пользователя при предпродажной подготовке методом **GrdTRU_SetKey** для ключа с ID равным *id*. При использовании GRDUTIL этот ключ берется из базы данных.

Рабочий ключ, находящийся у разработчика, инициализировать методом **GrdTRU_SetKey** с секретным ключом, таким же как у удаленного пользователя, необязательно. Все преобразования делаются на заранее запрограммированных алгоритмах, номера которых указываются в параметрах указываемых в параметрах **GrdAN.GSII64** и **GrdAN.Hash64**.

После расшифровки в *dongleTime* содержится расшифрованное время ключа из микросхемы таймера, а в *deadTimes* расшифрованное время деактивации алгоритмов.

Два старших байта параметра *dongleTime* - нули. 2 старших байта элементов массива *deadTimes* - числовое имя алгоритма. Остальные 6 байт - время жизни соответствующего алгоритма.

Время жизни (8 байт) имеет следующий формат:

1, 2 байты	числовое имя алгоритма (либо нули, если время из микросхемы таймера);
3 байт	год от 2000 (08 для 2008 года);
4 байт	месяц года (01 - январь, 02 - февраль, ...);
5 байт	день месяца (1 - 31);
6 байт	часы (0 - 23);
7 байт	минуты (0 - 59);
8 байт	секунды (0 - 59)