GrdCrypt

Функция(метод) **GrdCrypt** зашифровывает или расшифровывает блок данных при помощи аппаратного или программно-реализованного алгоритма.

C

```
int GRD_API GrdCrypt(
  HANDLE hGrd,
  DWORD dwAlgo,
  DWORD dwDataLng,
  void *pData,
  DWORD dwMethod,
  void *pIV,
  void *pKeyBuf,
  void *pContext
);
```

<table>
<tr><td>hGrd</td><td>хэндл, через который будет выполнена данная операция.</td></tr>
<tr><td>dwAlgo</td><td>номер аппаратного (GSII64) или программно-реализованного алгоритма (AES), которым будет производиться преобразование. Программно-реализованный алгоритм AES 256 имеет номер GrdSC_AES256.</td></tr>
<tr><td>dwDataLng</td><td>длина блока данных в байтах</td></tr>
<tr><td>pData</td><td>буфер данных для преобразования</td></tr>
<tr><td>dwMethod</td><td>метод преобразования, который задается комбинацией флагов GrdAM_XXX и GrdSC_XXX
<table>
<tr><td colspan="2">Биты 0-5 - режим работы алгоритма</td></tr>
<tr><td>GrdAM_ECB</td><td>Режим электронной кодовой книги</td></tr>
<tr><td>GrdAM_CBC</td><td>Режим сцепления кодированных блоков</td></tr>
<tr><td>GrdAM_CFB</td><td>Режим с кодированной обратной связью</td></tr>
<tr><td>GrdAM_OFB</td><td>Режим с обратной связью по выходу</td></tr>
<tr><td colspan="2">Бит 6 - резерв</td></tr>
<tr><td colspan="2">Бит 7 - тип операции</td></tr>
<tr><td>GrdAM_Encrypt</td><td>Кодировать блок</td></tr>
<tr><td>GrdAM_Decrypt</td><td>Декодировать блок</td></tr>
<tr><td colspan="2">Биты 8-9: тип блока данных</td></tr>
<tr><td>GrdSC_First</td><td>Первый блок данных</td></tr>
<tr><td>GrdSC_Next</td><td>Следующий блок данных</td></tr>
<tr><td>GrdSC_Last</td><td>Последний блок данных</td></tr>
<tr><td>GrdSC_All</td><td>Единственный блок данных</td></tr>
</table></td></tr>
<tr><td>pIV</td><td>вектор инициализации:
для GSII64 - 8 байтов, для AES - 16 байтов.
Если в качестве указателя на вектор инициализации задан NULL, то преобразование пройдет корректно, при этом будет использован нулевой вектор</td></tr>
<tr><td>pKeyBuf</td><td>буфер для передачи ключа шифрования для программно-реализованного алгоритма (AES). Длина ключа 256 бит (32 байта). Только для программно-реализованных алгоритмов. При использовании аппаратного алгоритма параметр должен быть равен NULL</td></tr>
</table>

pContext	буфер для контекста при шифровании больших массивов данных, которые разбиваются на несколько блоков. Для контекста должна быть зарезервирована память размером **GrdXXXXXX_CONTEXT_SIZE** байт в зависимости от алгоритма. Только для программно-реализованных алгоритмов. При использовании аппаратного алгоритма параметр должен быть равен **NULL** GrdAES256_KEY_SIZE — Длина ключа AES - 256 бит GrdAES256_BLOCK_SIZE — Длина блока данных AES - 128 бит GrdAES_CONTEXT_SIZE — Значение должно быть больше или равно sizeof(AES_CONTEXT)

Набор ошибок Guardant API

Функция **GrdCrypt** позволяет зашифровывать и расшифровывать данные с использованием аппаратно или программно-реализованных симметричных алгоритмов.

Преобразование производится алгоритмом с порядковым номером, заданным в параметре *dwAlgo*. В зависимости от номера алгоритма *dwAlgo* функция определяет, каким образом реализован алгоритм - аппаратным или программным. Если номер алгоритма соответствует аппаратному алгоритму, вызов переадресуется функции GrdTransform**.** Аппаратно реализованный алгоритм должен быть типа **GSII64.** Иначе, если алгоритм реализован программно (*dwAlgo***>=GrdSA_SoftAlgo**), вызывается соответствующая функция программного шифрования.

Программно-реализованные алгоритмы шифрования при шифровании больших массивов данных используют контекст, память для которого размером не менее **GrdXXXXXX_CONTEXT_SIZE** должна быть зарезервирована до вызова функции. Указатель на буфер для контекста передается через параметр *pContext*.

Длина шифруемых блоков данных зависит от метода шифрования (см. описание методов в *Руководство пользователя, часть 2*). Для методов CFB и OFB длина шифруемых блоков может быть произвольной.

Если шифрование на аппаратно-реализованном алгоритме выполнялось блоками произвольной длины (т. е. использовались флаги **GrdSC_First**, **GrdSC_Next**, **GrdSC_Last**), то для корректного расшифрования длина и порядок обработки блоков должны сохраняться. Для программно-реализованных алгоритмов такое ограничение отсутствует.

Если в дескрипторе аппаратного алгоритма установлен флаг **nsafl_GP_dec** (уменьшение счетчика), вычитание счетчика GP происходит при каждом вызове **GrdCrypt**.

C#

```
public static GrdE GrdCrypt(Handle grdHandle,GrdAlgNum algNum, byte[] data, GrdAM method, ref long iv)
public static GrdE GrdCrypt(Handle grdHandle,GrdAlgNum algNum, byte[] data, GrdAM method, byte[] iv)
public static GrdE GrdCrypt(Handle grdHandle,GrdAlgNum algNum, byte[] data, GrdAM method, byte[] iv, byte[] key)
public static GrdE GrdCrypt(Handle grdHandle, GrdAlgNum algNum, byte[] data, GrdAM method, byte[] iv, byte[]
key, byte[] context)
```

grdHandle [in]

Тип: Handle

хэндл, через который будет выполнена данная операция.

algNum [in]

Тип: GrdAlgNum

Номер аппаратного алгоритма, с помощью которого будет происходить преобразование.

data [in]

Тип: byte [ ]

Буфер данных для преобразования.

method [in]

Тип: GrdAM

Метод преобразования. Задается комбинацией флагов GrdAM.

iv [in]

Типы: byte [ ], long [ ]

Вектор инициализации.

key [in]

Тип: byte [ ]

Буфер для передачи ключа шифрования для программно-реализованного алгоритма AES.

context [in]

Тип: byte [ ]

Буфер для контекста при шифровании больших массивов данных. В этом случае они разбиваются на несколько блоков.
Набор ошибок Guardant API
Метод **GrdCrypt** позволяет зашифровывать и расшифровывать данные с использованием аппаратно или программно-реализованных симметричных алгоритмов.

Преобразование производится алгоритмом с порядковым номером, заданным в параметре *algNum*. В зависимости от номера алгоритма *algNum* метод определяет, каким образом реализован алгоритм - аппаратным или программным. Если номер алгоритма соответствует аппаратному алгоритму, вызов переадресуется методу GrdTransform. Аппаратно реализованный алгоритм должен быть типа GSII64. Иначе, если алгоритм реализован программно (*algNum*>=GrdSA_SoftAlgo), вызывается соответствующий метод программного шифрования.

Программно-реализованные алгоритмы шифрования при шифровании больших массивов данных используют контекст, память для которого размером не менее **GrdXXXXXX_CONTEXT_SIZE** должна быть зарезервирована до вызова метода. Указатель на буфер для контекста передается через параметр *context*.

Длина шифруемых блоков данных зависит от метода шифрования (см. описание методов в *Руководство пользователя, часть 2*). Для методов CFB и OFB длина шифруемых блоков может быть произвольной.

Если шифрование на аппаратно-реализованном алгоритме выполнялось блоками произвольной длины (т. е. использовались флаги **GrdSC_First**, **GrdSC_Next**, **GrdSC_Last**), то для корректного расшифрования длина и порядок обработки блоков должны сохраняться. Для программно-реализованных алгоритмов такое ограничение отсутствует.

Если в дескрипторе аппаратного алгоритма установлен флаг **nsafl_GP_dec** (уменьшение счетчика), вычитание счетчика GP происходит при каждом вызове **GrdCrypt**.

Java

```
public static GrdE GrdCrypt(Handle grdHandle, int algoNum, byte[] data, int method, byte[] iv)
public static GrdE GrdCrypt(Handle grdHandle, int algoNum, byte[] data, int method, byte[] iv, byte[] key)
public static GrdE GrdCrypt(Handle grdHandle, int algoNum, byte[] data, int method, byte[] iv, byte[] key, byte
[] context)
public static GrdE GrdCrypt( Handle grdHandle, int algoNum, byte[] data, int method, long iv)
```

grdHandle [in]

Тип: Handle

хэндл, через который будет выполнена данная операция.

algNum [in]

Тип: int

Номер аппаратного алгоритма, с помощью которого будет происходить преобразование.

data [in]

Тип: byte [ ]

Буфер данных для преобразования.

method [in]

Тип: int

Метод преобразования. Задается комбинацией флагов GrdAM.

iv [in]

Типы: byte [ ], long [ ]

Вектор инициализации.

key [in]

Тип: byte [ ]

Буфер для передачи ключа шифрования для программно-реализованного алгоритма AES.

context [in]

Тип: byte [ ]

Буфер для контекста при шифровании больших массивов данных. В этом случае они разбиваются на несколько блоков.
Набор ошибок Guardant API
Метод **GrdCrypt** позволяет зашифровывать и расшифровывать данные с использованием аппаратно или программно-реализованных симметричных алгоритмов.

Преобразование производится алгоритмом с порядковым номером, заданным в параметре *algNum*. В зависимости от номера алгоритма *algNum* метод определяет, каким образом реализован алгоритм - аппаратным или программным. Если номер алгоритма соответствует аппаратному алгоритму, вызов переадресуется методу GrdTransform. Аппаратно реализованный алгоритм должен быть типа GSII64. Иначе, если алгоритм реализован программно (*algNum*>=GrdSA_SoftAlgo), вызывается соответствующий метод программного шифрования.

Программно-реализованные алгоритмы шифрования при шифровании больших массивов данных используют контекст, память для которого размером не менее **GrdXXXXXX_CONTEXT_SIZE** должна быть зарезервирована до вызова метода. Указатель на буфер для контекста передается через параметр *context*.

Длина шифруемых блоков данных зависит от метода шифрования (см. описание методов в *Руководство пользователя, часть 2*). Для методов CFB и OFB длина шифруемых блоков может быть произвольной.

Если шифрование на аппаратно-реализованном алгоритме выполнялось блоками произвольной длины (т. е. использовались флаги **GrdSC_First**, **GrdSC_Next**, **GrdSC_Last**), то для корректного расшифрования длина и порядок обработки блоков должны сохраняться. Для программно-реализованных алгоритмов такое ограничение отсутствует.

Если в дескрипторе аппаратного алгоритма установлен флаг **nsafl_GP_dec** (уменьшение счетчика), вычитание счетчика GP происходит при каждом вызове **GrdCrypt**.