GrdTransformEx

Функция(метод) **GrdTransformEx** шифрует или расшифровывает блок данных при помощи аппаратно-реализованного криптографического алгоритма (GSII64 или AES128).

Синтаксис

C

```
int GRD_API GrdTransformEx(
  HANDLE hGrd,
  DWORD dwAlgoNum,
  DWORD dwLng,
  void *pData,
  DWORD dwMethod,
  DWORD dwIVLng,
  void *pIV,
  void pReserved
);
```

hGrd	хэндл, через который будет выполнена данная операция
dwAlgoNum	номер дескриптора алгоритма с помощью которого будет проводиться шифрование или расшифрование
dwLng	длина буфера данных в байтах
pData	указатель на буфер данных открытого или зашифрованного текста
dwMethod	следующие флаги позволяют указать режим работы алгоритма и тип операции:

Биты 0-5 - режим работы алгоритма	
GrdAM_ECB	Режим электронной кодовой книги (режим простой замены). Каждый блок открытого текста заменяется блоком шифротекста. Шифрование двух одинаковых блоков даст идентичный результат. Скорость обработки блоков в режиме ECB фиксирована. Недостаток ECB, в сравнении с другими режимами шифрования, — сохранение статистических особенностей открытого текста.
GrdAM_CBC	Режим сцепления блоков шифротекста. Каждый блок открытого текста (кроме первого) побитово складывается по модулю 2 (операция XOR) с предыдущим результатом шифрования. Таким образом, каждый блок зашифрованного текста зависит от всех блоков открытого текста, обработанных до него. Режим CBC лишён недостатка алгоритма ECB, но всё же имеет ряд недостатков с точки зрения безопасности.
GrdAM_CFB	Режим обратной связи по шифротексту (режим гаммирования с обратной связью). Для шифрования следующего блока открытого текста он складывается по модулю 2 с перешифрованным (блочным шифром) результатом шифрования предыдущего блока. Криптостойкость CFB определяется криптостойкостью используемого шифра.
GrdAM_OFB	Режим обратной связи по выходу. В этом режиме открытый текст используются только для конечного сложения. Операции блочного шифра могут быть выполнены заранее, позволяя выполнить заключительное шифрование параллельно с открытым текстом.
Бит 6 - резерв	
Бит 7 - тип операции	
GrdAM_Encrypt	Зашифровать блок данных
GrdAM_Decrypt	Расшифровать блок данных

dwIVLng	длина вектора инициализации: для GSII64 - 8 байт, для AES128 и AES256 - 16 байт.
pIV	указатель на вектор инициализации для режимов шифрования CBC, CFB и OFB. Для режима ECB данный параметр игнорируется
Reserved	не используется. Параметр должен быть равен **NULL**

GrdE_OK	Операция выполнена успешно.
GrdE_AlgoNotFound	Алгоритма с указанным номером не существует.

GrdE_InvalidArg	Задано недопустимое значение одного из аргументов функции.
GrdE_NoService	Для алгоритма/ячейки сервис не предусмотрен.
GrdE_GPis0	Счетчик алгоритма достиг нулевого значения.
GrdE_InactiveItem	Алгоритм/ячейка находятся в неактивном состоянии, команда не выполнена.
Перейти к списку всех ошибок Guardant API	

Функция **GrdTransformEx** позволяет зашифровать или расшифровать данные с помощью аппаратно-реализованного в электронном ключе криптографического алгоритма. Шифрование выполняется алгоритмом с номером, заданным в параметре *dwAlgoNum*. Этот алгоритм предварительно должен быть создан, в противном случае возвращается ошибка GrdE_AlgoNotFound. Если в дескрипторе алгоритма установлен флаг "уменьшение счетчика", то вычитание счетчика алгоритма происходит при каждом вызове **GrdTransformEx**. При достижении счетчиком нулевого значения, возвращается ошибка GrdE_GPis0. Если при создании алгоритма или в процессе работы он был переведен в неактивное состояние, возвращается ошибка GrdE_InactiveItem.

В целях безопасности, в электронном ключе существует возможность создавать алгоритмы позволяющие только зашифровывать или только расшифровывать данные. Для таких алгоритмов тип операции, передаваемый в параметре *dwMethod,* должен соответствовать типу алгоритма внутри электронного ключа, в противном случае возвращается ошибка GrdE_NoService.

Длина буфера данных задаётся параметром *dwLng*. Для режимов ECB и CBC она должна быть кратна размеру блока алгоритма шифрования, в противном случае возвращается ошибка GrdE_InvalidArg. Для режимов CFB и OFB длина может быть произвольной, т.к. в этих режимах открытый или зашифрованный текст складываются по модулю 2 с ключевыми блоками полученными в результате работы операции блочного шифра. Открытый или зашифрованный текст должны находиться по адресу, указанному в параметре *pData*. Если функция выполнена успешно, то по этому адресу будут размещены зашифрованные или расшифрованные данные той же длины. В этом случае функция возвращает GrdE_OK.

В таких режимах шифрования, как CBC, CFB и OFB функции необходимо передать указатель на вектор инициализации *pIV* и длину вектора инициализации в параметре *dwIVLng*. После выполнения операции буфер, на который указывает *pIV*, будет содержать значение необходимое функции для повторного вызова. Таким образом, начальное значение вектора инициализации будет изменено. Длина вектора инициализации должна быть равна размеру блока алгоритма шифрования. Если длина окажется больше указанной величины, то лишние байты будут проигнорированы. Если длина окажется меньше указанной величины, то исходный буфер будет дополнен нулями. При вызове функции с нулевым указателем на вектор инициализации, ситуация равноценна использованию вектора инициализации состоящего из нулевых значений или вектора инициализации нулевой длины.

Если шифрование выполнялось блоками произвольной длины, то для корректного расшифрования длина и порядок обработки блоков должны сохраняться. При выполнении операций шифрования и расшифрования следует использовать один и тот же вектор инициализации, если предполагается получить исходный открытый текст.

Скорость работы функции зависит от быстродействия электронного ключа и длины переданных данных. Следует заметить, что существует максимальная длина пакета данных передаваемых в ключ. Для достижения максимального быстродействия, длина передаваемых функции данных должна быть кратна этой величине. Если длина переданных данных превышает максимальную длину пакета, то поведение функции аналогично её многократному вызову.

Для алгоритмов GSII64

Размер блока алгоритма шифрования равен 8 байт. Для ключей **Stealth II** максимальная длина пакета 248 байт (**GrdAMRS_GSII64**), а для линейки моделей **Sign** 16384 байт.

Для алгоритмов AES128

Размер блока алгоритма шифрования равен 16 байт. Максимальная длина пакета 16384 байт.

Для алгоритмов ECC160

Для работы с аппаратно-реализованными алгоритмами ECC160 используйте функцию GrdSign.

Для алгоритмов HASH64

Для работы с аппаратно-реализованными алгоритмами HASH64 используйте функцию GrdHashEx.

C#

```
public static GrdE GrdTransformEx(Handle grdHandle, GrdAlgNum algNum, byte[] data, GrdAM method, byte[] iv)
```

grdHandle [in]

Тип: Handle

хэндл, через который будет выполнена данная операция.

algNum [in]

Тип: GrdAlgNum , GrdAN

Номер дескриптора алгоритма, которым будет проводиться шифрование или расшифрование. В примерах используются объекты класса GrdAN для передачи номеров демонстрационных алгоритмов.

data [in,out]

Тип: byte [ ]

Массив данных открытого или зашифрованного текста.

method [in]

Тип: GrdAM

Для одновременной передачи методу режима работы алгоритма и типа операции шифрования, класс GrdAM содержит перегрузки оператора сложения. Например, чтобы зашифровать блок данных в режиме электронной кодовой книги, методу достаточно передать GrdAM.ECB + GrdAM.Encrypt.

iv [in,out]

Тип: byte [ ]

Вектор инициализации для режимов шифрования CBC, CFB и OFB:

для GSII64 - 8 байт, для AES128 и AES256 - 16 байт.

GrdE.OK	Операция выполнена успешно.
GrdE.AlgoNotFound	Алгоритма с указанным номером не существует.
GrdE.InvalidArg	Задано недопустимое значение одного из аргументов функции.
GrdE.NoService	Для алгоритма/ячейки сервис не предусмотрен.
GrdE.GPis0	Счетчик алгоритма достиг нулевого значения.
GrdE.InactiveItem	Алгоритм/ячейка находятся в неактивном состоянии, команда не выполнена.
Перейти к списку всех ошибок Guardant API	

Метод **GrdTransformEx** позволяет зашифровать или расшифровать данные с помощью аппаратно-реализованного в электронном ключе криптографического алгоритма. Шифрование выполняется алгоритмом с номером, заданным в параметре *algNum*. Этот алгоритм предварительно должен быть создан, в противном случае возвращается ошибка GrdE.AlgoNotFound. Если в дескрипторе алгоритма установлен флаг "уменьшение счетчика", то вычитание счетчика алгоритма происходит при каждом вызове **GrdTransformEx**. При достижении счетчиком нулевого значения, возвращается ошибка GrdE.GPis0. Если при создании алгоритма или в процессе работы он был переведен в неактивное состояние, возвращается ошибка GrdE.InactiveItem.

В целях безопасности, в электронном ключе существует возможность создавать алгоритмы позволяющие только зашифровывать или только расшифровывать данные. Для таких алгоритмов тип операции, передаваемый в параметре *method,* должен соответствовать типу алгоритма внутри электронного ключа, в противном случае возвращается ошибка GrdE.NoService.

Для режимов ECB и CBC длина массива данных (в байтах) должна быть кратна размеру блока алгоритма шифрования, в противном случае возвращается ошибка GrdE.InvalidArg. Для режимов CFB и OFB длина может быть произвольной, т.к. в этих режимах открытый или зашифрованный текст складываются по модулю 2 с ключевыми блоками полученными в результате работы операции блочного шифра. Открытый или зашифрованный текст должны находиться в массиве передаваемом по ссылке в параметре *data*. Если метод выполнен успешно, то в этом же массиве будут размещены зашифрованные или расшифрованные данные той же длины. В этом случае метод возвращает GrdE.OK.

В таких режимах шифрования, как CBC, CFB и OFB методу необходимо передать ссылку на массив байт содержащий вектор инициализации (параметр *iv*). После выполнения операции данный массив будет содержать значения необходимые методу для повторного вызова. Таким образом, начальное значение вектора инициализации будет изменено. Длина вектора инициализации должна быть равна размеру блока алгоритма шифрования. Если длина окажется больше указанной величины, то лишние байты будут проигнорированы. Если длина окажется меньше указанной величины, то исходный массив будет дополнен нулями.

Если шифрование выполнялось блоками произвольной длины, то для корректного расшифрования длина и порядок обработки блоков должны сохраняться. При выполнении операций шифрования и расшифрования следует использовать один и тот же вектор инициализации, если предполагается получить исходный открытый текст.

Скорость работы метода зависит от быстродействия электронного ключа и длины переданных данных. Следует заметить, что существует максимальная длина пакета данных передаваемых в ключ. Для достижения максимального быстродействия, длина передаваемых методом данных должна быть кратна этой величине. Если длина переданных данных превышает максимальную длину пакета, то поведение метода аналогично её многократному вызову.

Для алгоритмов GSII64

Размер блока алгоритма шифрования равен 8 байт. Максимальная длина пакета 248 байт.

Для алгоритмов AES128

Размер блока алгоритма шифрования равен 16 байт. Максимальная длина пакета 16384 байт.

Для алгоритмов ECC160

Для работы с аппаратно-реализованными алгоритмами ECC160 используйте метод GrdSign.

Для алгоритмов HASH64

Для работы с аппаратно-реализованными алгоритмами HASH64 используйте метод GrdHashEx.

Java

```
public static GrdE GrdTransformEx(Handle grdHandle, int algoNum, byte[] data, GrdAM method, byte[] iv)
```

grdHandle [in]

Тип: Handle

хэндл, через который будет выполнена данная операция.

algNum [in]

Тип: int

Номер дескриптора алгоритма, которым будет проводиться шифрование или расшифрование. В примерах используются объекты класса GrdAN для передачи номеров демонстрационных алгоритмов.

data [in,out]

Тип: byte [ ]

Массив данных открытого или зашифрованного текста.

method [in]

Тип: GrdAM

Режим работы алгоритма и тип операции.

iv [in,out]

Тип: byte [ ]

Вектор инициализации для режимов шифрования CBC, CFB и OFB:

для GSII64 - 8 байт, для AES128 и AES256 - 16 байт.

GrdE.OK	Операция выполнена успешно.
GrdE.AlgoNotFound	Алгоритма с указанным номером не существует.
GrdE.InvalidArg	Задано недопустимое значение одного из аргументов функции.
GrdE.NoService	Для алгоритма/ячейки сервис не предусмотрен.
GrdE.GPis0	Счетчик алгоритма достиг нулевого значения.
GrdE.InactiveItem	Алгоритм/ячейка находятся в неактивном состоянии, команда не выполнена.

Перейти к списку всех ошибок Guardant API

Метод **GrdTransformEx** позволяет зашифровать или расшифровать данные с помощью аппаратно-реализованного в электронном ключе криптографического алгоритма. Шифрование выполняется алгоритмом с номером, заданным в параметре *algNum*. Этот алгоритм предварительно должен быть создан, в противном случае возвращается ошибка GrdE.AlgoNotFound. Если в дескрипторе алгоритма установлен флаг "уменьшение счетчика", то вычитание счетчика алгоритма происходит при каждом вызове **GrdTransformEx**. При достижении счетчиком нулевого значения, возвращается ошибка GrdE.GPis0. Если при создании алгоритма или в процессе работы он был переведен в неактивное состояние, возвращается ошибка GrdE.InactiveItem.

В целях безопасности, в электронном ключе существует возможность создавать алгоритмы позволяющие только зашифровывать или только расшифровывать данные. Для таких алгоритмов тип операции, передаваемый в параметре *method,* должен соответствовать типу алгоритма внутри электронного ключа, в противном случае возвращается ошибка GrdE.NoService.

Для режимов ECB и CBC длина массива данных (в байтах) должна быть кратна размеру блока алгоритма шифрования, в противном случае возвращается ошибка GrdE.InvalidArg. Для режимов CFB и OFB длина может быть произвольной, т.к. в этих режимах открытый или зашифрованный текст складываются по модулю 2 с ключевыми блоками полученными в результате работы операции блочного шифра. Открытый или зашифрованный текст должны находиться в массиве передаваемом по ссылке в параметре *data*. Если метод выполнен успешно, то в этом же массиве будут размещены зашифрованные или расшифрованные данные той же длины. В этом случае метод возвращает GrdE.OK.

В таких режимах шифрования, как CBC, CFB и OFB методу необходимо передать ссылку на массив байт содержащий вектор инициализации (параметр *iv*). После выполнения операции данный массив будет содержать значения необходимые методу для повторного вызова. Таким образом, начальное значение вектора инициализации будет изменено. Длина вектора инициализации должна быть равна размеру блока алгоритма шифрования. Если длина окажется больше указанной величины, то лишние байты будут проигнорированы. Если длина окажется меньше указанной величины, то исходный массив будет дополнен нулями.

Если шифрование выполнялось блоками произвольной длины, то для корректного расшифрования длина и порядок обработки блоков должны сохраняться. При выполнении операций шифрования и расшифрования следует использовать один и тот же вектор инициализации, если предполагается получить исходный открытый текст.

Скорость работы метода зависит от быстродействия электронного ключа и длины переданных данных. Следует заметить, что существует максимальная длина пакета данных передаваемых в ключ. Для достижения максимального быстродействия, длина передаваемых методом данных должна быть кратна этой величине. Если длина переданных данных превышает максимальную длину пакета, то поведение метода аналогично её многократному вызову.

Для алгоритмов GSII64

Размер блока алгоритма шифрования равен 8 байт. Максимальная длина пакета 248 байт.

Для алгоритмов AES128

Размер блока алгоритма шифрования равен 16 байт. Максимальная длина пакета 16384 байт.

Для алгоритмов ECC160

Для работы с аппаратно-реализованными алгоритмами ECC160 используйте метод GrdSign.

Для алгоритмов HASH64

Для работы с аппаратно-реализованными алгоритмами HASH64 используйте метод GrdHashEx.