Криптографический интерфейс ViPNet CNG 4.2

Руководство разработчика

infotecs

ФРКЕ.00106-04 33 03

ОАО «ИнфоТеКС»

127287, г. Москва, Старый Петровско-Разумовский проезд, дом 1/23, строение 1

Тел: (495) 737-61-96 (горячая линия), 737-61-92, факс 737-72-78

Сайт компании «ИнфоТеКС»: http://www.infotecs.ru

Электронный адрес службы поддержки: hotline@infotecs.ru

Содержание

Введение

О документе

Документ содержит описание идентификаторов и функций реализации интерфейса ViPNet CNG, предназначенной для встраивания криптографических операций в стороннее программное обеспечение.

Для кого предназначен документ

Документ предназначен для разработчиков программного обеспечения в области криптографического преобразования данных.

Соглашения документа

Ниже перечислены соглашения, принятые в этом документе для выделения информации.

Таблица 1. Обозначения, используемые в примечаниях

Обозначение	Описание
	**Внимание!** Указывает на обязательное для исполнения или следования действие или информацию.
	**Примечание.** Указывает на необязательное, но желательное для исполнения или следования действие или информацию.
	**Совет.** Содержит дополнительную информацию общего характера.

Таблица 2. Обозначения, используемые для выделения информации в тексте

Обозначение	Описание
Название	Название элемента интерфейса. Например, заголовок окна, название поля, кнопки или клавиши.
Клавиша+Клавиша	Сочетание клавиш. Чтобы использовать сочетание клавиш, следует нажать первую клавишу и, не отпуская ее, нажать вторую клавишу.
Меню > Подменю > Команда	Иерархическая последовательность элементов. Например, пункты меню или разделы на панели навигации.
`Код`	Имя файла, путь, фрагмент текстового файла (кода) или команда, выполняемая из командной строки.

Об интерфейсе ViPNet CNG

Назначение, область применения

ViPNet CNG — один из криптографических интерфейсов, реализованных в продуктах ViPNet CSP и ViPNet CSP Linux.

Реализация криптографического интерфейса ViPNet CNG создана в соответствии с криптографическим интерфейсом компании Microsoft — Cryptography API: Next Generation (CNG), пришедшим на смену интерфейсу CryptoAPI. Интерфейс CNG от компании Microsoft позволяет разработчикам использовать криптографические средства в собственных алгоритмах. Описание стандарта Cryptography API: Next Generation вы можете найти на сайте MSDN (http://msdn.microsoft.com/en-us/library/windows/desktop/aa376210(v=vs.85).aspx). Реализация интерфейса ViPNet CNG разработана для встраивания в прикладные программы российских алгоритмов шифрования и электронной подписи.

Ниже приведены особенности криптографического интерфейса CNG:

- Быстрота выполнения криптографических функций.
- Поддержка алгоритмов, созданных с использованием интерфейса CryptoAPI.
- Поддержка криптографических функций в режиме ядра.
- Возможность использования сторонних генераторов случайных чисел.
- Потокобезопасность.

Список алгоритмов, добавленных для поддержки ГОСТ

Ниже приведен список алгоритмов, добавленных для поддержки ГОСТ в CNG-функциях и структурах. Указанные значения используются в функциях `BCryptOpenAlgorithmProvider` для идентификации алгоритм-провайдера.

Таблица 3. Алгоритмы, добавленные для поддержки ГОСТ

Идентификатор и значение	Описание
Алгоритмы, поддерживающие интерфейс хэширования	
`BCRYPT_ITCS_HASH_94_ALGID` `L"GOST R 34.11-94"`	Алгоритм хэширования: ГОСТ. Стандарт: ГОСТ 34.11-94. Если в `BCryptOpenAlgorithmProvider` выставлен флаг `BCRYPT_ALG_HANDLE_HMAC_FLAG`, это значит, что используется алгоритм HMAC (RFC 2104) с функцией хэширования ГОСТ Р 34.11-94.

Идентификатор и значение	Описание
`BCRYPT_ITCS_HASH_2012_256_ALGID` `L"GOST R 34.11-2012/256"`	Алгоритм хэширования: ГОСТ. Стандарт: ГОСТ 34.11-2012 с длиной хэш-кода 256 бит. Если в `BCryptOpenAlgorithmProvider` выставлен флаг `BCRYPT_ALG_HANDLE_HMAC_FLAG`, это значит, что используется алгоритм HMAC (RFC 2104) с функцией хэширования ГОСТ 34.11-2012 с длиной хэш-кода 256 бит.
`BCRYPT_ITCS_HASH_2012_512_ALGID` `L"GOST R 34.11-2012/512"`	Алгоритм хэширования: ГОСТ. Стандарт: ГОСТ 34.11-2012 с длиной хэш-кода 512 бит. Если в `BCryptOpenAlgorithmProvider` выставлен флаг `BCRYPT_ALG_HANDLE_HMAC_FLAG`, это значит, что используется алгоритм HMAC (RFC 2104) с функцией хэширования ГОСТ 34.11-2012 с длиной хэш-кода 512 бит.
`BCRYPT_ITCS_MAC_89_ALGID` `L"GOST 28147-89 MAC"`	Симметричное шифрование по алгоритму ГОСТ 28147-89 в режиме выработки имитовставки.
Алгоритм, поддерживающий интерфейс шифрования	
`BCRYPT_ITCS_ENCRYPT_89_ALGID` `L"GOST 28147-89"`	Симметричное шифрование по алгоритму ГОСТ 28147-89.
Алгоритмы, поддерживающие интерфейс подписи	
`BCRYPT_ITCS_SIGN_2001_ALGID` `L"GOST R 34.10-2001"`	Алгоритм электронной подписи ГОСТ от 2001 года. Стандарт: ГОСТ 34.10-2001.
`BCRYPT_ITCS_SIGN_2012_256_ALGID` `L"GOST R 34.10-2012/512"`	Алгоритм электронной подписи ГОСТ от 2012 года. Стандарт: ГОСТ 34.10-2012 с длиной хэш-кода 256 бит.
`BCRYPT_ITCS_SIGN_2012_512_ALGID` `L"GOST R 34.10-2012/1024"`	Алгоритм электронной подписи ГОСТ от 2012 года. Стандарт: ГОСТ 34.10-2012 с длиной хэш-кода 512 бит.
Алгоритмы, поддерживающие интерфейс генерации случайных чисел	
`BCRYPT_ITCS_RNG_ALGID` `L"ITCS-RNG-ALGORITHM"`	Алгоритм можно использовать для генерации псевдослучайной последовательности в пользовательском ПО (за исключением генерации значений ключей). Для генерации ключей необходимо использовать соответствующую функцию: `BCryptGenerateKeyPair` или `BCryptGenerateSymmetricKey`.

Обратная связь

Дополнительная информация

Сведения о продуктах и решениях ViPNet, распространенные вопросы и другая полезная информация собраны на сайте ОАО «ИнфоТеКС»:

- Веб-портал документации ViPNet http://docs.infotecs.ru.
- Описание продуктов ViPNet http://www.infotecs.ru/products/line/.
- Информация о решениях ViPNet http://www.infotecs.ru/solutions/.
- Сборник часто задаваемых вопросов (FAQ) http://www.infotecs.ru/support/faq/.
- Форум пользователей продуктов ViPNet http://www.infotecs.ru/forum.

Контактная информация

С вопросами по использованию продуктов ViPNet, пожеланиями или предложениями свяжитесь со специалистами ОАО «ИнфоТеКС». Для решения возникающих проблем обратитесь в службу технической поддержки.

- Техническая поддержка для пользователей продуктов ViPNet: hotline@infotecs.ru.
- Форма запроса в службу технической поддержки http://www.infotecs.ru/support/request/.
- Консультации по телефону для клиентов, имеющих расширенный уровень технического сопровождения:

  8 (495) 737-6192,

  8 (800) 250-0260 — бесплатный звонок из любого региона России (кроме Москвы).

Распространение информации об уязвимостях продуктов ОАО «ИнфоТеКС» регулируется политикой ответственного разглашения http://infotecs.ru/products/disclosure.php. Если вы обнаружили уязвимости в продуктах компании, сообщите о них по адресу security-notifications@infotecs.ru.

1

Свойства и функции криптографических примитивов

Свойства криптографических примитивов

Приведенные ниже значения используются в функциях `BCryptGetProperty` и `BCryptSetProperty` для идентификации свойств. Если не указано иное, то свойство доступно как для чтения, так и для записи. Если не указано иное, то свойство имеет смысл как для алгоритм-провайдера, так и для связанных с ним CNG-объектов (например, объектов ключей и объектов хэша).

Примечание. Далее в тексте под термином «субобъект алгоритм-провайдера» понимается следующее:

- Для алгоритмов, поддерживающих интерфейс хэширования, — объект хэша.
- Для алгоритмов, поддерживающих интерфейс симметричного шифрования, — объект симметричного ключа.
- Для алгоритмов, поддерживающих интерфейс электронной подписи, — объект ключа, представляющий закрытый ключ (пару ключей), либо объект ключа, представляющий открытый ключ.

Таблица 4. Список стандартных свойств криптографических примитивов

Идентификатор и значение	Описание
`BCRYPT_ALGORITHM_NAME` `L"AlgorithmName"`	Оканчивающаяся нулем Юникод-строка, содержащая имя алгоритма. Свойство доступно только для чтения.
`BCRYPT_BLOCK_LENGTH` `L"BlockLength"`	Размер в байтах блока для шифрования по выбранному алгоритму. Это свойство применяется только в алгоритмах блочного шифрования. Тип данных — `DWORD`. Свойство доступно только для чтения.
`BCRYPT_CHAINING_MODE` `L"ChainingMode"`	Режим шифрования (chaining mode) для алгоритмов симметричного шифрования. Свойство есть как у алгоритм-провайдера симметричного шифрования, так и у объекта симметричного ключа. При создании на заданном алгоритм-провайдере объекты симметричных ключей по умолчанию будут иметь то же значение свойства. При установке этого свойства у объекта симметричного ключа состояние шифратора сбрасывается в начальное состояние. Возможны следующие значения: • `BCRYPT_CHAIN_MODE_CBC` (`L"ChainingModeCBC"`) — режим сцепления блоков шифротекста (англ. Cipher Block Chaining, CBC); • `BCRYPT_CHAIN_MODE_ECB` (`L"ChainingModeECB"`) —

Идентификатор и значение	Описание
	режим простой замены (или «режим электронной кодовой книги» (от англ. Electronic Codebook, ECB)); • `BCRYPT_CHAIN_MODE_GOST89_CNT` (`L"ChainingModeGost89CNT"`) — ГОСТ-режим счетчика. Для алгоритма шифрования ГОСТ 28147-89 соответствует режиму `CRYPT_MODE_OFB` в интерфейсе ViPNet CSP; • `BCRYPT_CHAIN_MODE_CFB` (`L"ChainingModeCFB"`) — режим гаммирования с обратной связью по шифротексту (от англ. Cipher Feedback Mode, CFB). Режим используется по умолчанию при симметричном шифровании.
`BCRYPT_ITCS_EXPORT_ID` `L"itcs_export_id"`	Идентификатор экспорта симметричного ключа. Выставляется у ключа защиты (параметр `hImportKey` в функции `BCryptImportKey` и параметр `hExportKey` в функции `BCryptExportKey`), на котором происходит экспорт/импорт другого симметричного ключа. Свойство есть как у алгоритм-провайдера симметричного шифрования, так и у объекта симметричного ключа. При создании на заданном алгоритм-провайдере объекты симметричных ключей по умолчанию будут иметь то же значение свойства. При установке этого свойства у объекта симметричного ключа состояние шифратора сбрасывается в начальное состояние. Возможны следующие значения: • `BCRYPT_ITCS_CRYPTO_PRO_EXPORT` (`L"itcs-crypto-pro-export"`) — указывает на то, что при экспорте/импорте ключа надо использовать алгоритм по RFC 4357 (п. 6.3), используется по умолчанию. • `BCRYPT_ITCS_CRYPTO_PRO12_EXPORT` (`L"itcs-crypto-pro12-export"`) — указывает на то, что при экспорте/импорте ключа надо использовать алгоритм по Р 50.1.113-2016. • `BCRYPT_ITCS_SIMPLE_EXPORT` (`L"itcs-simple-export"`) — указывает на то, что при экспорте/импорте ключа надо использовать алгоритм по RFC 4357 (п. 6.1). Применяется для хранения и конвертации ключей.
`BCRYPT_HASH_BLOCK_LENGTH` `L"HashBlockLength"`	Размер в байтах блока для хэширования. Это свойство применяется только в алгоритмах хэширования. Тип данных — `DWORD`. Свойство доступно только для чтения.
`BCRYPT_HASH_LENGTH` `L"HashDigestLength"`	Размер в байтах значения хэша от хэш-провайдера. Тип данных — `DWORD`. Свойство доступно только для чтения.

Идентификатор и значение	Описание
BCRYPT_IS_KEYED_HASH L"IsKeyedHash"	Указывает, что алгоритм, реализующий интерфейс хэширования, использует ключ (в функциях хэширования надо указывать ненулевой ключ). Тип данных — BOOL. Свойство доступно только для чтения. Свойство поддерживается только для таких алгоритмов, для которых поддерживается интерфейс хэширования.
BCRYPT_INITIALIZATION_VECTOR L"IV"	Содержит начальный вектор шифрования (IV) для симметричного ключа. Длина IV определяется свойством BCRYPT_BLOCK_LENGTH. Данное свойство есть только у объекта симметричного ключа. Данное свойство поддерживается только для алгоритмов симметричного шифрования. При установке этого свойства у объекта симметричного ключа состояние шифратора сбрасывается в начальное состояние. Если устанавливается значение NULL, то значение IV сбрасывается. Если данное свойство у объекта симметричного ключа не было задано явно или было сброшено, то начальный вектор генерируется случайным образом и сохраняется в объекте симметричного ключа в следующих функциях: • BCryptEncrypt; • BCryptDecrypt; • BCryptExportKey с типом экспорта BCRYPT_OPAQUE_KEY_BLOB; • BCryptGetProperty с параметром BCRYPT_INITIALIZATION_VECTOR.
BCRYPT_ITCS_HASH_OID L"itcs-hash-oid"	OID-идентификатор параметров хэширования. Данное свойство поддерживается только для алгоритма BCRYPT_ITCS_HASH_94_ALGID. При создании на заданном алгоритм-провайдере объекты хэширования по умолчанию будут иметь то же значение свойства. Поддерживаются следующие значения этого свойства: • BCRYPT_ITCS_HASH_DEF_PARAM (L"1.2.643.2.2.30.1") — параметры хэширования с OID id-GostR3411-94-CryptoProParamSet. • BCRYPT_ITCS_HASH_TEST_PARAM (L"1.2.643.2.2.30.0") — параметры хэширования с OID id-GostR3411-94-TestParamSet.
BCRYPT_ITCS_HASH_STATE L"itcs-hash-state"	Сериализованное состояние объекта хэширования. Данное свойство поддерживается только для алгоритма CSP_MAC_89_CNG_ALGID.

Идентификатор и значение	Описание
`BCRYPT_ITCS_CIPHER_OID` `L"itcs-cipher-oid"`	OID-идентификатор параметров шифрования. Свойство есть как у алгоритм-провайдера симметричного шифрования, так и у объекта симметричного ключа. При создании на заданном алгоритм-провайдере объекты симметричных ключей по умолчанию будут иметь то же значение свойства. При установке этого свойства у объекта симметричного ключа состояние шифратора сбрасывается в начальное состояние. Для алгоритма `BCRYPT_ITCS_ENCRYPT_89_ALGID` поддерживаются следующие значения этого свойства: • `BCRYPT_ITCS_CIPHER_DEF_PARAM` или `BCRYPT_ITCS_CIPHER_A_PARAM` (`L"1.2.643.2.2.31.1"`) — таблица узлов замены с OID `id-Gost28147-89-CryptoPro-A-ParamSet`. Это значение используется по умолчанию. • `BCRYPT_ITCS_CIPHER_B_PARAM` (`L"1.2.643.2.2.31.2"`) — таблица узлов замены с OID `id-Gost28147-89-CryptoPro-B-ParamSet`. • `BCRYPT_ITCS_CIPHER_C_PARAM` (`L"1.2.643.2.2.31.3"`) — таблица узлов замены с OID `id-Gost28147-89-CryptoPro-C-ParamSet`. • `BCRYPT_ITCS_CIPHER_D_PARAM` (`L"1.2.643.2.2.31.4"`) — таблица узлов замены с OID `id-Gost28147-89-CryptoPro-D-ParamSet`. • `BCRYPT_ITCS_CIPHER_TEST_PARAM` (`L"1.2.643.2.2.31.0"`) — таблица узлов замены с OID `id-Gost28147-89-TestParamSet`. • `BCRYPT_ITCS_CIPHER_ISO_GOST_28147` (`L"1.2.643.7.1 .2.5.1.1"`) — таблица узлов замены с OID `id-tc26-gost-28147-paramSetISO`.

Идентификатор и значение	Описание
`BCRYPT_ITCS_SIGNATURE_OID` `L"itcs-signature-oid"`	OID-идентификатор параметров электронной подписи. Свойство поддерживается только алгоритмами, для которых реализован интерфейс электронной подписи. Свойство есть как у алгоритм-провайдера, так и у объекта ключа. При создании на заданном алгоритм-провайдере объекты хэширования по умолчанию будут иметь то же значение свойства. Алгоритмы `BCRYPT_ITCS_SIGN_2001_ALGID` и `BCRYPT_ITCS_SIGN_2012_256_ALGID` поддерживают следующие значения этого свойства: • `BCRYPT_ITCS_SIGNATURE_256_DEF_PARAM` (`L"1.2.643.2.2.35.1"`) — параметры асимметричной схемы с OID `id-GostR3410-2001-CryptoPro-A-ParamSet`. Это значение используется по умолчанию. • `BCRYPT_ITCS_SIGNATURE_256_B_PARAM` (`L"1.2.643.2.2.35.2"`) — параметры асимметричной схемы с OID `id-GostR3410-2001-CryptoPro-B-ParamSet`. • `BCRYPT_ITCS_SIGNATURE_256_C_PARAM` (`L"1.2.643.2.2.35.3"`) — параметры асимметричной схемы с OID `id-GostR3410-2001-CryptoPro-C-ParamSet`. • `BCRYPT_ITCS_SIGNATURE_256_TEST_PARAM` (`L"1.2.643.2.2.35.3"`) — параметры асимметричной схемы с OID `id-GostR3410-2001-TestParamSet`. • `BCRYPT_ITCS_DH_256_DEF_PARAM` (`L"1.2.643.2.2.36.0"`) — то же самое, что и `BCRYPT_ITCS_SIGNATURE_256_DEF_PARAM`. • `BCRYPT_ITCS_DH_256_1_PARAM` (`L"1.2.643.2.2.36.1"`) — то же самое, что и `BCRYPT_ITCS_SIGNATURE_256_C_PARAM`. Алгоритм `BCRYPT_ITS_SIGN_2012_512_ALGID` поддерживают следующие значения этого свойства: • `BCRYPT_ITCS_SIGNATURE_512_DEF_PARAM` (`L"1.2.643.7.1.1.2.1"`) — параметры асимметричной схемы с OID `id-tc26-gost-3410-12-512-paramSetA`. Данное значение используется по умолчанию. • `BCRYPT_ITCS_SIGNATURE_512_B_PARAM` (`L"1.2.643.7.1.1.2.2"`) — параметры асимметричной схемы с OID `id-tc26-gost-3410-12-512-paramSetB`. • `BCRYPT_ITCS_SIGNATURE_512_TEST_PARAM` (`L"1.2.643.7.1.1.2.0"`) — параметры асимметричной схемы с OID `id-tc26-gost-3410-12-512-paramSetTest`.

Идентификатор и значение	Описание
BCRYPT_ITCS_PRIVATE_KEY_VALUE L"itcs-private-key-value"	Значение закрытого ключа. Свойство доступно только для записи и служит для выставления значения закрытого ключа в объект закрытого асимметричного ключа. Свойство есть только у объектов асимметричного ключа и выставляется, если для объекта ключа не вызвана функция BCryptFinalizeKeyPair. Если для объекта симметричного ключа указанная функция уже вызвана, то при установке этого свойства генерируется ошибка STATUS_INVALID_PARAMETER.
BCRYPT_ITCS_USE_MECHING_PERIOD L"itcs-use-meshing-period"	Указывает на необходимость использования мешинга ключа. Тип данных — BOOL. Значение по умолчанию — TRUE. Если выставлено это свойство, то значение симметричного ключа будет меняться через каждые 1024 байта данных, в противном случае ключ не изменяется. Свойство есть как у алгоритм-провайдера симметричного шифрования, так и у объекта симметричного ключа. При создании на заданном алгоритм-провайдере объекты симметричных ключей по умолчанию будут иметь то же значение свойства. При установке этого свойства у объекта симметричного ключа состояние шифратора сбрасывается в начальное состояние.
BCRYPT_KEY_LENGTH L"KeyLength"	Размер в битах значения ключа от провайдера симметричного ключа. Тип данных — DWORD. Свойство доступно только для чтения.
BCRYPT_KEY_LENGTHS L"KeyLengths"	Размеры ключей, поддерживаемые алгоритмом. Свойство имеет структуру BCRYPT_KEY_LENGTHS_STRUCT. Свойство доступно только для чтения.
BCRYPT_KEY_STRENGTH L"KeyStrength"	Для объекта симметричного ключа — размер ключа в битах; Для объекта закрытого асимметричного ключа — размер закрытого ключа в битах; для объекта открытого ключа — 0. Тип данных — DWORD. Свойство применимо только к ключам. Свойство доступно только для чтения.
BCRYPT_OBJECT_LENGTH L"ObjectLength"	Размер субобъекта провайдера в байтах. Тип данных — DWORD. Современные провайдеры алгоритмов хэширования и симметричного шифрования используют для хранения своих субобъектов буферы, выделяемые вызывающей стороной. Например, хэш-провайдер требует выделения памяти для объекта хэша, полученного функцией BCryptCreateHash. Свойство предоставляет размер буфера для объекта провайдера, и вы можете выделить память для объекта, созданного провайдером. Свойство доступно только для чтения.

Идентификатор и значение	Описание
BCRYPT_PROVIDER_HANDLE L"ProviderHandle"	Дескриптор алгоритм-провайдера, с которым связан данный объект. Тип данных — BCRYPT_ALG_HANDLE. Свойство доступно только для чтения.
BCRYPT_SIGNATURE_LENGTH L"SignatureLength"	Длина подписи в байтах. Тип данных — DWORD. Свойство поддерживается провайдерами, реализующими интерфейс электронной подписи и у соответствующих объектов закрытого и открытого ключей. Свойство доступно только для чтения.

Функции криптографических примитивов

В этом разделе описаны функции криптографических примитивов интерфейса ViPNet CNG BCrypt.

Список поддерживаемых функций:

- `BCryptCloseAlgorithmProvider`;
- `BCryptCreateHash`;
- `BCryptDestroyHash`;
- `BCryptDestroyKey`;
- `BCryptDuplicateHash`;
- `BCryptDuplicateKey`;
- `BCryptEncrypt`;
- `BCryptExportKey`;
- `BCryptFinalizeKeyPair`;
- `BCryptFinishHash`;
- `BCryptGenerateKeyPair`;
- `BCryptGenerateSymmetricKey`;
- `BCryptGenRandom`;
- `BCryptGetProperty`;
- `BCryptHashData`;
- `BCryptImportKey`;
- `BCryptImportKeyPair`;
- `BCryptOpenAlgorithmProvider`;
- `BCryptSetProperty`;
- `BCryptSignHash`;
- `BCryptVerifySignature`.

Описание указанных функций вы можете найти на сайте MSDN (http://msdn.microsoft.com/en-us/library/windows/desktop/aa376210(v=vs.85).aspx).

Для вызова функций ViPNet CNG BCrypt на уровне ядра используйте библиотеку `Cng.lib`, которая является частью Driver Development Kit (DDK). Для получения дополнительной информации см. WDK and Developer Tools http://msdn.microsoft.com/en-US/windows/hardware/gg454513. Вызовы функций ViPNet CNG BCrypt на уровне ядра в режиме `DISPATCH_LEVEL IRQL` не поддерживаются.

Функция BCryptCloseAlgorithmProvider

Функция `BCryptCloseAlgorithmProvider` закрывает алгоритм-провайдер.

```
NTSTATUS WINAPI BCryptCloseAlgorithmProvider(
    _inout BCRYPT_ALG_HANDLE hAlgorithm,
  _in    ULONG dwFlags
);
```

Параметры:

- `hAlgorithm [in, out]`. Дескриптор, представляющий закрываемый алгоритм-провайдер. Этот дескриптор необходимо вызывать заранее с помощью функции `BCryptOpenAlgorithmProvider`.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. На данный момент не используется и должен быть равен нулю.

Возвращает код состояния, показывающий, успешно ли была выполнена функция.

Таблица 5. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_INVALID_HANDLE`	Недопустимый дескриптор ключа в параметре `hAlgorithm`.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция `BCryptCloseAlgorithmProvider` может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция должна выполняться в `PASSIVE_LEVEL IRQL`.

Функция BCryptCreateHash

Функция `BCryptCreateHash` предназначена для создания объекта хэша или MAC-объекта.

```
NTSTATUS WINAPI BCryptCreateHash(
    _Inout_   BCRYPT_ALG_HANDLE hAlgorithm,
    _Out_     BCRYPT_HASH_HANDLE *phHash,
    _Out_     PUCHAR pbHashObject,
    _In_opt_  ULONG cbHashObject,
    _In_opt_  PUCHAR pbSecret,
    _In_      ULONG cbSecret,
    _In_      ULONG dwFlags
);
```

Параметры:

- `hAlgorithm [in, out]`. Дескриптор, представляющий алгоритм-провайдер. Этот дескриптор ранее должен был быть создан вызовом функции `BCryptOpenAlgorithmProvider` с идентификатором алгоритма, поддерживающим интерфейс хэширования.
- `phHash [out]`. Указатель на значение типа `BCRYPT_HASH_HANDLE`, куда будет записан указатель на созданный объект хэша или MAC-объект. Этот дескриптор используется при последующих вызовах функций хэширования или MAC-функций, таких как `BcryptHashData`. После окончания использования этот дескриптор должен быть освобождён вызовом функции `BCryptDestroyHash`.
- `pbHashObject [out]`. Указатель на буфер, куда будет записан созданный объект хэша или MAC-объект. Требуемый размер этого буфера определяется чтением свойства `BCRYPT_OBJECT_LENGTH` посредством вызова функции `BCryptGetProperty`, которая вернёт размер объекта хэша или MAC-объекта для заданного алгоритма.

  Эта память может быть освобождена только после уничтожения дескриптора, на который указывает `phHash`.

  Если данный параметр равен `NULL` и значение параметра `cbHashObject` равно 0, то память под создаваемый объект выделяется системой.

  Примечание. Данный способ выделения памяти поддерживается в ОС Windows, начиная с версии Vista SP1.

- `cbHashObject [in, optional]`. Размер буфера, на который указывает `pbHashObject`, в байтах. Если данный параметр равен 0 и значение параметра `pbHashObject` равно `NULL`, то память под создаваемый объект выделяется системой.

  Примечание. Данный способ выделения памяти поддерживается в ОС Windows, начиная с версии Vista SP1.

- `pbSecret [in, optional]`. Указатель на буфер, содержащий ключ, используемый при расчете хэш-функции или MAC. Если алгоритм не требует ключа, данный параметр должен быть равен `NULL`, а параметр `cbSecret` должен быть равен 0, в противном случае будет сгенерирована ошибка. Детальное описание данного параметра см. ниже.
- `cbSecret [in]`. Размер буфера `pbSecret` в байтах. Если ключ для расчета хэша не используется, то данный параметр должен быть равен нулю.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. На данный момент не используется и должен быть равен нулю.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 6. Возвращаемые значения

Возвращаемое значение	Описание
STATUS_SUCCESS	Функция выполнена успешно.
STATUS_BUFFER_TOO_SMALL	Размер, определенный в параметре cbHashObject, недостаточен для объекта хэша или MAC.
STATUS_INVALID_HANDLE	Недопустимый дескриптор алгоритм-провайдера в параметре hAlgorithm.
STATUS_INVALID_PARAMETER	Один или несколько недопустимых параметров.
STATUS_NOT_SUPPORTED	Алгоритм не поддерживает интерфейс хэширования

Алгоритмы реализации интерфейса ViPNet CNG BCrypt, поддерживающие интерфейс хэширования и работающие без ключа:

- BCRYPT_ITCS_HASH_94_ALGID без флага BCRYPT_ALG_HANDLE_HMAC_FLAG;
- BCRYPT_ITCS_HASH_2012_256_ALGID без флага BCRYPT_ALG_HANDLE_HMAC_FLAG;
- BCRYPT_ITCS_HASH_2012_512_ALGID без флага BCRYPT_ALG_HANDLE_HMAC_FLAG.

Алгоритмы реализации интерфейса ViPNet CNG BCrypt, поддерживающие интерфейс хэширования и требующие задания ключа:

- BCRYPT_ITCS_HASH_94_ALGID с установленным флагом BCRYPT_ALG_HANDLE_HMAC_FLAG;
- BCRYPT_ITCS_HASH_2012_256_ALGID с установленным флагом BCRYPT_ALG_HANDLE_HMAC_FLAG;
- BCRYPT_ITCS_HASH_2012_512_ALGID с установленным флагом BCRYPT_ALG_HANDLE_HMAC_FLAG;
- BCRYPT_ITCS_MAC_89_ALGID (значение флага игнорируется BCRYPT_ALG_HANDLE_HMAC_FLAG).

Все алгоритмы реализации интерфейса ViPNet CNG BCrypt, поддерживающие интерфейс хэширования и требующие задания ключа, могут содержать в pbSecret и cbSecret следующие значения:

- cbSecret равен 32, pbSecret указывает на буфер, содержащий значение симметричного ключа;
- cbSecret равен размеру объекта ключа для алгоритма BCRYPT_ITCS_ENCRYPT_89_ALGID, pbSecret равен значению дескриптора ключа для алгоритма BCRYPT_ITCS_ENCRYPT_89_ALGID.

Для алгоритма BCRYPT_ITCS_MAC_89_ALGID возможны еще два варианта:

- cbSecret равен размеру экспортированного состояния объекта MAC, получаемого через свойство BCRYPT_ITCS_HASH_STATE, pbSecret указывает на буфер, содержащий это экспортированное состояние — в этом случае будет создан объект с тем же ключом, но с начальным состоянием хэширования;

Примечание. Данный вариант реализован в текущей версии ViPNet CNG BCrypt, но может быть удален в будущих реализациях, поэтому не рекомендован к использованию

- `cbSecret` равен 0 и `pbSecret` равен `NULL` — в этом случае будет создан объект с ключом, состоящим из нулей.

Примечание. Использовать данный объект для расчета MAC нецелесообразно — рассматриваемый вариант был реализован, чтобы после создания объекта выставить состояние объекта через свойство `BCRYPT_ITCS_HASH_STATE`.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptDecrypt

Функция `BCryptDecrypt` предназначена для расшифрования блоков данных. В ViPNet CNG реализована только для симметричного шифрования `BCRYPT_ITCS_ENCRYPT_89_ALGID`.

```
NTSTATUS WINAPI BCryptDecrypt(
    __inout      BCRYPT_KEY_HANDLE hKey,
    __in         PUCHAR pbInput,
    __in         ULONG cbInput,
    __in_opt     VOID *pPaddingInfo,
    __inout_opt  PUCHAR pbIV,
    __in         ULONG cbIV,
    __out_opt    PUCHAR pbOutput,
    __in         ULONG cbOutput,
    __out        ULONG *pcbResult,
    __in         ULONG dwFlags
);
```

Параметры:

- `hKey [in, out]`. Дескриптор ключа для расшифрования данных. Может быть получен от одной из функций создания ключей, например `BCryptGenerateSymmetricKey`, `BCryptGenerateKeyPair` или `BCryptImportKey`.
- `pbInput [in]`. Адрес буфера с зашифрованным текстом, предназначенным для расшифрования. Параметр `cbInput` содержит размер этого текста.

- `cbInput [in]`. Размер в байтах содержимого буфера `pbInput`, предназначенного для расшифрования.
- `pPaddingInfo [in, optional]`. Указатель на структуру, содержащую информацию о паддинге. Данный параметр в реализации ViPNet CNG имеет смысл только в том случае, если выставлен флаг `BCRYPT_BLOCK_PADDING`, который является указателем на структуру типа `BCRYPT_ITCS_PADDING_INFO`, определяющую тип паддинга. Если этот параметр не задан, то производится паддинг по умолчанию. В случае если не выставлен флаг `BCRYPT_BLOCK_PADDING`, данный параметр должен иметь значение NULL.

  Если выставлен флаг `BCRYPT_BLOCK_PADDING`, то параметр `pPaddingInfo` определяет тип паддинга. В противном случае (если `pPaddingInfo` равен `NULL` и выставлен флаг `BCRYPT_BLOCK_PADDING`) паддинг считается паддингом по умолчанию (PKCS5-паддинг).

  Паддинг имеет смысл только для блочных режимов шифрования (CBC и ECB), поэтому для потоковых режимов шифрования данный флаг и поле `pPaddingInfo` игнорируются:

  ```
  typedef struct __BCRYPT_ITCS_PADDING_INFO
  {
  LPCWSTR  szPaddingType;
  } BCRYPT_ITCS_PADDING_INFO, *PBCRYPT_ITCS_PADDING_INFO;
  ```

  `szPaddingType` может принимать следующие значения:

  - `BCRYPT_ITCS_PADDING_NO(L"itcs-no-padding")` – паддинг не производить;
  - `BCRYPT_ITCS_PADDING_DEFAULT(L"itcs-default-padding")` — паддинг по умолчанию;
  - `BCRYPT_ITCS_PADDING_PKCS5(L"itcs-PKCS5-padding")` — PKCS5-паддинг;
  - `BCRYPT_ITCS_PADDING_ZERO(L"itcs-zero-padding")` — паддинг нулями;
  - `BCRYPT_ITCS_PADDING_UEC(L"itcs-UEC-padding")` — UEC-паддинг.
- `pbIV [in, out, optional]`. Адрес буфера, содержащего начальный вектор (IV) для расшифрования. Параметр `cbIV` содержит размер этого буфера. Если этот параметр задан, функция изменяет значение начального вектора (IV) в объекте ключа (см. описание свойства `BCRYPT_INITIALIZATION_VECTOR` объекта ключа (см. «Свойства криптографических примитивов» на стр. 11)).

  Это необязательный параметр. Если начальный вектор не используется или прежде был задан в свойстве объекта ключа, параметр может иметь значение `NULL`.

  Чтобы узнать необходимый размер IV, вызовите функцию `BCryptGetProperty` для получения свойства `BCRYPT_BLOCK_LENGTH`. Будет выдан размер блока для алгоритма, который совпадает с размером IV.

  Реализовано следующее поведение: параметр `pbIV` учитывается только при расшифровании первого блока данных, во время любых других вызовов это поле игнорируется. На такое поведение не стоит ориентироваться, поскольку в будущих реализациях выставление этого свойства будет вызывать сброс шифратора в начальное состояние, то есть в вызывающем приложении данное поле должно быть определено только при расшифровании первого блока данных, а для последующих блоков оно должно быть равно `NULL`.
- `cbIV [in]`. Размер в байтах буфера `pbIV`.

- `pbOutput [out, optional]`. Адрес буфера, предназначенного для получения открытого текста от функции. Параметр `cbOutput` содержит размер этого буфера.

  Если параметр имеет значение `NULL`, функция `BCryptDecrypt` вычисляет размер, необходимый для размещения открытого текста из зашифрованных данных, переданных в параметре `pbInput`. В этом случае расположение, указанное в параметре `pcbResult`, содержит этот размер, а функция возвращает `STATUS_SUCCESS`.

- `cbOutput [in]`. Размер в байтах буфера `pbOutput`. Этот параметр игнорируется, если параметр `pbOutput` имеет значение `NULL`.

- `pcbResult [out]`. Указатель на переменную типа `ULONG`, предназначенный для получения числа байтов информации, скопированной в буфер `pbOutput`. Если параметр `pbOutput` имеет значение `NULL`, описываемый параметр получает размер в байтах, необходимый для открытого текста.

- `dwFlags [in]`. Может принимать только значение `BCRYPT_BLOCK_PADDING`. Это означает, что расшифровывается последний блок данных и надо произвести анпаддинг расшифрованных данных.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 7. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_BUFFER_TOO_SMALL`	Размер, определенный в параметре `cbOutput`, недостаточен для зашифрованных данных.
`STATUS_INVALID_BUFFER_SIZE`	Параметр `cbInput` не кратен размеру блока, используемого в алгоритме, и флаг `BCRYPT_BLOCK_PADDING` не определен в параметре `dwFlags`.
`STATUS_INVALID_HANDLE`	Недопустимый дескриптор ключа в параметре `hKey`.
`STATUS_INVALID_PARAMETER`	Один или несколько недопустимых параметров.
`STATUS_NOT_SUPPORTED`	Алгоритм не поддерживает расшифрование.

Параметры `pbInput` и `pbOutput` могут указывать на один и тот же буфер. В этом случае будет происходить расшифрование “in place”.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptDestroyHash

Функция `BCryptDestroyHash` уничтожает объект хэша.

```
NTSTATUS WINAPI BCryptDestroyHash(
    _Inout BCRYPT_HASH_HANDLE hHash
);
```

Параметры:

- `hHash [in, out]`. Дескриптор, представляющий уничтожаемый объект хэша. Этот дескриптор ранее должен был быть создан посредством вызова функции `BCryptCreateHash`.

Возвращает код состояния, показывающий, успешно ли была выполнена функция.

Таблица 8. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_INVALID_HANDLE`	Недопустимый дескриптор хэша или MAC в параметре `hHash`.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

В зависимости от режимов, поддерживаемых процессором, функция `BCryptDestroyHash` может быть вызвана на пользовательском уровне или на уровне ядра. На уровне ядра функция должна выполняться в одном из двух режимов: `PASSIVE_LEVEL IRQL` или `DISPATCH_LEVEL IRQL`. Если текущий `IRQL` — `DISPATCH_LEVEL`, то дескриптор, передаваемый в параметр `hHash`, должен быть создан от алгоритма провайдера, открытого с флагом `BCRYPT_PROV_DISPATCH`.

В зависимости от режимов, поддерживаемых процессором, функция может быть вызвана на пользовательском уровне или на уровне ядра. На уровне ядра функция должна выполняться в одном из двух режимов: `PASSIVE_LEVEL IRQL` или `DISPATCH_LEVEL IRQL`. Если текущий `IRQL` — `DISPATCH_LEVEL`, то дескриптор, передаваемый в параметр `hKey`, должен быть создан от алгоритма провайдера, открытого с флагом `BCRYPT_PROV_DISPATCH`.

Функция BCryptDestroyKey

Функция `BCryptDestroyKey` уничтожает объект ключа.

```
NTSTATUS WINAPI BCryptDestroyKey(
    _Inout_  BCRYPT_KEY_HANDLE hKey
);
```

Параметры:

- `hKey [in, out]`. Дескриптор, представляющий уничтожаемый объект ключа.

Возвращает код состояния, показывающий, успешно ли была выполнена функция.

Таблица 9. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_INVALID_HANDLE`	Недопустимый дескриптор ключа в параметре hKey.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptDuplicateHash

Функция `BCryptDuplicateHash` делает полную копию объекта хэша или MAC. Копия объекта содержит состояние и все данные исходного объекта.

```
NTSTATUS WINAPI BCryptDuplicateHash(
    _In_   BCRYPT_HASH_HANDLE hHash,
    _Out_  BCRYPT_HASH_HANDLE *phNewHash,
    _Out_  PUCHAR pbHashObject,
    _In_   ULONG cbHashObject,
    _In_   ULONG dwFlags
);
```

Параметры:

- `hHash [in, out]`. Дескриптор копируемого хэш или MAC.
- `phNewHash [out]`. Указатель на значение типа `BCRYPT_HASH_HANDLE`, куда будет записан указатель на созданную копию объекта хэша или MAC-объекта.
- `pbHashObject [out]`. Указатель на буфер, куда будет записана созданная копия объекта хэша или MAC-объекта. Требуемый размер этого буфера определяется чтением свойства `BCRYPT_OBJECT_LENGTH` посредством вызова функции `BCryptGetProperty`, которая возвращает размер объекта хэша или MAC-объекта для заданного алгоритма. Эта память может быть освобождена только после уничтожения дескриптора, на который указывает `phNewHash`. Если данный параметр равен `NULL` и значение параметра `cbHashObject` равно нулю, то память под создаваемый объект выделяется системой.

Примечание. Данный способ выделения памяти поддерживается в ОС Windows, начиная с версии Vista SP1.

- `cbHashObject [in, optional]`. Размер буфера в байтах, на который указывает `pbHashObject`.

  Если данный параметр равен 0 и значение параметра `pbHashObject` — `NULL`, то память под создаваемый объект копии выделяется системой.

Примечание. Данный способ выделения памяти поддерживается в ОС Windows, начиная с версии Vista SP1.

- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. На данный момент не используется и должен быть равен нулю.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 10. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_BUFFER_TOO_SMALL`	Размер, определенный в параметре `cbHashObject`, недостаточен для объекта хэша или MAC.
`STATUS_INVALID_HANDLE`	Недопустимый дескриптор объекта хэша или MAC-объекта в параметре `hHash`.
`STATUS_INVALID_PARAMETER`	Один или несколько недопустимых параметров.

Данная функция полезна, когда вычисляется хэш-функция или MAC над общими данными. После того как общие данные обработаны, делается полное копирование объекта хэша или MAC-объекта, затем добавляются уникальные данные в каждый индивидуальный объект.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptDuplicateKey

Функция `BCryptDuplicateKey` делает полную копию объекта симметричного ключа. Копия объекта содержит состояние и все данные исходного объекта.

```
NTSTATUS WINAPI BCryptDuplicateKey(
    _In_   BCRYPT_KEY_HANDLE hKey,
    _Out_  BCRYPT_KEY_HANDLE *phNewKey,
    _Out_  PUCHAR pbKeyObject,
    _In_   ULONG cbKeyObject,
    _In_   ULONG dwFlags
);
```

Параметры:

- `hKey [in, out]`. Дескриптор копируемого объекта симметричного ключа.
- `phNewKey [out]`. Указатель на значение типа `BCRYPT_KEY_HANDLE`, куда будет записан указатель на созданную копию объекта симметричного ключа.
- `pbKeyObject [out]`. Указатель на буфер, куда будет записана созданная копия объекта симметричного ключа. Требуемый размер этого буфера определяется чтением свойства `BCRYPT_OBJECT_LENGTH` посредством вызова функции `BCryptGetProperty`, которая возвращает размер объекта симметричного ключа заданного алгоритма.

  Эта память может быть освобождена только после уничтожения дескриптора, на который указывает `phNewKey`. Если данный параметр равен `NULL` и значение параметра `cbKeyObject` — 0, то память под создаваемый объект выделяется системой.

  

  Примечание. Данный способ выделения памяти поддерживается в ОС Windows, начиная с версии Vista SP1.

- `cbKeyObject [in, optional]`. Размер буфера в байтах, на который указывает `pbKeyObject`. Если данный параметр равен 0 и значение параметра `pbKeyObject` — NULL, то память под создаваемый объект копии выделяется системой.

  

  Примечание. Данный способ выделения памяти поддерживается в ОС Windows, начиная с версии Vista SP1.

- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. На данный момент не используется и должен быть равен нулю.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 11. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_BUFFER_TOO_SMALL`	Размер, определенный в параметре `cbKeyObject`, недостаточен для объекта симметричного ключа.
`STATUS_INVALID_HANDLE`	Недопустимый дескриптор объекта симметричного ключа в параметре `hKey`.

Возвращаемое значение	Описание
STATUS_INVALID_PARAMETER	Один или несколько недопустимых параметров.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима PASSIVE_LEVEL IRQL.

Функция BCryptEncrypt

Функция BCryptEncrypt предназначена для зашифрования блоков данных.

```
NTSTATUS WINAPI BCryptEncrypt(
    __inout      BCRYPT_KEY_HANDLE hKey,
    __in         PUCHAR pbInput,
    __in         ULONG cbInput,
    __in_opt     VOID *pPaddingInfo,
    __inout_opt  PUCHAR pbIV,
    __in         ULONG cbIV,
    __out_opt    PUCHAR pbOutput,
    __in         ULONG cbOutput,
    __out        ULONG *pcbResult,
    __in         ULONG dwFlags
);
```

Параметры:

- hKey [in, out]. Дескриптор ключа для зашифрования данных. Может быть получен от одной из функций создания ключей, например BCryptGenerateSymmetricKey, BCryptGenerateKeyPair или BCryptImportKey.
- pbInput [in]. Адрес буфера с открытым текстом, предназначенным для зашифрования. Параметр cbInput содержит размер этого текста.
- cbInput [in]. Размер в байтах содержимого буфера pbInput, предназначенного для зашифрования.
- pPaddingInfo [in, optional]. Имеет смысл только в том случае, если выставлен флаг BCRYPT_BLOCK_PADDING, который является указателем на структуру типа BCRYPT_ITCS_PADDING_INFO, определяющую тип паддинга. Если этот параметр не задан, производится паддинг по умолчанию.

  Если выставлен флаг BCRYPT_BLOCK_PADDING, то параметр pPaddingInfo определяет тип паддинга. В противном случае (если pPaddingInfo равен NULL и выставлен флаг BCRYPT_BLOCK_PADDING) паддинг считается паддингом по умолчанию (PKCS5-паддинг).

Паддинг имеет смысл только для блочных режимов шифрования (CBC и ECB), поэтому для потоковых режимов шифрования данный флаг и поле `pPaddingInfo` игнорируются:

```
typedef struct __BCRYPT_ITCS_PADDING_INFO
{
LPCWSTR  szPaddingType;
} BCRYPT_ITCS_PADDING_INFO, *PBCRYPT_ITCS_PADDING_INFO;
```

`szPaddingType` может принимать следующие значения:

- `BCRYPT_ITCS_PADDING_NO(L"itcs-no-padding")` – паддинг не производить;
- `BCRYPT_ITCS_PADDING_DEFAULT(L"itcs-default-padding")` — паддинг по умолчанию;
- `BCRYPT_ITCS_PADDING_PKCS5(L"itcs-PKCS5-padding")` — PKCS5-паддинг;
- `BCRYPT_ITCS_PADDING_ZERO(L"itcs-zero-padding")` — паддинг нулями;
- `BCRYPT_ITCS_PADDING_UEC(L"itcs-UEC-padding")` — UEC-паддинг.

- `pbIV [in, out, optional]`. Адрес буфера, содержащего начальный вектор (IV) для зашифрования. Параметр `cbIV` содержит размер этого буфера. Функция изменяет значение начального вектора (IV) в объекте ключа, если этот параметр задан (см. описание свойства `BCRYPT_INITIALIZATION_VECTOR` объекта ключа (см. «Свойства криптографических примитивов» на стр. 11)).

  Это необязательный параметр. Если начальный вектор не используется или прежде был выставлен в объект ключа, параметр может иметь значение `NULL`.

  Чтобы узнать необходимый размер IV, вызовите функцию `BCryptGetProperty` для получения свойства `BCRYPT_BLOCK_LENGTH`. Будет выдан размер блока для алгоритма, который совпадает с размером IV.

  На данный момент реализовано следующее поведение: параметр `pbIV` учитывается только при шифровании первого блока данных, во время любых других вызовов это поле игнорируется. На такое поведение не стоит ориентироваться, поскольку в будущих реализация выставление этого свойства будет вызывать сброс шифратора в начальное состояние, то есть в вызывающем приложении данное поле должно быть определено только при шифровании первого блока данных, а для последующих блоков это поле должно иметь значение `NULL`.

- `cbIV [in]`. Размер в байтах буфера `pbIV`.
- `pbOutput [out, optional]`. Адрес буфера, который получает зашифрованный текст от функции. Параметр `cbOutput` содержит размер этого буфера.

  Если параметр имеет значение `NULL`, функция `BCryptEncrypt` вычисляет размер, необходимый для размещения зашифрованного текста из данных, переданных в параметре `pbInput`. В этом случае расположение, указанное в параметре `pcbResult`, содержит этот размер, а функция возвращает `STATUS_SUCCESS`.

- `cbOutput [in]`. Размер в байтах буфера `pbOutput`. Этот параметр игнорируется, если параметр `pbOutput` имеет значение `NULL`.
- `pcbResult [out]`. Указатель на переменную типа `ULONG`, который получает число байтов информации, скопированное в буфер `pbOutput`. Если параметр `pbOutput` имеет значение

NULL, описываемый параметр получает размер в байтах, необходимый для зашифрованного текста.

- dwFlags [in]. Может принимать только значение BCRYPT_BLOCK_PADDING, которое обозначает, что расшифровывается последний блок данных и необходимо произвести анпаддинг расшифрованных данных.

Возвращает код состояния, показывающий, успешно ли была выполнена функция.

Таблица 12. Возвращаемые значения

Возвращаемое значение	Описание
STATUS_SUCCESS	Функция выполнена успешно.
STATUS_BUFFER_TOO_SMALL	Размер, определенный в параметре cbOutput, недостаточен для зашифрованных данных.
STATUS_INVALID_BUFFER_SIZE	Параметр cbInput не кратен размеру блока, используемого в алгоритме, и в параметре dwFlags не определен флаг BCRYPT_BLOCK_PADDING.
STATUS_INVALID_HANDLE	Недопустимый дескриптор ключа в параметре hKey.
STATUS_INVALID_PARAMETER	Один или несколько недопустимых параметров.
STATUS_NOT_SUPPORTED	Алгоритм не поддерживает зашифрование.

Параметры pbInput и pbOutput могут указывать на один и тот же буфер. В этом случае будет происходить зашифрование “in place”. Если зашифрованные данные имеют большую длину, чем незашифрованные данные, буфер должен иметь достаточную длину для зашифрованных данных.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима PASSIVE_LEVEL IRQL.

Функция BCryptExportKey

Функция BCryptExportKey предназначена для экспорта ключей в блоб, который может быть использован в дальнейшем.

```
NTSTATUS WINAPI BCryptExportKey(
    __in   BCRYPT_KEY_HANDLE hKey,
    __in   BCRYPT_KEY_HANDLE hExportKey,
    __in   LPCWSTR pszBlobType,
    __out  PUCHAR pbOutput,
    __in   ULONG cbOutput,
```

```
    __out  ULONG *pcbResult,
    __in   ULONG dwFlags
);
```

Параметры:

- `hKey [in]`. Дескриптор ключа, предназначенный для экспорта.
- `hExportKey [in]`. Дескриптор симметричного ключа, на котором происходит экспорт (дескриптор ключа защиты).

Примечание. Хотя для Microsoft CNG-провайдеров в операционных системах Windows Server 2008 и Windows Vista этот параметр не используется и должен иметь значение `NULL`, для провайдера ViPNet CNG ключ защиты должен быть указан при экспорте любого симметричного ключа и любой асимметричной ключевой пары (закрытого ключа), иначе возвращается ошибка.

При экспорте только открытой части асимметричного ключа (указан тип экспорта `BCRYPT_PUBLIC_KEY_BLOB`) этот параметр должен быть равен `NULL`.

- `pszBlobType [in]`. Указатель на оканчивающуюся нулем Юникод-строку, которая содержит идентификатор типа экспортируемого блоба. Параметр может принимать одно из следующих значений:

Таблица 13. Значения параметра `pszBlobType`

Значение	Описание
`BCRYPT_OPAQUE_KEY_BLOB`	Экспорт симметричного ключа в формате, специфичном для определенного криптопровайдера. Объекты Opaque BLOB передавать невозможно, поэтому их необходимо импортировать с помощью криптопровайдера, создавшего данный блоб. Объекты Opaque BLOB предназначены только для передачи ключей при межпроцессном взаимодействии и не могут продолжать существование при смене версии провайдера. Экспортируется полное состояние объекта ключа: первоначальное значение ключа, текущее значение ключа, значение IV, режим шифрования, OID-параметры шифрования, признак использования мешинга ключа, состояние шифратора. Формат блоба совпадает с форматом OPAQUEKEYBLOB из CSP. Соответственно, данный блоб может быть импортирован в CSP.
`BCRYPT_PUBLIC_KEY_BLOB`	Экспорт общих открытых ключей любых типов. Тип ключа в блобе определяется элементом `Magic` структуры `BCRYPT_KEY_BLOB`. Формат блоба совпадает с форматом `PUBLICKEYBLOB` из CSP. Соответственно, данный блоб может быть импортирован в CSP.
`BCRYPT_ITCS_SIMPLE_KEY_BLOB`	Используется для экспорта одного симметричного ключа на другом симметричном ключе. Экспортируются только значение симметричного ключа перед началом шифрования и OID-параметры шифрования. При импорте блоба данного типа все параметры шифрования, кроме OID-параметров, необходимо выставлять явно, иначе будут использованы

Значение	Описание
	параметры по умолчанию. Формат блоба совпадает с форматом SIMPLEBLOB из CSP. Соответственно, данный блоб может быть импортирован в CSP.
BCRYPT_PRIVATE_KEY_BLOB	Представляет собой общий закрытый ключ любого типа (на данный момент реализован только для асимметричных ключей). Закрытый ключ не обязательно содержит открытый ключ. Тип ключа в блобе определяется элементом Magic структуры BCRYPT_KEY_BLOB. Формат блоба совпадает с форматом PRIVATEKEYBLOB из CSP. Соответственно, данный блоб может быть импортирован в CSP.

- pbOutput [out]. Адрес буфера, который получает блоб ключа. Параметр cbOutput содержит размер этого буфера. Если этот параметр принимает значение NULL, рассматриваемая функция поместит необходимый размер в байтах в переменную типа ULONG, на которую указывает параметр pcbResult.
- cbOutput [in]. Содержит размер в байтах буфера pbOutput.
- pcbResult [out]. Указатель на переменную типа ULONG, который получает число байтов информации, скопированное в буфер pbOutput. Если параметр pbOutput принимает значение NULL, рассматриваемая функция поместит необходимый размер в байтах в переменную типа ULONG, на которую указывает этот параметр.
- dwFlags [in]. Набор флагов, изменяющих поведение функции. На данный момент не определено ни одного флага, поэтому параметр должен быть равен нулю.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 14. Возвращаемые значения

Возвращаемое значение	Описание
STATUS_SUCCESS	Функция выполнена успешно.
STATUS_BUFFER_TOO_SMALL	Размер, определенный в параметре cbOutput, недостаточен для зашифрованных данных.
STATUS_INVALID_HANDLE	Недопустимый дескриптор ключа в параметре hKey.
STATUS_INVALID_PARAMETER	Один или несколько недопустимых параметров.
STATUS_NOT_SUPPORTED	Указанный тип блоба не поддерживается провайдером.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptFinalizeKeyPair

Функция `BCryptFinalizeKeyPair` предназначена для завершения создания ключевой пары. Функция должна быть вызвана после `BCryptGenerateKeyPair`. Если до вызова функции ключ не был задан явно через выставление параметра `BCRYPT_ITCS_PRIVATE_KEY_VALUE`, то при вызове данной функции генерируется случайная эфемерная ключевая пара.

```
NNTSTATUS WINAPI BCryptFinalizeKeyPair(
    __inout  BCRYPT_KEY_HANDLE hKey,
    __in     ULONG dwFlags
);
```

Параметры:

- `hKey [in, out]`. Дескриптор объекта закрытого асимметричного ключа для завершения.
- `dwFlags`. Набор флагов, изменяющих поведение функции. На данный момент не используется и должен быть равен нулю.

Возвращает код состояния, показывающий, успешно ли выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 15. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_INVALID_HANDLE`	Недопустимый дескриптор ключа в параметре `hKey`.
`STATUS_INVALID_PARAMETER`	Один или несколько недопустимых параметров.
`STATUS_NOT_SUPPORTED`	Заданный провайдер не поддерживает асимметричное шифрование ключей.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptFinishHash

Функция `BcryptFinishHash` извлекает значение хэш-функции для данных, ранее накопленных через вызов `BCryptHashData`.

```
NTSTATUS WINAPI BCryptFinishHash(
    _Inout_  BCRYPT_HASH_HANDLE hHash,
    _Out_    PUCHAR pbOutput,
    _In_     ULONG cbOutput,
    _In_     ULONG dwFlags
);
```

Параметры:

- `hHash [in, out]`. Дескриптор хэш или MAC-объекта, используемый для вычисления хэша или MAC. Этот дескриптор должен быть прежде открыт функцией `BcryptCreateHash`. После того как будет вызвана данная функция, значение дескриптора `hHash` становится недействительным и не может быть использовано в других операциях хэширования или вычисления MAC.
- `pbOutput [out]`. Указатель на буфер, куда будет записано вычисленное значение хэш-функции или MAC. Параметр `cbOutput` содержит размер этого буфера в байтах.
- `cbOutput [in]`. Размер буфера `pbOutput` в байтах — размер буфера, необходимого для записи значения хэш-функции или MAC. Значение может быть получено через свойство `BCRYPT_HASH_LENGTH` посредством вызова функции `BCryptGetProperty`.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. На данный момент не используется и должен быть равен нулю.

Возвращает код состояния, показывающий, успешно ли выполнена функция.

Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 16. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_INVALID_HANDLE`	Недопустимый дескриптор хэша в `hHash`. После вызова функции `BcryptFinishHash` дескриптор становится недействительным и не может быть повторно использован в других операциях, кроме функции `BCryptDestroyHash`.
`STATUS_INVALID_PARAMETER`	Один или несколько недопустимых параметров.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptGenerateKeyPair

Функция `BCryptGenerateKeyPair` предназначена для создания пустых ключевых пар. После создания ключа с помощью этой функции используйте функцию `BCryptSetProperty` для задания его свойств. Для завершения создания ключевой пары необходимо вызвать функцию `BCryptFinalizeKeyPair`.

```
NTSTATUS WINAPI BCryptGenerateKeyPair(
    __inout  BCRYPT_ALG_HANDLE hAlgorithm,
    __out    BCRYPT_KEY_HANDLE *phKey,
    __in     ULONG dwLength,
    __in     ULONG dwFlags
);
```

Параметры:

- `hAlgorithm [in, out]`. Дескриптор провайдера алгоритма, созданного с помощью функции `BCryptOpenAlgorithmProvider`. Этот алгоритм задается, когда созданный провайдер должен поддерживать асимметричное шифрование ключей.
- `phKey [out]`. Указатель на `BCRYPT_KEY_HANDLE`, который получает дескриптор ключа. Этот дескриптор затем используется функциями, которым необходим ключ, например функцией `BCryptEncrypt`. Когда дескриптор более не требуется, его необходимо передать функции `BCryptDestroyKey`.
- `dwLength [in]`. Значение должно быть равно 0 или длине закрытого ключа в битах.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. Если не определено ни одного флага, параметр должен равняться нулю.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 17. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_INVALID_HANDLE`	Недопустимый дескриптор в параметре `hKey`.
`STATUS_INVALID_PARAMETER`	Один или несколько недопустимых параметров.
`STATUS_NOT_SUPPORTED`	Заданный провайдер не поддерживает асимметричное шифрование ключей.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptGenerateSymmetricKey

Функция предназначена для создания объекта симметричного ключа.

```
NTSTATUS WINAPI BCryptGenerateSymmetricKey(
    _Inout_    BCRYPT_ALG_HANDLE hAlgorithm,
    _Out_      BCRYPT_KEY_HANDLE *phKey,
    _Out_opt_  PUCHAR pbKeyObject,
    _In_       ULONG cbKeyObject,
    _In_       PUCHAR pbSecret,
    _In_       ULONG cbSecret,
    _In_       ULONG dwFlags
);
```

Параметры:

- `hAlgorithm [in]`. Дескриптор провайдера алгоритма, созданного с помощью функции `BCryptOpenAlgorithmProvider`. Заданный алгоритм должен поддерживать интерфейс симметричного шифрования. Для реализации интерфейса ViPNet CNG BCrypt — это только алгоритм `BCRYPT_ITCS_ENCRYPT_89_ALGID`.
- `phKey [out]`. Указатель на `BCRYPT_KEY_HANDLE`, который получает дескриптор созданного объекта ключа. Этот дескриптор затем используется функциями, которым необходим ключ, например функцией `BCryptEncrypt`. Когда дескриптор более не требуется, его необходимо передать функции `BCryptDestroyKey`.
- `pbKeyObject [out]`. Указатель на буфер, куда будет записан созданный объект симметричного ключа. Требуемый размер этого буфера определяется чтением свойства `BCRYPT_OBJECT_LENGTH` посредством вызова функции `BCryptGetProperty`, которая возвращает размер объекта симметричного ключа заданного алгоритма.

  Эта память может быть освобождена только после уничтожения дескриптора, на который указывает `phKey`. Если данный параметр равен `NULL` и значение параметра `cbKeyObject` равно 0, то память под создаваемый объект выделяется системой.

  Примечание. Данный способ выделения памяти поддерживается в ОС Windows, начиная с версии Vista SP1.

- `cbKeyObject [in, optional]`. Размер буфера в байтах, на который указывает `pbKeyObject`. Если данный параметр равен 0 и значение параметра `pbKeyObject` — `NULL`, то память под создаваемый объект копии выделяется системой.

Примечание. Данный способ выделения памяти поддерживается в ОС Windows, начиная с версии Vista SP1.

- `pbSecret [in]`. Указатель на буфер, в который будет записан создаваемый объект симметричного ключа. При этом в параметре `cbSecret` должно содержаться значение в точности равное размеру значения симметричного ключа. Для алгоритма `BCRYPT_ITCS_ENCRYPT_89_ALGID` — это 32 байта. Если `pbSecret` равен `NULL`, то и `cbSecret` должен быть равен нулю. Тогда в созданном объекте симметричного ключа будет сгенерировано случайное значение сессионного ключа.
- `cbSecret [in]`. Размер буфера `pbSecret` в байтах.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. На данный момент не используется и должен быть равен нулю.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 18. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_BUFFER_TOO_SMALL`	Размер, определенный в параметре `cbKeyObject`, недостаточен для объекта симметричного ключа.
`STATUS_INVALID_HANDLE`	Недопустимый дескриптор алгоритм провайдера в параметре `hAlgorithm`.
`STATUS_INVALID_PARAMETER`	Один или несколько недопустимых параметров.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptGenRandom

Функция `BcryptGenRandom` предназначена для генерации случайных чисел.

```
NTSTATUS WINAPI BCryptGenRandom(
    _Inout_  BCRYPT_ALG_HANDLE hAlgorithm,
    _Inout_  PUCHAR pbBuffer,
    _In_     ULONG cbBuffer,
    _In_     ULONG dwFlags
);
```

Параметры:

- `hAlgorithm [in]`. Дескриптор провайдера алгоритма, созданного с помощью функции `BCryptOpenAlgorithmProvider`. Заданный алгоритм должен поддерживать интерфейс генерации случных чисел (для реализации интерфейса ViPNet CNG BCrypt этим алгоритмом является `BCRYPT_ITCS_RNG_ALGID`);
- `pbBuffer [in, out]`. Указатель на буфер, в который будет записана сгенерированная случайная последовательность байтов. Размер этого буфера определяется параметром `cbBuffer`.
- `cbBuffer [in]`. Размер буфера `pbBuffer` в байтах.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. На данный момент не используется и должен быть равен нулю.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 19. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_INVALID_HANDLE`	Недопустимый дескриптор провайдера алгоритма в параметре `hAlgorithm`.
`STATUS_INVALID_PARAMETER`	Один или несколько недопустимых параметров.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptGetProperty

Функция `BCryptGetProperty` возвращает значение именованного свойства объекта CNG.

```
NTSTATUS WINAPI BCryptGetProperty(
    __in   BCRYPT_HANDLE hObject,
    __in   LPCWSTR pszProperty,
```

```
    __out  PUCHAR pbOutput,
    __in   ULONG cbOutput,
    __out  ULONG *pcbResult,
    __in   ULONG dwFlags
);
```

Параметры:

- `hObject [in]`. Дескриптор, представляющий объект CNG для получения значения свойства.
- `pszProperty [in]`. Указатель на оканчивающуюся нулем Юникод-строку, содержащую имя свойства, которое необходимо извлечь. Для реализации интерфейса ViPNet CNG BCrypt множество поддерживаемых свойства перечислено выше в разделе Свойства криптографических примитивов (на стр. 11).
- `pbOutput [out]`. Адрес буфера, который получает значение свойства. Параметр `cbOutput` содержит размер этого буфера.
- `cbOutput [in]`. Размер в байтах буфера `pbOutput`.
- `pcbResult [out]`. Указатель на переменную типа `ULONG`, который получает число байтов информации, скопированное в буфер `pbOutput`. Если параметр `pbOutput` принимает значение `NULL`, рассматриваемая функция поместит необходимый размер в байтах в расположение, на которое указывает этот параметр.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. На данный момент не определено ни одного флага, поэтому параметр должен равняться нулю.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 20. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_BUFFER_TOO_SMALL`	Размер буфера, определенный в параметре `cbOutput`, недостаточен для значения свойства.
`STATUS_INVALID_HANDLE`	Недопустимый дескриптор в параметре `hObject`.
`STATUS_INVALID_PARAMETER`	Один или несколько недопустимых параметров.
`STATUS_NOT_SUPPORTED`	Параметром `pszProperty` задано неподдерживаемое именованное свойство.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptHashData

Функция `BCryptHashData` предназначена для одного шага хэширования блока данных.

```
NTSTATUS WINAPI BCryptHashData(
    _Inout_  BCRYPT_HASH_HANDLE hHash,
    _In_     PUCHAR pbInput,
    _In_     ULONG cbInput,
    _In_     ULONG dwFlags
);
```

Параметры:

- `hHash [in, out]`. Дескриптор хэш или MAC-объекта, используемый для выполнения операции. Этот дескриптор должен быть предварительно открыт функцией `BcryptCreateHash`.
- `pbInput [in]`. Указатель на буфер, содержащий данные для обработки. Параметр `cbInput` содержит число байтов в этом буфере. Функция не изменяет содержимое этого буфера.
- `cbInput [in]`. Число байтов в буфере `pbInput`.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. На данный момент не используется и должен быть равен нулю.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 21. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_INVALID_HANDLE`	Недопустимый дескриптор хэша в `hHash`. После вызова функции `BcryptFinishHash` дескриптор становится недействительным и не может быть использован повторно в других операциях, кроме функции `BCryptDestroyHash`.
`STATUS_INVALID_PARAMETER`	Один или несколько недопустимых параметров.

Для того чтобы объединить несколько буферов при вычислении хэша или MAC, надо вызвать функцию `BcryptHashData` несколько раз на разных буферах данных. После вызова функции `BcryptFinishHash` дескриптор `hHash` становится недействительным и не может быть повторно использован в других операциях, кроме функции `BCryptDestroyHash`.

Если вызвать подряд последовательно функции `BcryptCreateHash` и `BcryptFinishHash` (без вызова `BcryptHashData`), то будет вычислено значение хэш-функции или MAC на «пустых» данных.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptImportKey

Функция `BCryptImportKey` предназначена для импорта симметричного ключа из блоба ключа. Для импорта ключевой пары используется функция `BCryptImportKeyPair`.

```
NTSTATUS WINAPI BCryptImportKey(
    __in     BCRYPT_ALG_HANDLE hAlgorithm,
    __inout  BCRYPT_KEY_HANDLE hImportKey,
    __in     LPCWSTR pszBlobType,
    __out    BCRYPT_KEY_HANDLE *phKey,
    __out    PUCHAR pbKeyObject,
    __in     ULONG cbKeyObject,
    __in     PUCHAR pbInput,
    __in     ULONG cbInput,
    __in     ULONG dwFlags
);
```

Параметры:

- `hAlgorithm [in]`. Дескриптор провайдера алгоритма для импорта ключа. Чтобы получить этот дескриптор, вызовите функцию `BCryptOpenAlgorithmProvider`.
- `hImportKey [in, out]`. Дескриптор симметричного ключа шифрования ключей, необходимый для распаковки блоба ключа в параметре `pbInput`.

  Примечание. Дескриптор должен поддерживаться провайдером, который предоставляет импортируемый ключ.

- `pszBlobType [in]`. Указатель на оканчивающуюся нулем Юникод-строку, которая содержит идентификатор типа экспортируемого блоба, содержащегося в буфере `pbInput`. Параметр может принимать одно из следующих значений:

  Таблица 22. Значения параметра `pszBlobType`

Значение	Описание
`BCRYPT_OPAQUE_KEY_BLOB`	Импорт блоба симметричного ключа в формате, специфичном для определенного криптопровайдера. Объекты Opaque BLOB передавать невозможно, поэтому их необходимо импортировать с помощью криптопровайдера, сгенерировавшего данный блоб. Объекты Opaque BLOB предназначены только для передачи ключей при межпроцессном взаимодействии и не могут продолжать существование при смене версии провайдера.

Значение	Описание
	При данном типе экспорта импортируется полное состояние объекта ключа: первоначальное значение ключа, текущее значение ключа, значение IV, режим шифрования, OID-параметры шифрования, признак использования мешинга ключа, состояние шифратора. Формат блоба совпадает с форматом OPAQUEKEYBLOB из CSP. Соответственно, данный блоб может быть получен экспортом из CSP.
`BCRYPT_ITCS_SIMPLE_KEY_BLOB`	Используется для импорта одного симметричного ключа на другом симметричном ключе. В этом блобе содержится только значение симметричного ключа перед началом шифрования и OID-параметры шифрования. При импорте блоба данного типа все параметры шифрования, кроме OID-параметров шифрования, необходимо выставлять явно, иначе будут использованы параметры по умолчанию. Формат блоба совпадает с форматом `SIMPLEBLOB` из CSP, соответственно, данный блоб может быть получен экспортом из CSP.

- `phKey [out]`. Указатель на `BCRYPT_KEY_HANDLE`, который получает дескриптор импортированного ключа. Этот дескриптор затем используется функциями, которым необходим ключ, например, функцией `BCryptEncrypt`. Когда дескриптор более не требуется, его необходимо передать функции `BCryptDestroyKey`.
- `pbKeyObject [out]`. Указатель на буфер, куда будет записан созданный объект импортированного симметричного ключа. Требуемый размер этого буфера определяется чтением свойства `BCRYPT_OBJECT_LENGTH` посредством вызова функции `BCryptGetProperty`, которая возвращает размер объекта симметричного ключа заданного алгоритма.

  Эта память может быть освобождена только после уничтожения дескриптора, на который указывает `phKey`.

  Если данный параметр равен `NULL` и значение параметра `cbKeyObject` равно 0, то память под создаваемый объект выделяется системой.

  Примечание. Данный способ выделения памяти поддерживается в ОС Windows, начиная с версии Vista SP1.

- `cbKeyObject [in, optional]`. Размер буфера в байтах, на который указывает `pbKeyObject`. Если этот параметр равен 0 и значение параметра `pbKeyObject` равно `NULL`, то память под создаваемый объект копии выделяется системой.

  Примечание. Данный способ выделения памяти поддерживается в ОС Windows, начиная с версии Vista SP1.

- `pbInput [in]`. Адрес буфера, который содержит импортируемый блоб ключа. Параметр `cbInput` содержит размер этого буфера. Параметр `pszBlobType` задает тип блоба ключа, содержащегося в этом буфере.
- `cbInput [in]`. Размер в байтах буфера `pbIV`.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. На данный момент не определено ни одного флага, поэтому параметр должен равняться нулю.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 23. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_BUFFER_TOO_SMALL`	Размер объекта ключа, определенный в параметре `cbKeyObject`, недостаточен для объекта ключа.
`STATUS_INVALID_HANDLE`	Недопустимый алгоритм в параметре `hAlgorithm`.
`STATUS_INVALID_PARAMETER`	Один или несколько недопустимых параметров.
`STATUS_NOT_SUPPORTED`	Провайдер алгоритма, определенный в параметре `hAlgorithm`, не поддерживает тип блоба, определенный в параметре `pszBlobType`.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptImportKeyPair

Функция `BCryptImportKeyPair` предназначена для импорта ключевой пары из блоба ключа. Для импорта пары симметричных ключей используется функция `BCryptImportKey`.

```
NTSTATUS WINAPI BCryptImportKeyPair(
    __in     BCRYPT_ALG_HANDLE hAlgorithm,
    __inout  BCRYPT_KEY_HANDLE hImportKey,
    __in     LPCWSTR pszBlobType,
    __out    BCRYPT_KEY_HANDLE *phKey,
    __in     PUCHAR pbInput,
    __in     ULONG cbInput,
    __in     ULONG dwFlags
);
```

Параметры:

- `hAlgorithm [in]`. Дескриптор провайдера алгоритма для импорта ключа. Чтобы получить этот дескриптор, вызовите функцию `BCryptOpenAlgorithmProvider`.
- `hImportKey [in, out]`. Дескриптор симметричного ключа шифрования ключей, необходимый для распаковки блоба ключа в параметре `pbInput`.
  Если тип объекта — `BLOB BCRYPT_PUBLIC_KEY_BLOB`, то данный параметр должен быть равен `NULL`.

Примечание. Дескриптор должен поддерживаться тем же провайдером, который предоставляет импортируемый ключ.

- `pszBlobType [in]`. Указатель на оканчивающуюся нулем Юникод-строку, которая содержит идентификатор типа экспортируемого блоба, содержащегося в буфере `pbInput`. Параметр может принимать одно из следующих значений:

  Таблица 24. Значения параметра `pszBlobType`

Значение	Описание
`BCRYPT_PUBLIC_KEY_BLOB`	Представляет собой общий открытый ключ любого типа. Тип ключа в блобе определяется элементом `Magic` структуры `BCRYPT_KEY_BLOB`. Формат блоба совпадает с форматом `PUBLICKEYBLOB` из CSP. Соответственно, данный блоб может быть получен экспортом из CSP.
`BCRYPT_PRIVATE_KEY_BLOB`	Представляет собой общий закрытый ключ любого типа. Закрытый ключ не обязательно содержит открытый ключ. Тип ключа в блобе определяется элементом `Magic` структуры `BCRYPT_KEY_BLOB`. Формат блоба совпадает с форматом `PRIVATEKEYBLOB` из CSP. Соответственно, данный блоб может быть получен экспортом из CSP.

- `phKey [out]`. Указатель на `BCRYPT_KEY_HANDLE`, который получает дескриптор импортированного ключа. Этот дескриптор затем используется функциями, которым необходим ключ, например функцией `BCryptSignHash`. Когда дескриптор более не требуется, его необходимо передать функции `BCryptDestroyKey`.
- `pbInput [in]`. Адрес буфера, который содержит импортируемый блоб ключа. Параметр `cbInput` содержит размер этого буфера. Параметр `pszBlobType` задает тип блоба ключа, содержащегося в этом буфере.
- `cbInput [in]`. Размер в байтах буфера `pbInput`.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. На данный момент не определено ни одного флага, поэтому параметр должен равняться нулю.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 25. Возвращаемые значения

Возвращаемое значение	Описание
STATUS_SUCCESS	Функция выполнена успешно.
STATUS_INVALID_HANDLE	Недопустимый алгоритм в параметре hAlgorithm.
STATUS_INVALID_PARAMETER	Один или несколько недопустимых параметров.
STATUS_NOT_SUPPORTED	Провайдер алгоритма, определенный в параметре hAlgorithm, не поддерживает тип блоба, определенный в параметре pszBlobType.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима PASSIVE_LEVEL IRQL.

Функция BCryptOpenAlgorithmProvider

Функция BCryptOpenAlgorithmProvider предназначена для загрузки и инициализации алгоритм-провайдера CNG BCrypt.

```
NTSTATUS WINAPI BCryptOpenAlgorithmProvider(
    __out  BCRYPT_ALG_HANDLE *phAlgorithm,
    __in   LPCWSTR pszAlgId,
    __in   LPCWSTR pszImplementation,
    __in   DWORD dwFlags
);
```

Параметры:

- phAlgorithm [out]. Указатель на переменную BCRYPT_ALG_HANDLE, которая получает дескриптор создаваемого алгоритм-провайдера CNG BCrypt. Когда дескриптор больше не требуется, его необходимо передать функции BCryptCloseAlgorithmProvider.
- pszAlgId [in]. Указатель на оканчивающуюся нулем Юникод-строку, определяющую запрашиваемый криптографический алгоритм. Для реализации интерфейса ViPNet CNG BCrypt множество поддерживаемых алгоритмов перечислено в разделе Список алгоритмов, добавленных для поддержки ГОСТ (на стр. 7).
- pszImplementation [in]. Указатель на Юникод-строку, оканчивающуюся нулем, идентифицирующую требуемый провайдер CNG. Это необязательный параметр. Если он не нужен, параметр может иметь значение NULL. Если параметр имеет значение NULL, будет загружен провайдер по умолчанию для определенного алгоритма. Для открытия свойств алгоритмов ГОСТ необходимо указывать NULL или имя провайдера ViPNet CNG BCrypt "Infotecs Primitive Provider CU".

- `dwFlags [in]`. На данный момент поддерживается только флаг `BCRYPT_ALG_HANDLE_HMAC_FLAG`, который имеет смысл только для алгоритмов, поддерживающих интерфейс хэширования, и обозначает, что надо считать не хэш-функцию, а HMAC для заданного алгоритма (см. «Список алгоритмов, добавленных для поддержки ГОСТ» на стр. 7). Для алгоритма `BCRYPT_ITCS_MAC_89_ALGID` значение флага `BCRYPT_ALG_HANDLE_HMAC_FLAG` игнорируется (алгоритм работает одинаково как при установленном флаге `BCRYPT_ALG_HANDLE_HMAC_FLAG`, так и при сброшенном).

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 26. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_NOT_FOUND`	Для заданного ID алгоритма провайдер не найден.
`STATUS_INVALID_PARAMETER`	Один или несколько недопустимых параметров.
`STATUS_NO_MEMORY`	Обнаружен сбой выделения памяти.

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptSetProperty

Функция `BCryptSetProperty` предназначена для задания значения именованного свойства объекта CNG BCrypt.

```
NTSTATUS WINAPI BCryptSetProperty(
    __inout  BCRYPT_HANDLE hObject,
    __in     LPCWSTR pszProperty,
    __in     PUCHAR pbInput,
    __in     ULONG cbInput,
    __in     ULONG dwFlags
);
```

Параметры:

- `hObject [in, out]`. Дескриптор, представляющий объект CNG BCrypt для задания значения свойства.
- `pszProperty [in]`. Указатель на оканчивающуюся нулем Юникод-строку, содержащую имя свойства, которое необходимо задать. Для реализации интерфейса ViPNet CNG BCrypt множество поддерживаемых свойств перечислено в разделе Свойства криптографических примитивов (на стр. 11).

- `pbInput [in]`. Адрес буфера, который содержит новое значение свойства. Параметр `cbInput` содержит размер этого буфера.
- `cbInput [in]`. Размер в байтах буфера `pbInput`.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. На данный момент не определено ни одного флага, поэтому параметр должен равняться нулю.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 27. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_INVALID_HANDLE`	Недопустимый дескриптор в параметре `hObject`.
`STATUS_INVALID_PARAMETER`	Один или несколько недопустимых параметров.
`STATUS_NOT_SUPPORTED`	Параметром `pszProperty` задано неподдерживаемое или доступное только для чтения именованное свойство.

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptSignHash

Функция `BCryptSignHash` предназначена для создания подписи значения хэша.

```
NTSTATUS WINAPI BCryptSignHash(
    __in      BCRYPT_KEY_HANDLE hKey,
    __in_opt  VOID *pPaddingInfo,
    __in      PBYTE pbInput,
    __in      DWORD cbInput,
    __out     PBYTE pbOutput,
    __in      DWORD cbOutput,
    __out     DWORD *pcbResult,
    __in      ULONG dwFlags
);
```

Параметры:

- `hKey [in]`. Дескриптор ключа для подписи хэша.
- `pPaddingInfo [in, optional]`. На данный момент не используется и игнорируется.

- `pbInput [in]`. Указатель на буфер, который содержит значение хэша для подписи. Параметр `cbInput` содержит размер этого буфера.
- `cbInput [in]`. Размер в байтах содержимого буфера `pbInput`, предназначенного для подписания.
- `pbOutput [out]`. Адрес буфера, предназначенного для получения подписи от функции. Параметр `cbOutput` содержит размер этого буфера.

  Если этот параметр принимает значение `NULL`, рассматриваемая функция рассчитывает необходимый размер в байтах для подписи и возвращает размер в расположении, указанном в параметре `pcbResult`.
- `cbOutput [in]`. Размер в байтах буфера `pbOutput`. Этот параметр игнорируется, если параметр `pbOutput` имеет значение `NULL`.
- `pcbResult [out]`. Указатель на переменную типа `ULONG`, который получает число байтов информации, скопированное в буфер `pbOutput`.

  Если параметр `pbOutput` имеет значение `NULL`, описываемый параметр получает размер в байтах, необходимый для подписи.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. На данный момент не используется и должен быть равен нулю.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 28. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_INVALID_HANDLE`	В параметре `hKey` задан недопустимый дескриптор.
`STATUS_NOT_SUPPORTED`	Провайдер алгоритма, используемый для создания дескриптора ключа, определенного параметром `hKey`, — не алгоритм подписи.
`STATUS_NO_MEMORY`	Обнаружен сбой выделения памяти.
`STATUS_BUFFER_TOO_SMALL`	Размер, определенный в параметре `cbOutput`, недостаточен для подписи.

Внимание! Результатом выполнения данной функции является подпись, в которой инвертирован порядок байтов относительно результата работы аналогичной функции в программе ViPNet CSP. Для использования данной подписи в ViPNet CSP необходимо ее инвертировать (то есть первый байт становится последним, второй — предпоследним и так далее).

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Функция BCryptVerifySignature

Функция `BCryptVerifySignature` предназначена для проверки соответствия подписи и хэша.

```
NTSTATUS WINAPI BCryptVerifySignature(
    __in      BCRYPT_KEY_HANDLE hKey,
    __in_opt  VOID *pPaddingInfo,
    __in      PUCHAR pbHash,
    __in      ULONG cbHash,
    __in      PUCHAR pbSignature,
    __in      ULONG cbSignature,
    __in      ULONG dwFlags
);
```

Параметры:

- `hKey [in]`. Дескриптор ключа для проверки подписи. Это должен быть соответствующий ключ или часть открытого ключа из ключевой пары, использованной для подписания данных функцией `BCryptSignHash`.
- `pPaddingInfo [in, optional]`. Параметр на данный момент не используется и должен быть равен нулю.
- `pbHash [in]`. Адрес буфера, который содержит хэш данных. Параметр `cbOutput` содержит размер этого буфера.
- `cbHash [in]`. Размер в байтах буфера `pbHash`.
- `pbSignature [in]`. Адрес буфера, который содержит подписанный хэш данных. Функция `BCryptSignHash` используется для создания подписи. Параметр `cbSignature` содержит размер этого буфера.
- `cbSignature [in]`. Размер в байтах буфера `pbSignature`. Функция `BCryptSignHash` используется для создания подписи.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. На данный момент не используется и должен быть равен нулю.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 29. Возвращаемые значения

Возвращаемое значение	Описание
`STATUS_SUCCESS`	Функция выполнена успешно.
`STATUS_INVALID_SIGNATURE`	Подпись не проверена.
`NTE_NO_MEMORY`	Обнаружен сбой выделения памяти.
`STATUS_INVALID_HANDLE`	В параметре `hKey` задан недопустимый дескриптор.
`STATUS_NOT_SUPPORTED`	Провайдер алгоритма, используемый для создания дескриптора ключа, определенного параметром `hKey`, — не алгоритм подписи.

Внимание! Если необходимо проверить подпись, сформированную в программе ViPNet CSP, ее надо инвертировать (то есть первый байт становится последним, второй — предпоследним и так далее).

О потокобезопасности вызова функции см. в разделе Потокобезопасность вызова функций реализации ViPNet CNG (на стр. 52).

Примечание. Функция может быть вызвана как на пользовательском уровне, так и на уровне ядра. На уровне ядра функция реализована только для режима `PASSIVE_LEVEL IRQL`.

Потокобезопасность вызова функций реализации ViPNet CNG

Примечание. Далее в тексте операции, которые можно выполнять параллельно в любом количестве потоков, называются «полностью потокобезопасными».

Параллельно могут выполняться любые полностью потокобезопасные операции.

Вызов функций реализации ViPNet CNG на уровне ядра в режиме `DISPATCH_LEVEL IRQL` не реализован.

Вызов функций реализации ViPNet CNG на уровне ядра в режиме `PASSIVE_LEVEL IRQL` имеет такую же потокобезопасность, как и на пользовательском уровне, поэтому приведенная ниже информация справедлива как для вызова на пользовательском уровне, так и для вызова на уровне ядра в режиме `PASSIVE_LEVEL IRQL`.

Все функции над алгоритм-провайдерами и субобъектами алгоритм-провайдеров являются полностью потокобезопасными, поэтому ниже описана потокобезопасность операций только над одним объектом.

Вызов функций `BCryptOpenAlgorithmProvider` (см. «Функция BCryptOpenAlgorithmProvider» на стр. 46) и `BCryptCloseAlgorithmProvider` (см. «Функция BCryptCloseAlgorithmProvider» на стр. 19) полностью потокобезопасен для всех алгоритмов реализации ViPNet CNG (см. раздел Список алгоритмов, добавленных для поддержки ГОСТ (на стр. 7)). На алгоритм-провайдере в подсистеме Microsoft CNG реализован потокобезопасный подсчет ссылок, поэтому если хотя бы один субобъект алгоритм-провайдера не удален, вызов функции `BCryptCloseAlgorithmProvider` для него не вызывает удаление алгоритм-провайдера, а всего лишь уменьшается количество ссылок на него. Реальное же его уничтожение происходит во время удаления последнего субобъекта алгоритм-провайдера.

Функции уничтожения субобъектов алгоритм-провайдеров не являются потокобезопасными над одним и тем же объектом. Перед их вызовом надо быть уверенным, что работа с этими объектами прекращена во всех потоках.

К функциям уничтожения субобъектов алгоритм-провайдеров относятся:

- `BCryptDestroyHash` (см. «Функция BCryptDestroyHash» на стр. 25);
- `BCryptDestroyKey` (см. «Функция BCryptDestroyKey» на стр. 25).

Вызов функций `BCryptSetProperty` (см. «Функция BCryptSetProperty» на стр. 47) не является потокобезопасным ни для алгоритм-провайдеров, ни для субобъектов алгоритм-провайдеров. Предполагается следующее:

- После создания алгоритм-провайдера или субобъекта алгоритм-провайдера его свойства не выставляются одновременно в разных потоках.

- После выставления свойств дескриптор алгоритм-провайдера или субобъекта алгоритм-провайдера передается в разные потоки для работы, и его свойства более не будут изменяться.

К функциям создания субобъектов на алгоритм-провайдере относятся:

- `BCryptCreateHash` (см. «Функция BCryptCreateHash» на стр. 19);
- `BCryptGenerateSymmetricKey` (см. «Функция BCryptGenerateSymmetricKey» на стр. 37);
- `BCryptGenerateKeyPair` (см. «Функция BCryptGenerateKeyPair» на стр. 36);
- `BCryptImportKey` (см. «Функция BCryptImportKey» на стр. 42);
- `BCryptImportKeyPair` (см. «Функция BCryptImportKeyPair» на стр. 44).

Эти функции являются полностью потокобезопасными как на одном объекте алгоритм-провайдера, так и на разных объектах.

Для алгоритмов, поддерживающих интерфейс электронной подписи, следующие функции работы с объектами закрытых ключей и объектами открытых ключей являют полностью потокобезопасными как на одном объекте, так и на разных объектах:

- `BCryptSignHash` (см. «Функция BCryptSignHash» на стр. 48);
- `BCryptVerifySignature` (см. «Функция BCryptVerifySignature» на стр. 50);
- `BCryptExportKey` (см. «Функция BCryptExportKey» на стр. 31);
- `BCryptGetProperty` (см. «Функция BCryptGetProperty» на стр. 39).

Для объекта закрытого ключа функция `BCryptExportKey` (см. «Функция BCryptExportKey» на стр. 31) не является потокобезопасной над одним и тем же объектом, так как в ней происходит операция перемаскирования закрытого ключа, которая не является потокобезопасной. Для объекта открытого ключа данная функция является полностью потокобезопасной как на одном объекте, так и на разных объектах.

Объект закрытого ключа в интервале между вызовами `BCryptGenerateKeyPair` (см. «Функция BCryptGenerateKeyPair» на стр. 36) и `BCryptFinalizeKeyPair` (см. «Функция BCryptFinalizeKeyPair» на стр. 34) не является потокобезопасным.

Функции работы с симметричными ключами, с объектами хэша и MAC-объектами не являются потокобезопасными на одном и том же объекте, так как данные объекты хранят состояние, и их параллельное применение в разных потоках не имеет смысла.

К таким функциям относятся:

- `BCryptHashData` (см. «Функция BCryptHashData» на стр. 41);
- `BCryptFinishHash` (см. «Функция BCryptFinishHash» на стр. 35);
- `BCryptDuplicateHash` (см. «Функция BCryptDuplicateHash» на стр. 26);
- `BCryptEncrypt` (см. «Функция BCryptEncrypt» на стр. 29);
- `BCryptDecrypt` (см. «Функция BCryptDecrypt» на стр. 22);
- `BCryptDuplicateKey` (см. «Функция BCryptDuplicateKey» на стр. 27);

- `BCryptExportKey` (см. «Функция BCryptExportKey» на стр. 31);
- `BCryptGetProperty` (см. «Функция BCryptGetProperty» на стр. 39);
- `BCryptSetProperty` (см. «Функция BCryptSetProperty» на стр. 47).

Функции экспорта и импорта ключей на одном и том же объекте симметричного ключа защиты не являются потокобезопасными.

К таким функциям относятся:

- `BCryptExportKey` (см. «Функция BCryptExportKey» на стр. 31);
- `BCryptImportKey` (см. «Функция BCryptImportKey» на стр. 42);
- `BCryptImportKeyPair` (см. «Функция BCryptImportKeyPair» на стр. 44).

2

Свойства и функции хранилища ключей

Свойства объектов хранилища ключей

Внимание! Свойства и функции хранилища ключей (библиотека CNG NCrypt) не поддерживаются в программном обеспечении ViPNet CSP Linux.

Таблица 30. Список стандартных свойств объектов ключевого хранилища

Идентификатор и значение	Описание
`NCRYPT_ALGORITHM_GROUP_PROPERTY L"Algorithm Group"`	Не поддерживается.
`NCRYPT_ALGORITHM_PROPERTY L"Algorithm Name"`	Оканчивающаяся нулем Юникод-строка, содержащая имя алгоритма объекта. Это может быть один из предварительно заданных идентификаторов CNG или идентификатор другого зарегистрированного алгоритма. Это свойство применяется только к ключам.
`NCRYPT_ASSOCIATED_ECDH_KEY L"SmartCardAssociatedECDHKey"`	Не поддерживается.
`NCRYPT_BLOCK_LENGTH_PROPERTY L"Block Length"`	Не поддерживается.
`NCRYPT_CERTIFICATE_PROPERTY L"SmartCardKeyCertificate"`	Блоб, содержащий сертификат ключа смарт-карты.
`NCRYPT_DH_PARAMETERS_PROPERTY L"DHParameters"`	Не поддерживается.
`NCRYPT_EXPORT_POLICY_PROPERTY L"Export Policy"`	Выставляемый параметр сохраняется и возвращается по запросу, но фактически на данный момент не используется.
`NCRYPT_IMPL_TYPE_PROPERTY L"Impl Type"`	Объект типа `DWORD`, содержащий набор флагов, которые определяют подробности реализации провайдера. Это свойство применяется только к провайдерам хранилища ключей. Возвращает значение `NCRYPT_IMPL_SOFTWARE_FLAG`.
`NCRYPT_KEY_TYPE_PROPERTY L"Key Type"`	На данный момент не поддерживается.
`NCRYPT_KEY_USAGE_PROPERTY L"Key Usage"`	Объект типа `DWORD`, содержащий набор флагов, которые определяют подробности использования ключа. Это свойство применяется только к ключам.

Идентификатор и значение	Описание
NCRYPT_LAST_MODIFIED_PROPERTY L"Modified"	Не поддерживается.
NCRYPT_LENGTH_PROPERTY L"Length"	Объект типа DWORD, содержащий длину ключа в битах. Это свойство применяется только к ключам.
NCRYPT_LENGTHS_PROPERTY L"Lengths"	Показывает размеры ключей, поддерживаемых данным ключом. Тип данных содержится в структуре NCRYPT_SUPPORTED_LENGTHS. Это свойство применяется только к ключам.
NCRYPT_MAX_NAME_LENGTH_PROPERTY L"Max Name Length"	Объект типа DWORD, содержащий максимальное количество символов имени постоянного ключа. Это свойство применяется только к провайдерам. Свойство в первую очередь предназначено для использования провайдерами хранилища ключей, которые хранят ключи на устройстве с ограниченным объемом памяти, например на смарт-карте.
NCRYPT_NAME_PROPERTY L"Name"	Указатель на оканчивающуюся нулем Юникод-строку, содержащую имя объекта.
NCRYPT_PIN_PROMPT_PROPERTY L"SmartCardPinPrompt"	Не поддерживается.
NCRYPT_PIN_PROPERTY L"SmartCardPin"	Не поддерживается.
NCRYPT_PROVIDER_HANDLE_PROPERTY L"Provider Handle"	Не поддерживается.
NCRYPT_READER_PROPERTY L"SmartCardReader"	Не поддерживается.
NCRYPT_ROOT_CERTSTORE_PROPERTY L"SmartcardRootCertStore"	Не поддерживается.
NCRYPT_SCARD_PIN_ID L"SmartCardPinId"	Не поддерживается.
NCRYPT_SCARD_PIN_INFO L"SmartCardPinInfo"	Не поддерживается.
NCRYPT_SECURE_PIN_PROPERTY L"SmartCardSecurePin"	Не поддерживается.

Идентификатор и значение	Описание
NCRYPT_SECURITY_DESCR_PROPERTY L"Security Descr"	Указатель на структуру SECURITY_DESCRIPTOR, содержащую информацию о контроле доступа для ключа. Это свойство применяется только к постоянным ключам. Параметр dwFlagsr функции NCryptGetProperty или NCryptSetProperty, показывающий часть дескриптора безопасности, которую необходимо получить или настроить. На данный момент сохраняется и возвращает полностью построенный дескриптор, но его проверка не осуществляется.
NCRYPT_SECURITY_DESCR_SUPPORT_PROPERTY L"Security Descr Support"	Не поддерживается.
NCRYPT_SMARTCARD_GUID_PROPERTY L"SmartCardGuid"	Не поддерживается.
NCRYPT_UI_POLICY_PROPERTY L"UI Policy"	При использовании с функциями NCryptSetProperty или NCryptGetProperty является указателем на структуру NCRYPT_UI_POLICY, содержащую политику сильной защиты ключа пользователя. Это свойство применяется только к постоянным ключам. Свойство может быть задано только при создании ключа. После того как функция NCryptFinalizeKey вызывается для ключа, это свойство становится доступно только для чтения. Выставляемый параметр сохраняется и возвращается по запросу, но фактически на данный момент не используется.
NCRYPT_UNIQUE_NAME_PROPERTY L"Unique Name"	Не поддерживается.
NCRYPT_USE_CONTEXT_PROPERTY L"Use Context"	Указатель на оканчивающуюся нулем Юникод-строку, описывающую контекст операции. Это свойство не постоянно и может быть задано для провайдера или ключа. У ключа нет права доступа к свойству NCRYPT_USE_CONTEXT_PROPERTY провайдера, потому что это свойство специфично для дескриптора, для которого задано.
NCRYPT_USE_COUNT_ENABLED_PROPERTY L"Enabled Use Count"	Не поддерживается.
NCRYPT_USE_COUNT_PROPERTY L"Use Count"	Не поддерживается.
NCRYPT_USER_CERTSTORE_PROPERTY L"SmartCardUserCertStore"	Не поддерживается.

Идентификатор и значение	Описание
`NCRYPT_VERSION_PROPERTY` `L"Version"`	Объект типа `DWORD`, содержащий версию ПО провайдера. Старшее слово содержит основную версию, а младшее — вспомогательную версию. Это свойство применяется только к провайдерам.
`NCRYPT_WINDOW_HANDLE_PROPERTY` `L"HWND Handle"`	Объект типа `DWORD`, содержащий дескриптор окна (HWND), предназначенный для использования в качестве родительского объекта для любого отображаемого пользовательского интерфейса.

Таблица 31. Список свойств объектов ключевого хранилища, специфичных для реализации ViPNet CNG

Идентификатор и значение	Описание
`NCRYPT_CIPHEROID` `"Cipher OID"`	Параметр ключа, содержащий OID-идентификатор подписи (аналог параметра `KP_CIPHEROID`)
`NCRYPT_HASH_OID` `"HashOID"`	Установка OID идентификатора алгоритма хэширования. Соответствует параметру `KP_HASHOID` из CAPI
`NCRYPT_DH_OID` `"DHOID"`	Установка OID идентификатора алгоритма DH. Соответствует параметру `KP_DHOID` из CAPI
`NCRYPT_KEY_CP_PROV_HANDLE` `"CP Provider Handle"`	Используется только для внутренних нужд компании «ИнфоТеКС».
`NCRYPT_KEY_CP_KEY_SPEC` `"CP KeySpec"`	Используется только для внутренних нужд компании «ИнфоТеКС».

Функции хранилища ключей

Данный раздел содержит описание функций ключевого хранилища интерфейса ViPNet CNG, которые имеют специфику для использования алгоритмов ГОСТ.

Список поддерживаемых функций полностью соответствующих стандартному описанию:

- `NCryptEnumStorageProviders`;
- `NCryptEnumKeys`;
- `NCryptFreeBuffer`;
- `NCryptFreeObject`;
- `NCryptIsKeyHandle`;
- `NCryptTranslateHandle`;
- `NCryptDeleteKey`;
- `NCryptOpenKey`.

Описание указанных функций вы можете найти на сайте MSDN (http://msdn.microsoft.com/en-us/library/windows/desktop/aa376210(v=vs.85).aspx).

Список функций, которые не поддерживаются в интерфейсе ViPNet CNG:

- `NCryptDeriveKey`;
- `NCryptEncrypt`;
- `NCryptNotifyChangeKey`;
- `NCryptSecretAgreement`;
- `NCryptVerifySignature`.

Функция NCryptCreatePersistedKey

Функция `NCryptCreatePersistedKey` предназначена для генерации новых ключей и хранения их в заданном провайдере хранилищ ключей.

```
SECURITY_STATUS WINAPI NCryptCreatePersistedKey(
    __in      NCRYPT_PROV_HANDLE hProvider,
    __out     NCRYPT_KEY_HANDLE *phKey,
    __in      LPCWSTR pszAlgId,
    __in_opt  LPCWSTR pszKeyName,
    __in      DWORD dwLegacyKeySpec,
    __in      DWORD dwFlags
);
```

Параметры:

- `hProvider [in]`. Дескриптор провайдера хранилища ключей, в котором необходимо создать ключ. Этот дескриптор можно получить с помощью функции `NCryptOpenStorageProvider`.
- `phKey [out]`. Адрес переменной `NCRYPT_KEY_HANDLE`, которая получает дескриптор ключа. Когда дескриптор больше не требуется, его необходимо передать функции `NCryptFreeObject`.
- `pszAlgId [in]`. Указатель на оканчивающуюся нулем Юникод-строку, содержащую идентификатор криптографического алгоритма для создания ключа. Это может быть один из стандартных идентификаторов алгоритма CNG или идентификатор другого зарегистрированного алгоритма.

  В систему добавлены идентификаторы поддержки алгоритмов ГОСТ, описанных в Список алгоритмов, добавленных для поддержки ГОСТ (на стр. 7), которые указываются здесь для открытия соответствующего интерфейса.
- `pszKeyName [in, optional]`. Указатель на оканчивающуюся нулем Юникод-строку, содержащую имя ключа. Если параметр имеет значение `NULL`, эта функция создаст временный ключ, который не сохраняется.
- `dwLegacyKeySpec [in]`. Устаревший идентификатор, задающий тип ключа. Параметр может принимать одно из следующих значений:

  Таблица 32. Значения параметра `dwLegacyKeySpec`

Значение	Описание
`AT_KEYEXCHANGE`	Ключ является ключом обмена.
`AT_SIGNATURE`	Ключ является ключом подписи.
`0`	Тип ключа отличается от приведенных выше типов.

- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. Значением параметра может быть 0 или любое из перечисленных ниже значений и их комбинаций.

  Таблица 33. Значения параметра `dwFlags`

Значение	Описание
`NCRYPT_MACHINE_KEY_FLAG`	Ключ применяется к хранилищу локального компьютера. Если флаг отсутствует, ключ применяется к пользовательскому хранилищу.
`NCRYPT_OVERWRITE_KEY_FLAG`	Если в контейнере с указанным именем ключ уже существует, он будет переписан. Если флаг не задан, а ключ с заданным именем уже существует, функция возвращает `NTE_EXISTS`.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 34. Возвращаемые значения

Возвращаемое значение	Описание
ERROR_SUCCESS	Функция выполнена успешно.
NTE_BAD_FLAGS	Параметр dwFlags содержит недопустимое значение.
NTE_EXISTS	Ключ с заданным именем уже существует, а NCRYPT_OVERWRITE_KEY_FLAG не был задан.
NTE_INVALID_HANDLE	Недопустимый параметр hProvider.
NTE_INVALID_PARAMETER	Один или несколько недопустимых параметров.
NTE_NO_MEMORY	Обнаружен сбой выделения памяти.

Функция NCryptDecrypt

Функция NCryptDecrypt предназначена для расшифрования блоков зашифрованных данных.

```
SECURITY_STATUS WINAPI NCryptDecrypt(
    __in      NCRYPT_KEY_HANDLE hKey,
    __in      PBYTE pbInput,
    __in      DWORD cbInput,
    __in_opt  VOID *pPaddingInfo,
    __out     PBYTE pbOutput,
    __in      DWORD cbOutput,
    __out     DWORD *pcbResult,
    __in      DWORD dwFlags
);
```

Параметры:

- hKey [in]. Дескриптор ключа для расшифрования данных.
- pbInput [in]. Адрес буфера с данными, предназначенными для расшифрования. Параметр cbInput содержит размер этих данных.
- cbInput [in]. Размер в байтах содержимого буфера pbInput, предназначенного для расшифрования.
- pPaddingInfo [in, optional] — на данный момент не используется, передаваемое значение игнорируется.
- pbOutput [out]. Адрес буфера, предназначенного для получения расшифрованных данных от функции. Параметр cbOutput содержит размер этого буфера.

  Если этот параметр принимает значение NULL, рассматриваемая функция рассчитывает необходимый размер в байтах для расшифрованных данных и возвращает размер в расположении, указанном в параметре pcbResult.
- cbOutput [in]. Размер в байтах буфера pbOutput. Этот параметр игнорируется, если параметр pbOutput имеет значение NULL.

- `pcbResult [out]`. Указатель на переменную типа `DWORD`, который получает число байтов информации, скопированное в буфер `pbOutput`. Если параметр `pbOutput` имеет значение `NULL`, описываемый параметр получает размер в байтах, необходимый для расшифрованных данных.
- `dwFlags [in]` — на данный момент не используется, передаваемое значение игнорируется.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 35. Возвращаемые значения

Возвращаемое значение	Описание
`ERROR_SUCCESS`	Функция выполнена успешно.
`NTE_BAD_FLAGS`	Параметр `dwFlags` содержит недопустимое значение.
`NTE_BUFFER_TOO_SMALL`	Размер, определенный в параметре `cbOutput`, недостаточен для расшифрованных данных.
`NTE_INVALID_HANDLE`	Недопустимый параметр `hKey`.
`NTE_INVALID_PARAMETER`	Один или несколько недопустимых параметров.
`NTE_PERM`	Ключ с идентификатором `hKey` нельзя использовать для расшифрования.

Функция NCryptEnumAlgorithms

Функция `NCryptEnumAlgorithms` получает имена алгоритмов, поддерживаемых заданным провайдером хранилищ ключей.

```
SECURITY_STATUS WINAPI NCryptEnumAlgorithms(
    __in   NCRYPT_PROV_HANDLE hProvider,
    __in   DWORD dwAlgOperations,
    __out  DWORD *pdwAlgCount,
    __out  NCryptAlgorithmName **ppAlgList,
    __in   DWORD dwFlags
);
```

Параметры:

- `hProvider [in]`. Дескриптор провайдера хранилища ключей для перечисления алгоритмов. Чтобы получить этот дескриптор, вызовите функцию `NCryptOpenStorageProvider`.
- `dwAlgOperations [in]`. Набор значений, определяющих классы алгоритмов для перечисления. Значением параметра может быть любое из перечисленных ниже значений и их комбинаций.

Таблица 36. Значения параметра `dwAlgOperations`

Значение	Описание
`NCRYPT_CIPHER_OPERATION` `0x00000001`	Интерфейс поддерживается. Имеется алгоритм ГОСТ.
`NCRYPT_HASH_OPERATION` `0x00000002`	Интерфейс поддерживается. Имеется алгоритм ГОСТ.
`NCRYPT_ASYMMETRIC_ENCRYPTION_OPERATION` `0x00000004`	Интерфейс не поддерживается.
`NCRYPT_SECRET_AGREEMENT_OPERATION` `0x00000008`	Интерфейс не поддерживается.
`NCRYPT_SIGNATURE_OPERATION` `0x00000010`	Интерфейс поддерживается. Имеется алгоритм ГОСТ.
`NCRYPT_RNG_OPERATION` `0x00000020`	Интерфейс не поддерживается.

- `pdwAlgCount [out]`. Адрес объекта типа `DWORD`, который получает число элементов массива `ppAlgList`.
- `ppAlgList [out]`. Адрес указателя на структуру `NCryptAlgorithmName`, получающую массив имен зарегистрированных алгоритмов. Переменная, на которую указывает параметр `pdwAlgCount`, получает число элементов в этом массиве.

  Когда этот объем памяти больше не нужен, его необходимо освободить, передав этот указатель функции `NCryptFreeBuffer`.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. Параметр может принимать значение 0 или одно из следующих значений:

  Таблица 37. Значения параметра `dwFlags`

Значение	Описание
`NCRYPT_SILENT_FLAG`	Не отображать пользовательский интерфейс. Этот флаг доступен не для всех типов ключей.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 38. Возвращаемые значения

Возвращаемое значение	Описание
`ERROR_SUCCESS`	Функция выполнена успешно.
`NTE_BAD_FLAGS`	Параметр `dwFlags` содержит недопустимое значение.
`NTE_INVALID_HANDLE`	Недопустимый параметр `hProvider`.

Возвращаемое значение	Описание
NTE_INVALID_PARAMETER	Один или несколько недопустимых параметров.
NTE_NO_MEMORY	Обнаружен сбой выделения памяти.

Функция NCryptExportKey

Функция `NCryptExportKey` предназначена для экспорта CNG-ключа в блоб.

```
SECURITY_STATUS WINAPI NCryptExportKey(
    __in       NCRYPT_KEY_HANDLE hKey,
    __in_opt   NCRYPT_KEY_HANDLE hExportKey,
    __in       LPCWSTR pszBlobType,
    __in_opt   NCryptBufferDesc *pParameterList,
    __out_opt  PBYTE pbOutput,
    __in       DWORD cbOutput,
    __out      DWORD *pcbResult,
    __in       DWORD dwFlags
);
```

Параметры:

- `hKey [in]`. Дескриптор ключа, предназначенный для экспорта.
- `hExportKey [in, optional]`. Дескриптор ключа экспорта. Блоб ключа будет содержать `hKey` зашифрованный на данном ключе.
- `pszBlobType [in]`. Оканчивающаяся нулем Юникод-строка, которая содержит идентификатор типа экспортируемого блоба. Параметр может принимать одно из следующих значений:

  Таблица 39. Значения параметра `pszBlobType`

Значение	Описание
BCRYPT_DH_PRIVATE_BLOB	Не поддерживается
BCRYPT_DH_PUBLIC_BLOB	Не поддерживается
BCRYPT_DSA_PRIVATE_BLOB	Не поддерживается
BCRYPT_DSA_PUBLIC_BLOB	Не поддерживается
BCRYPT_ECCPRIVATE_BLOB	Не поддерживается
BCRYPT_ECCPUBLIC_BLOB	Не поддерживается
BCRYPT_KEY_DATA_BLOB	Не поддерживается
BCRYPT_PUBLIC_KEY_BLOB	Экспорт общих открытых ключей любых типов. Тип ключа в блобе определяется элементом `Magic` структуры `BCRYPT_KEY_BLOB`. `PUBLICKEYBLOB` из CAPI.

Значение	Описание
BCRYPT_PRIVATE_KEY_BLOB	Экспорт общих открытых ключей любых типов. Закрытый ключ не обязательно содержит открытый ключ. Тип ключа в блобе определяется элементом Magic структуры BCRYPT_KEY_BLOB. PRIVATEKEYBLOB из CAPI.
BCRYPT_RSAFULLPRIVATE_BLOB	Не поддерживается
BCRYPT_RSAPRIVATE_BLOB	Не поддерживается
BCRYPT_RSAPUBLIC_BLOB	Не поддерживается
LEGACY_DH_PRIVATE_BLOB	Не поддерживается
LEGACY_DH_PUBLIC_BLOB	Не поддерживается
LEGACY_DSA_PRIVATE_BLOB	Не поддерживается
LEGACY_DSA_PUBLIC_BLOB	Не поддерживается
LEGACY_RSAPRIVATE_BLOB	Не поддерживается
LEGACY_RSAPUBLIC_BLOB	Не поддерживается
NCRYPT_OPAQUETRANSPORT_BLOB	Экспорт ключа в формате, специфичном для определенного криптопровайдера и допускающем возможность передачи. Объекты Opaque BLOB передавать невозможно, поэтому их необходимо импортировать с помощью криптопровайдера, сгенерировавшего данный блоб.
NCRYPT_PKCS7_ENVELOPE_BLOB	Не поддерживается
NCRYPT_PKCS8_PRIVATE_KEY_BLOB	Не поддерживается

- pParameterList [in, optional]. Не используется на данный момент, передаваемое значение игнорируется.
- pbOutput [out, optional]. Адрес буфера, который получает блоб ключа. Параметр cbOutput содержит размер этого буфера. Если этот параметр принимает значение NULL, рассматриваемая функция поместит необходимый размер в байтах в переменную типа DWORD, на которую указывает параметр pcbResult.
- cbOutput [in]. Размер в байтах буфера pbOutput.
- pcbResult [out]. Адрес переменной типа DWORD, которая получает число байтов информации, скопированное в буфер pbOutput. Если параметр pbOutput принимает значение NULL, рассматриваемая функция поместит необходимый размер в байтах в объект DWORD, на который указывает этот параметр.
- dwFlags [in]. Набор флагов, изменяющих поведение функции. Для функции не определено ни одного флага.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 40. Возвращаемые значения

Возвращаемое значение	Описание
ERROR_SUCCESS	Функция выполнена успешно.
NTE_BAD_FLAGS	Параметр dwFlags содержит недопустимое значение.
NTE_BAD_KEY_STATE	В параметре hKey задан недопустимый ключ. Чаще всего такая ошибка возникает из-за того, что функция NCryptFinalizeKey еще не закончила работу с ключом.
NTE_BAD_TYPE	Ключ, определенный в параметре hKey, невозможно экспортировать в объект типа блоб, определенный в параметре pszBlobType.
NTE_INVALID_HANDLE	Недопустимый параметр hKey или hExportKey.
NTE_INVALID_PARAMETER	Один или несколько недопустимых параметров.

Функция NCryptFinalizeKey

Функция NCryptFinalizeKey предназначена для завершения работы с ключом из хранилища ключей CNG.

```
SECURITY_STATUS NCryptFinalizeKey(
    __in  NCRYPT_KEY_HANDLE hKey,
    __in  DWORD dwFlags
);
```

Параметры:

- hKey [in]. Дескриптор ключа для завершения. Чтобы получить этот дескриптор, вызовите функцию NCryptCreatePersistedKey.
- dwFlags [in]. На данный момент не используется, передаваемое значение игнорируется.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 41. Возвращаемые значения

Возвращаемое значение	Описание
ERROR_SUCCESS	Функция выполнена успешно.
NTE_BAD_FLAGS	Параметр dwFlags содержит недопустимое значение.
NTE_INVALID_HANDLE	Недопустимый параметр hKey.

Функция NCryptGetProperty

Функция `NCryptGetProperty` возвращает значение именованного свойства для объекта хранилища ключей.

```
SECURITY_STATUS WINAPI NCryptGetProperty(
    __in   NCRYPT_HANDLE hObject,
    __in   LPCWSTR pszProperty,
    __out  PBYTE pbOutput,
    __in   DWORD cbOutput,
    __out  DWORD *pcbResult,
    __in   DWORD dwFlags
);
```

Параметры:

- `hObject [in]`. Дескриптор объекта для получения свойства. Это может быть дескриптор провайдера (`NCRYPT_PROV_HANDLE`) или дескриптор ключа (`NCRYPT_KEY_HANDLE`).
- `pszProperty [in]`. Указатель на оканчивающуюся нулем Юникод-строку, содержащую имя свойства, которое необходимо извлечь. Это может быть один из предварительно заданных идентификаторов свойств хранилищ ключей или идентификатор свойства, заданный пользователем.

  Особенности поддержки системных свойств и специфические свойства для поддержки алгоритмов ГОСТ описаны в разделе Свойства объектов ключевого хранилища (см. «Свойства объектов хранилища ключей» на стр. 56).
- `pbOutput [out]`. Адрес буфера, который получает значение свойства. Параметр `cbOutput` содержит размер этого буфера.

  Для вычисления размера, необходимого для буфера, установите для этого параметра значение `NULL`. Размер в байтах, который необходимо возвратить в расположение, указанное в параметре `pcbResult`.
- `cbOutput [in]`. Размер в байтах буфера `pbOutput`.
- `pcbResult [out]`. Указатель на переменную типа `DWORD`, которая получает число байтов информации, скопированное в буфер `pbOutput`.

  Если параметр `pbOutput` имеет значение `NULL`, то размер в байтах, необходимый для буфера, помещается в расположение, указанное в этом параметре.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. Может иметь значение 0. Значение `NCRYPT_PERSIST_ONLY_FLAG` не поддерживается.

  Для свойства `NCRYPT_SECURITY_DESCR_PROPERTY` этот параметр должен также содержать одно из следующих значений, указывающих часть дескриптора безопасности, которую необходимо получить.

Таблица 42. Значения параметра `dwFlags`

Значение	Описание
`OWNER_SECURITY_INFORMATION`	Получение идентификатора безопасности (SID) владельца объекта. Используйте функцию `GetSecurityDescriptorOwner` для получения SID владельца из структуры `SECURITY_DESCRIPTOR`.
`GROUP_SECURITY_INFORMATION`	Получение SID основной группы объекта. Используйте функцию `GetSecurityDescriptorGroup` для получения SID владельца из структуры `SECURITY_DESCRIPTOR`.
`DACL_SECURITY_INFORMATION`	Получение списка управления доступом на уровне пользователей (DACL) Используйте функцию `GetSecurityDescriptorSacl` для получения DACL из структуры `SECURITY_DESCRIPTOR`.
`SACL_SECURITY_INFORMATION`	Получение системного списка управления доступом (SACL). Используйте функцию `GetSecurityDescriptorDacl` для получения SACL из структуры `SECURITY_DESCRIPTOR`.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 43. Возвращаемые значения

Возвращаемое значение	Описание
`ERROR_SUCCESS`	Функция выполнена успешно.
`NTE_BAD_FLAGS`	Параметр `dwFlags` содержит недопустимое значение.
`NTE_INVALID_HANDLE`	Недопустимый параметр `hObject`.
`NTE_INVALID_PARAMETER`	Один или несколько недопустимых параметров.
`NTE_NO_MEMORY`	Обнаружен сбой выделения памяти.
`NTE_NOT_SUPPORTED`	Указанное свойство не поддерживается объектом.

Функция NCryptImportKey

Функция `NCryptImportKey` предназначена для импорта CNG-ключа из блоба.

```
SECURITY_STATUS WINAPI NCryptImportKey(
    __in      NCRYPT_PROV_HANDLE hProvider,
    __in_opt  NCRYPT_KEY_HANDLE hImportKey,
    __in      LPCWSTR pszBlobType,
    __in_opt  NCryptBufferDesc *pParameterList,
    __out     NCRYPT_KEY_HANDLE *phKey,
    __in      PBYTE pbData,
```

```
    __in      DWORD cbData,
    __in      DWORD dwFlags
);
```

Параметры:

- `hProvider [in]`. Дескриптор провайдера хранилища ключей.
- `hImportKey [in, optional]`. Дескриптор ключа импорта. На этом ключе будет расшифрован ключ из исходного блоба. Это должен быть дескриптор того же ключа, что и передавался в параметре `hExportKey` функции `NCryptExportKey`. Если параметр имеет значение `NULL`, предполагается, что блоб ключа не расшифрован.
- `pszBlobType [in]`. Оканчивающаяся нулем Юникод-строка, которая содержит идентификатор, определяющий формат блоба ключа. Параметр может принимать одно из следующих значений:

Таблица 44. Значения параметра `pszBlobType`

Значение	Описание
`BCRYPT_DH_PRIVATE_BLOB`	Не поддерживается
`BCRYPT_DH_PUBLIC_BLOB`	Не поддерживается
`BCRYPT_DSA_PRIVATE_BLOB`	Не поддерживается
`BCRYPT_DSA_PUBLIC_BLOB`	Не поддерживается
`BCRYPT_ECCPRIVATE_BLOB`	Не поддерживается
`BCRYPT_ECCPUBLIC_BLOB`	Не поддерживается
`BCRYPT_KEY_DATA_BLOB`	Не поддерживается
`BCRYPT_PUBLIC_KEY_BLOB`	Этот блоб — общий открытый ключ любого типа. Тип ключа в блобе определяется элементом Magic структуры `BCRYPT_KEY_BLOB`.
`BCRYPT_PRIVATE_KEY_BLOB`	Этот блоб — общий закрытый ключ любого типа. Закрытый ключ не обязательно содержит открытый ключ. Тип ключа в блобе определяется элементом Magic структуры `BCRYPT_KEY_BLOB`. `PRIVATEKEYBLOB` из CAPI.
`BCRYPT_RSAPRIVATE_BLOB`	Не поддерживается
`BCRYPT_RSAPUBLIC_BLOB`	Не поддерживается
`LEGACY_DH_PRIVATE_BLOB`	Не поддерживается
`LEGACY_DH_PUBLIC_BLOB`	Не поддерживается
`LEGACY_DSA_PRIVATE_BLOB`	Не поддерживается
`LEGACY_DSA_PUBLIC_BLOB`	Не поддерживается
`LEGACY_DSA_V2_PRIVATE_BLOB`	Не поддерживается
`LEGACY_DSA_V2_PUBLIC_BLOB`	Не поддерживается

Значение	Описание
LEGACY_RSAPRIVATE_BLOB	Не поддерживается
LEGACY_RSAPUBLIC_BLOB	Не поддерживается
NCRYPT_OPAQUETRANSPORT_BLOB	Блоб — это ключ в формате, специфичном для определенного криптопровайдера и допускающем возможность передачи. Объекты Opaque BLOB передавать невозможно, поэтому их необходимо импортировать с помощью криптопровайдера, сгенерировавшего данный блоб. Аналог OPAQUEKEYBLOB из CAPI.
NCRYPT_PKCS7_ENVELOPE_BLOB	Не поддерживается
NCRYPT_PKCS8_PRIVATE_KEY_BLOB	Не поддерживается

- pParameterList [in, optional] — на данный момент не используется, передаваемое значение игнорируется.
- phKey [out]. Адрес переменной NCRYPT_KEY_HANDLE, которая получает дескриптор ключа. Когда дескриптор больше не требуется, его необходимо передать функции NCryptFreeObject.
- pbData [in]. Адрес буфера, который содержит импортируемый блоб ключа. Параметр cbOutput содержит размер этого буфера.
- cbData [in]. Размер в байтах блоба ключа в буфере pbData.
- dwFlags [in] — на данный момент не используется, передаваемое значение игнорируется.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 45. Возвращаемые значения

Возвращаемое значение	Описание
ERROR_SUCCESS	Функция выполнена успешно.
NTE_BAD_FLAGS	Параметр dwFlags содержит недопустимое значение.
NTE_EXISTS	Ключ с заданным именем уже существует, а NCRYPT_OVERWRITE_KEY_FLAG не был задан.
NTE_INVALID_HANDLE	Недопустимый параметр hProvider.
NTE_INVALID_PARAMETER	Один или несколько недопустимых параметров.
NTE_NO_MEMORY	Обнаружен сбой выделения памяти.

Функция NCryptIsAlgSupported

Функция `NCryptIsAlgSupported` определяет, поддерживает ли провайдер хранилища ключей CNG заданный криптографический алгоритм.

```
SECURITY_STATUS WINAPI NCryptIsAlgSupported(
    __in  NCRYPT_PROV_HANDLE hProvider,
    __in  LPCWSTR pszAlgId,
    __in  DWORD dwFlags
);
```

Параметры:

- `hProvider [in]`. Дескриптор провайдера хранилища ключей. Чтобы получить этот дескриптор, вызовите функцию `NCryptOpenStorageProvider`.
- `pszAlgId [in]`. Указатель на оканчивающуюся нулем Юникод-строку, определяющую запрашиваемый криптографический алгоритм. Это может быть один из стандартных идентификаторов алгоритма CNG или идентификатор другого зарегистрированного алгоритма. В систему добавлены свойства поддержки алгоритмов ГОСТ, описанных в разделе Список алгоритмов, добавленных для поддержки ГОСТ (на стр. 7), которые указываются здесь для открытия соответствующего интерфейса.
- `dwFlags [in]`. Флаги, изменяющие поведение функции. Для функции не определено ни одного флага.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 46. Возвращаемые значения

Возвращаемое значение	Описание
`ERROR_SUCCESS`	Провайдер поддерживает заданный алгоритм.
`NTE_BAD_FLAGS`	Параметр `dwFlags` содержит один или несколько неподдерживаемых флагов.
`NTE_INVALID_HANDLE`	В параметре `hProvider` задан недопустимый дескриптор.
`NTE_INVALID_PARAMETER`	Один или несколько недопустимых параметров.
`NTE_NOT_SUPPORTED`	Провайдер не поддерживает заданный алгоритм.

Функция NCryptOpenStorageProvider

Функция `NCryptOpenStorageProvider` предназначена для загрузки и инициализации провайдера хранилища ключей CNG.

```
SECURITY_STATUS WINAPI NCryptOpenStorageProvider(
    __out     NCRYPT_PROV_HANDLE *phProvider,
```

```
    __in_opt  LPCWSTR pszProviderName,
    __in      DWORD dwFlags
);
```

Параметры:

- `phProvider [out]`. Указатель на переменную `NCRYPT_PROV_HANDLE`, которая получает дескриптор провайдера. Когда дескриптор больше не требуется, его необходимо передать функции `NCryptFreeObject`.
- `pszProviderName [in, optional]`. Указатель на оканчивающуюся нулем Юникод-строку, показывающую провайдер хранилища ключей для загрузки. Это зарегистрированный псевдоним провайдера хранилища ключей. Параметр не является обязательным и может иметь значение `NULL`. Если параметр имеет значение `NULL`, загружается провайдер хранилища ключей по умолчанию. Следующие значения соответствуют встроенным провайдерам хранилищ ключей. Для открытия свойств алгоритмов ГОСТ необходимо указывать `NULL` или имя провайдера ViPNet CNG `"Infotecs Primitive Provider"`.
- `dwFlags [in]`. Флаги, изменяющие поведение функции. Для функции не определено ни одного флага.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 47. Возвращаемые значения

Возвращаемое значение	Описание
`ERROR_SUCCESS`	Функция выполнена успешно.
`NTE_BAD_FLAGS`	Параметр `dwFlags` содержит один или несколько неподдерживаемых флагов.
`NTE_INVALID_PARAMETER`	Один или несколько недопустимых параметров.
`NTE_NO_MEMORY`	Обнаружен сбой выделения памяти.

Функция NCryptSetProperty

Функция `NCryptSetProperty` задает значение именованного свойства для объекта хранилища ключей CNG.

```
SECURITY_STATUS WINAPI NCryptSetProperty(
    __in  NCRYPT_HANDLE hObject,
    __in  LPCWSTR pszProperty,
    __in  PBYTE pbInput,
    __in  DWORD cbInput,
    __in  DWORD dwFlags
);
```

Параметры:

- `hObject [in]`. Дескриптор объекта хранилища ключей, для которого необходимо задать свойство.
- `pszProperty [in]`. Указатель на не оканчивающуюся нулем строку в формате Юникод, содержащую имя свойства, которое необходимо задать. Это может быть один из предварительно заданных идентификаторов свойств хранилищ ключей или идентификатор свойства, заданный пользователем.

  Особенности поддержки системных свойств и специфические свойства для поддержки алгоритмов ГОСТ описаны в разделе Свойства объектов ключевого хранилища (см. «Свойства объектов хранилища ключей» на стр. 56).
- `pbInput [in]`. Адрес буфера, который содержит новое значение свойства. Параметр `cbInput` содержит размер этого буфера.
- `cbInput [in]`. Размер в байтах буфера `pbInput`.
- `dwFlags [in]`. Набор флагов, изменяющих поведение функции. Может иметь значение 0. Значения `NCRYPT_PERSIST_FLAG` и `NCRYPT_PERSIST_ONLY_FLAG` не поддерживаются.

  Для свойства `NCRYPT_SECURITY_DESCR_PROPERTY` этот параметр должен также содержать одно из следующих значений, указывающих часть дескриптора безопасности, которую необходимо задать.

  Таблица 48. Значения параметра `dwFlags`

Значение	Описание
`OWNER_SECURITY_INFORMATION`	Задание идентификатора безопасности (SID) владельца объекта. Используйте функцию `SetSecurityDescriptorOwner` для задания SID владельца в структуре `SECURITY_DESCRIPTOR`.
`GROUP_SECURITY_INFORMATION`	Задание SID основной группы объекта. Используйте функцию `SetSecurityDescriptorGroup` для задания SID группы в структуре `SECURITY_DESCRIPTOR`.
`DACL_SECURITY_INFORMATION`	Задание списка управления доступом на уровне пользователей (DACL). Используйте функцию `SetSecurityDescriptorSacl` для задания DACL в структуре `SECURITY_DESCRIPTOR`.
`SACL_SECURITY_INFORMATION`	Задание системного списка управления доступом (SACL). Используйте функцию `SetSecurityDescriptorDacl` для задания SACL в структуре `SECURITY_DESCRIPTOR`.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 49. Возвращаемые значения

Возвращаемое значение	Описание
ERROR_SUCCESS	Функция выполнена успешно.
NTE_BAD_FLAGS	Параметр dwFlags содержит недопустимое значение.
NTE_INVALID_HANDLE	Недопустимый параметр hObject.
NTE_INVALID_PARAMETER	Один или несколько недопустимых параметров.
NTE_NO_MEMORY	Обнаружен сбой выделения памяти.
NTE_NOT_SUPPORTED	Указанное свойство не поддерживается объектом.

Функция NCryptSignHash

Функция NCryptSignHash предназначена для генерации подписи значения хэша.

```
SECURITY_STATUS WINAPI NCryptSignHash(
    __in      NCRYPT_KEY_HANDLE hKey,
    __in_opt  VOID *pPaddingInfo,
    __in      PBYTE pbHashValue,
    __in      DWORD cbHashValue,
    __out     PBYTE pbSignature,
    __in      DWORD cbSignature,
    __out     DWORD *pcbResult,
    __in      DWORD dwFlags
);
```

Параметры:

- hKey [in]. Дескриптор ключа для подписи хэша.
- pPaddingInfo [in, optional]. Указатель на структуру, содержащую информацию о паддинге. Настоящий тип структуры, на которую указывает этот параметр, зависит от значения параметра dwFlags. Это параметр используется только с асимметричными ключами и в противном случае должен принимать значение NULL. При использовании флагов BCRYPT_PKCS1_PADDING_INFO и BCRYPT_PSS_PADDING_INFO в паддинг-схеме указывается алгоритм хэширования ГОСТ 34.11-94 или ГОСТ 34.11-2012.
- pbHashValue [in]. Указатель на буфер, который содержит значение хэша для подписи. Параметр cbInput содержит размер этого буфера.
- cbHashValue [in]. Размер в байтах содержимого буфера pbHashValue, предназначенного для подписания.
- pbSignature [out]. Адрес буфера, предназначенного для получения подписи от функции. Параметр cbSignature содержит размер этого буфера.

Если этот параметр принимает значение NULL, рассматриваемая функция рассчитывает необходимый размер в байтах для подписи и возвращает размер в расположении, указанном в параметре pcbResult.

- cbSignature [in]. Размер в байтах буфера pbSignature. Этот параметр игнорируется, если параметр pbSignature имеет значение NULL.
- pcbResult [out]. Указатель на переменную типа DWORD, который получает число байтов информации, скопированное в буфер pbSignature.

  Если параметр pbSignature имеет значение NULL, описываемый параметр получает размер в байтах, необходимый для подписи.
- dwFlags [in]. Набор флагов, изменяющих поведение функции. Допустимый набор флагов зависит от типа ключа, определенного в параметре hKey.

  Если ключ симметричный, параметр не используется и должен быть равным нулю.

  Если ключ асимметричный, параметр может принимать одно из следующих значений:

Таблица 50. Значения параметра dwFlags

Значение	Описание
BCRYPT_PAD_PKCS1	Используется паддинг-схема PKCS1. Параметр pPaddingInfo — указатель на структуру BCRYPT_PKCS1_PADDING_INFO.
BCRYPT_PAD_PSS	Используется паддинг-схема PSS. Параметр pPaddingInfo — указатель на структуру BCRYPT_PSS_PADDING_INFO.

Возвращает код состояния, показывающий, успешно ли была выполнена функция. Некоторые из значений, возвращаемых функцией, приведены в таблице:

Таблица 51. Возвращаемые значения

Возвращаемое значение	Описание
ERROR_SUCCESS	Функция выполнена успешно.
NTE_BAD_ALGID	Ключ, представленный в параметре hKey, не поддерживает подпись.
NTE_BAD_FLAGS	Параметр dwFlags содержит недопустимое значение.
NTE_INVALID_HANDLE	Недопустимый параметр hKey.
NTE_INVALID_PARAMETER	Один или несколько недопустимых параметров.
NTE_NO_MEMORY	Обнаружен сбой выделения памяти.