ООО «ВАЛИДАТА»

УТВЕРЖДЕН
ВАМБ.00134-06-ЛУ

ПРОГРАММНЫЙ КОМПЛЕКС «BROWSER PLUG-IN ДЛЯ СКАД СИГНАТУРА»

Руководство программиста

ВАМБ.00134-06 3 01

2020

АННОТАЦИЯ

Данный документ содержит описание библиотеки прикладного программного интерфейса для работы с сертификатами для Web-браузеров Google Chrome, Opera и Mozilla Firefox, входящей в состав СКАД Сигнатура.

При встраивании библиотеки предполагается, что системный программист имеет знания о существующей архитектуре системы сертификатов открытых ключей, используемых рекомендациях и стандартах.

Содержание

1 БИБЛИОТЕКА ПРИКЛАДНОГО ПРОГРАММНОГО ИНТЕРФЕЙСА РАБОТЫ С СЕРТИФИКАТАМИ ДЛЯ БРАУЗЕРОВ GOOGLE CHROME, MOZILLA FIREFOX, OPERA

1.1 Назначение

Библиотека прикладного программного интерфейса работы с сертификатами для браузеров Google Chrome, Mozilla Firefox, Opera (далее – библиотека или библиотека API) является составной частью ВАМБ 00134-06 «Программный комплекс "Browser plug-in для СКАД Сигнатура"» (далее – ПК Web Plugin) и обеспечивает функции аутентификации в соответствии с рекомендациями X.509, электронной подписи (ЭП) по ГОСТ Р 34.10-2001, ГОСТ Р 34.11-94, ГОСТ Р 34.11-2012 и шифрования по ГОСТ 28147-89.

Примечания

1 ПК Web Plugin работает под управлением ОС Microsoft Windows (как 32-битных - x86, так и 64-битных - x64), в зависимости от выбора пользователя при установке ПО с передаточного носителя. Перечень ОС, в которых функционирует ПК Web Plugin, приведен в документе ВАМБ.00134-06 30 01 «Программный комплекс "Browser plug-in для СКАД Сигнатура". Формуляр».

2 При использовании блочных функций ПК Web Plugin обеспечивает выполнение функций зашифрования/расшифрования и создания/проверки ЭП под областью памяти объёмом максимум до 100 Мбайт. При этом данные величины могут быть уменьшены в зависимости от текущего использования виртуальной памяти браузера.

3 При использовании потоковых функций ПК Web Plugin обеспечивает выполнение функций зашифрования/расшифрования и создания/проверки ЭП под областью памяти без ограничения общего объема.

Реализация криптографических функций, доступных посредством библиотеки API, основана на использовании криптографического ядра.

Библиотека обеспечивает обращение к следующим функциям:

- зашифрование и расшифрование области памяти;
- создание ЭП области памяти;
- проверка ЭП области памяти;
- удаление ЭП из области памяти;
- выработка хеш-значения для области памяти;
- преобразование отделенной и совмещенной ЭП;

– создание ЭП хэш-функции данных;
– проверка ЭП хеш-функции данных;

1.2 Характеристики библиотеки

В качестве алгоритма ЭП используется асимметричный вариант ЭП, а именно криптосистема с двумя ключевыми элементами – открытым (общедоступным) и закрытым – согласно ГОСТ Р 34.10-2001 и ГОСТ Р 43.10-2012. Для формирования хэш-функции сообщения используются алгоритмы согласно ГОСТ Р 34.11-94 и ГОСТ Р 34.11-2012. Шифрование информации выполняется согласно ГОСТ 28147-89.

1.3 Использование библиотеки

1.3.1 Условия, необходимые для использования библиотеки

При использовании библиотеки необходимо соблюдение следующих условий:

– библиотека предназначена для использования в web приложениях, написанных с использованием языка программирования JavaScript;
– Web приложения написанные с использованием библиотеки предназначены для использования в браузерах Google Chrome, Mozilla Firefox и Opera с установленным ПК Web Plugin.

1.4 Описание данных

1.4.1 Константы

Заполненные поля сертификата (Прототипа Certificate)

vcert.FIELD_SERIAL = (1 « 0)

Поле с серийным номером.

vcert.FIELD_ISSUER = (1 « 1)

Поле издателя сертификата.

vcert.FIELD_SUBJECT = (1 « 2)

Поле владельца сертификата.

vcert.FIELD_NOTBEFORE = (1 « 5)

Поле с датой начала действия сертификата.

vcert.FIELD_NOTAFTER = (1 « 6)

Поле с датой окончания действия сертификата.

vcert.FIELD_KEYUSAGE = (1 « 7)

Поле с возможной областью применения сертификата.

vcert.FIELD_ISSUERALTNAME = (1 « 8)

Поле с альтернативным именем издателя сертификата.

vcert.FIELD_SUBJECTALTNAME = (1 « 9)

Поле с альтернативным именем владельца сертификата.

vcert.FIELD_EXTKEYUSAGE = (1 « 10)

Поля с расширенной областью использования ключа.

vcert.FIELD_POLICY = (1 « 11)

Поля с политиками использования сертификата.

vcert.FIELD_EXTENSIONS = (1 « 12)

Поля с расширениями сертификата.

vcert.FIELD_NOTBEFOREPRIVATE = (1 « 13)

Поле с датой начала действия закрытого ключа сертификата.

vcert.FIELD_NOTAFTERPRIVATE = (1 « 14)

Поле с датой окончания действия закрытого ключа сертификата.

vcert.FIELD_KEYID = (1 « 15)

Поле с идентификатором ключа, соответствующего сертификату (может использоваться для уникального выбора сертификата).

vcert.FIELD_CERTENCODED = (1 « 16)

Сертификат в DER-кодировке.

vcert.FIELD_CERTHASH = (1 « 17)

Хэш сертификата (может использоваться для уникального выбора сертификата).

vcert.FIELD_ALGORITHM = (1 « 18)

Алгоритм открытого ключа сертификата (может использоваться при работе с несколькими криптографическими системами).

vcert.FIELD_ALL = (FIELD_SERIAL | FIELD_ISSUER | FIELD_SUBJECT | FIELD_ISSUERUID | FIELD_SUBJECTUID | FIELD_NOTBEFORE | FIELD_NOTAFTER | FIELD_KEYUSAGE | FIELD_ISSUERALTNAME | FIELD_SUBJECTALTNAME | FIELD_EXTKEYUSAGE | FIELD_POLICY | FIELD_EXTENSIONS | FIELD_NOTBEFOREPRIVATE | FIELD_NOTAFTERPRIVATE | FIELD_KEYID | FIELD_CERTENCODED | FIELD_CERTHASH | FIELD_ALGORITHM)

Все поля сертификата.

vcert.FIELD_ALGORITHM_OID_FORMAT = (1 « 31)

Возвращать алгоритм открытого ключа сертификата как OID в текстовом виде

Биты возможных областей использования открытого ключа сертификата:

vcert.KEYUSAGE_DIGITAL_SIGNATURE = (1 « 0)

Использование для вычисления ЭП.

vcert.KEYUSAGE_NON_REPUDIATION = (1 « 1)

Неотрекаемость.

vcert.KEYUSAGE_KEY_ENCIPHERMENT = (1 « 2)

Использование для шифрования сессионного ключа.

vcert.KEYUSAGE_DATA_ENCIPHERMENT = (1 « 3)

Использование для шифрования данных.

vcert.KEYUSAGE_KEY_AGREEMENT = (1 « 4)

Использование для выполнения соглашения о ключах (выработки сессионного ключа).

vcert.KEYUSAGE_KEY_CERT_SIGN = (1 « 5)

Использование для подписи сертификатов (установлен в сертификатах Центра Регистрации (ЦР) и Центра Сертификации (ЦС)).

vcert.KEYUSAGE_CRL_SIGN = (1 « 6)

Использование для подписи списков отозванных сертификатов (СОС) (установлен в сертификатах ЦР и ЦС).

vcert.KEYUSAGE_ENCIPHER_ONLY = (1 « 7)

Использование для только для зашифрования данных (при условии выработки сессионного ключа).

vcert.KEYUSAGE_DECIPHER_ONLY = (1 « 8)

Использование для только для расшифрования данных (при условии выработки сессионного ключа).

Общие флаги криптографических функций:

vcert.FLAG_PKCS7 = (1 « 0)

Формат сообщения PKCS#7.

Общие флаги функций вычисления/проверки ЭП (прототипов SignParam и VerifyParam):

vcert.FLAG_DETACHED = (1 « 1)

Сообщение в формате с отсоединенной подписью.

Флаги инициализации (только при использовании локального СКЗИ):

Используются только при работе с локальным СКЗИ в функции vcert.initialize()

vcert.FLAG_INIT_NOCRLUPDATE = (1 « 0)

Не выполнять автоматическое обновление списка отозванных сертификатов (СОС) при инициализации.

vcert.FLAG_INIT_CHECKEXPIRED = (1 « 1)

Показывать объекты с истекающим сроком действия при инициализации. При задании данного флага производится поиск рабочего сертификата с истекающим сроком действия закрытого ключа, и истекающих сертификатов ЦС и СОС. Интервалы истечения объектов задаются в ключе Реестра HKEY_CURRENT_USER\SOTWARE\Validata\zcs. Значение TimeWarnExpiredCert типа REG_DWORD указывает интервал истечения закрытого ключа рабочего сертификата или сертификатов ЦС в днях (по умолчанию – 30 дней). Значение TimeWarnExpiredCrl типа REG_DWORD указывает интервал истечения СОС в днях (по умолчанию – 14 дней).

vcert.FLAG_INIT_NOLDAP = (1 « 2)

Не использовать сетевой справочник при поиске сертификатов.

vcert.FLAG_INIT_NOSAVECACHE = (1 « 5)

Не сбрасывать объекты из кэша в локальный справочник при деинициализации контекста библиотеки.

vcert.FLAG_INIT_REGISTRY = (1 « 6)

Использовать профили Справочника сертификатов из Реестра вместо конфигурационного файла pki1.conf.

vcert.FLAG_INIT_FORCE_CACHE_CERTS = (1 « 7)

Принудительно кэшировать сертификаты, предназначенные для шифрования, при инициализации контекста библиотеки. Данный флаг поддерживается только для локальных справочников, расположенных в хранилище ODBC, при выполнении следующих условий:

- необходимо установить значение параметра **PkiODBCUseMars** (типа REG\DWORD), расположенного в ключе Реестра **HKCU\SOFTWARE\Validata\zpki\Parameters**, в **1**;
- для DSN локального справочника необходимо использовать ODBC драйвер **SQL Server Native Client**;
- для DSN локального справочника необходимо установить значение параметра **Mars_Connection** (типа REG_SZ), расположенного в ключе Реестра **HKCU\SOFTWARE\ODBC\ODBC.INI\<DSN>**, в **Yes**.

vcert.FLAG_INIT_ALLOW_RA_IN_CHAIN = (1 « 8)

Разрешить наличие сертификата и СОС ЦР в качестве промежуточных объектов в цепочке при ее проверке.

vcert.FLAG_INIT_LDAP_CHAIN_SEARCH = (1 « 9)

Разрешить выполнение поиска сертификатов промежуточных ЦС и СОС в сетевом справочнике при построении цепочек.

vcert.FLAG_INIT_USE_AIA_CDP = (1 « 10)

Разрешить доступ к точкам AIA и CDP для загрузки сертификатов промежуточных ЦС и СОС при построении цепочек.

vcert.FLAG_INIT_SILENT_KEYLOAD = (1 « 11)

Не выдавать пользовательский интерфейс при загрузке закрытого ключа.

vcert.FLAG_INIT_VERIFYCONTEXT = (0x80000000)

Загрузка контекста без доступа к закрытому ключу. При этом будут выполняться только функции, не требующие работы с закрытым ключом (хэширования, проверки ЭП).

Флаги функций вычисления ЭП (прототипа SignParam):

Используются флаги FLAG_PKCS7 и FLAG_DETACHED

vcert.FLAG_SIGN_SENDCERT = (1 « 3)

Добавить свой сертификат в подписанное сообщение (при использовании PKCS#7).

Флаги функций проверки ЭП (прототипа VerifyParam):

Используются флаги FLAG_PKCS7 и FLAG_DETACHED

vcert.FLAG_VERIFY_DELSIGN = (1 « 2)

Удалить подпись при проверке ЭП.

vcert.FLAG_VERIFY_KEYUSAGE = (1 « 3)

Проверять использование ключа сертификата.

vcert.FLAG_VERIFY_POLICY = (1 « 4)

Проверять политику сертификата.

vcert.FLAG_VERIFY_EXTKEYUSAGE = (1 « 5)

Проверять расширенное использование ключа.

vcert.FLAG_VERIFY_MINSIGNS = (1 « 7)

Проверять минимальное количество ЭП.

vcert.FLAG_VERIFY_NOTIMECHECK = (1 « 10)

Не проверять времена действия сертификатов и СОС. Данный флаг позволяет избежать проверки сроков действия сертификатов и СОС при построении цепочки сертификата отправителя.

vcert.FLAG_VERIFY_NOCACHECERT = (1 « 11)

Не добавлять автоматически найденные сертификаты из сообщения в справочник.

vcert.FLAG_VERIFY_NOATTACHEDCERT = (1 « 12)

Не использовать сертификаты из сообщения в формате PKCS#7.

vcert.FLAG_VERIFY_USEREVTIME = (1 « 13)

Использовать время отзыва сертификата из COC при проверке цепочки.

vcert.DELETE_ALL_SIGNS = (-1)

Удалить все подписи.

Флаги функций выполнения зашифрования (прототипа EncryptParam):

Используется флаг FLAG_PKCS7

vcert.FLAG_ENCRYPT_REMOTE = (1 « 2)

Искать сертификаты получателей в сетевом справочнике сертификатов.

vcert.FLAG_ENCRYPT_NOTIMECHECK = (1 « 4)

Не проверять времена действия сертификата получателя.

vcert.FLAG_ENCRYPT_NOVERIFY = (1 « 5)

Не проверять действительность сертификата получателя.

vcert.FLAG_ENCRYPT_ADD_LDAP2LOCAL = (1 << 7)

Добавлять сертификат получателя, найденный в сетевом справочнике, в локальный справочник. Данный флаг действует только для поиска сертификатов получателей по уникальному критерию (издателю и серийному номеру, идентификатору ключа или хэшу сертификата).

vcert.FLAG_ENCRYPT_NOKEYTIMECHECK = (1 « 8)

Не проверять времена действия закрытого ключа сертификата получателя.

vcert.FLAG_ENCRYPT_PARTIAL_SUBJECT = (1 « 9)

При установке данного флага поиск сертификатов получателей зашифрованного сообщения производится по части X.500-имени владельца, т.е. поле subject шаблона сертификата считается заполненным частично (например, «OU=7395399001»). Такой поиск поддерживается только для локальных справочников, расположенных в хранилище ODBC, а также для кэша и для сетевых справочников (для сетевых справочников поиск выполняется по атрибуту **vdSubjName**).

Примечание - Данный тип поиска выполняется медленно, поскольку не могут быть использованы индексы локальных справочников, расположенных в хранилище ODBC, и сетевых справочников.

vcert.FLAG_ENCRYPT_IGNORE_CACHED = (1 « 10)

Не выполнять поиск среди кэшированных объектов.

vcert.FLAG_ENCRYPT_IGNORE_LOCAL = (1 « 11)

Не выполнять поиск в локальном справочнике.

vcert.FLAG_ENCRYPT_SUBJECT_ATTRIBUTE = (1 « 12)

При установке данного флага поиск сертификатов получателей по полностью заданному X.500-имени владельца в сетевых справочниках выполняется по атрибуту **vdSubjName**, а не по атрибуту **distinguishedName**. Данный механизм позволяет находить сертификаты получателей в

сетевых справочниках независимо от имен контейнеров, в которых они фактически расположены.

Флаги функций выполнения расшифрования (прототипа DecryptParam):

Используется флаг FLAG_PKCS7

vcert.FLAG_DECRYPT_NOCACHE = (1 « 1)

Не кэшировать найденный сертификат отправителя (только при использовании неанонимного шифрования).

vcert.FLAG_DECRYPT_NOCRLCHECK = (1 « 2)

Не проверять наличие сертификата отправителя в СОС (только при использовании неанонимного шифрования).

Флаги функции поиска сертификатов (прототипа FindParam):

vcert.FLAG_FIND_MY = (1 « 0)

Искать только сертификаты, для которых есть закрытый ключ (используется для получения информации о «своем» сертификате при использовании локального СКЗИ).

vcert.FLAG_FIND_REMOTE = (1 « 2)

Искать также в сетевом справочнике (директории LDAP) (по умолчанию поиск производится только в локальном справочнике).

vcert.FLAG_FIND_SELECTUI = (1 « 3)

Показывать диалог выбора сертификата при нахождении нескольких сертификатов (только при использовании локального СКЗИ).

vcert.FLAG_FIND_NOTIMECHECK = (1 « 4)

Не проверять времена действия сертификата.

vcert.FLAG_FIND_NOVERIFY = (1 « 5)

Не проверять действительность сертификата.

vcert.FLAG_FIND_ADD_LDAP2LOCAL = (1 « 7)

Добавлять сертификат, найденный в сетевом справочнике, в локальный справочник. Данный флаг действует только для поиска сертификатов по уникальному критерию (издателю и серийному номеру, идентификатору ключа или хэшу сертификата).

vcert.FLAG_FIND_NOKEYTIMECHECK = (1 « 8)

Не проверять времена действия закрытого ключа сертификата.

vcert.FLAG_FIND_PARTIAL_SUBJECT = (1 « 9)

При установке данного флага поиск сертификатов производится по части X.500-имени владельца, т.е. поле subject шаблона сертификата считается заполненным частично (например, «OU=7395399001»). Такой поиск поддерживается только для локальных справочников, располо-

женных в хранилище ODBC, а также для кэша и для сетевых справочников (для сетевых справочников поиск выполняется по атрибуту **vdSubjName**).

Примечание - Данный тип поиска выполняется медленно, поскольку не могут быть использованы индексы локальных справочников, расположенных в хранилище ODBC, и сетевых справочников.

vcert.FLAG_FIND_IGNORE_CACHED = (1 « 10)

Не выполнять поиск среди кэшированных объектов.

vcert.FLAG_FIND_IGNORE_LOCAL = (1 « 11)

Не выполнять поиск в локальном справочнике.

vcert.FLAG_FIND_SUBJECT_ATTRIBUTE = (1 « 12)

При установке данного флага поиск сертификатов по полностью заданному X.500-имени владельца в сетевых справочниках выполняется по атрибуту **vdSubjName**, а не по атрибуту **distinguishedName**. Данный механизм позволяет находить сертификаты в сетевых справочниках независимо от имен контейнеров, в которых они фактически расположены.

Флаги функции импорта (прототипа ImportParam):

vcert.FLAG_IMPORT_UI = (1 « 0)

Отображать пользовательский интерфейс показа объекта при импорте.

vcert.FLAG_IMPORT_CERTIFICATE = (1 « 1)

Импортировать сертификат в DER-кодировке.

vcert.FLAG_IMPORT_CRL = (1 « 2)

Импортировать список отозванных сертификатов в DER-кодировке.

vcert.FLAG_IMPORT_MY_CERTIFICATE = (1 « 3)

Импортировать свой новый сертификат (с проверкой доступа к закрытому ключу, соответствующему сертификату) и установить как рабочий.

Примечание - Эта операция не поддерживает параллельное (многопоточное) выполнение. Для использования нового сертификата как рабочего необходимо завершить текущий сеанс работы с библиотекой, вызвав функцию VCERT_Uninitialize.

vcert.FLAG_IMPORT_UPDATE = (1 « 4)

Импортировать обновление от ЦР или ЦС.

Флаги функции экспорта (прототипа ExportParam):

vcert.FLAG_EXPORT_UI = (1 « 0)

Отображать пользовательский интерфейс показа объекта при экспорте.

vcert.FLAG_EXPORT_ASN1 = (1 « 1)

Экспортировать объект в DER-кодировке. По умолчанию объект подписывается на рабочем сертификате в формате PKCS#7.

vcert.FLAG_EXPORT_NEWREQUEST = (1 « 2)

Создать запрос на новый сертификат с генерацией новой ключевой пары в зависимости от того, запрещено или, соответственно, разрешено использование шифрования в текущем рабочем сертификате.

vcert.FLAG_EXPORT_REVREQUEST = (1 « 3)

Создать запрос на отзыв текущего рабочего сертификата.

vcert.FLAG_EXPORT_ETOKEN_GOST = (1 « 5)

Выполнить генерацию закрытого ключа с помощью СКЗИ КриптоПро eToken CSP.

vcert.FLAG_EXPORT_RUTOKEN_GOST = (1 « 6)

Выполнить генерацию закрытого ключа с помощью СКЗИ КриптоПро Rutoken CSP.

vcert.FLAG_EXPORT_GOST_R_34_10_2012 = (1 « 8)

Выполнить генерацию закрытого ключа для квалифицированного сертификата по ГОСТ Р 34.10-2012.

Флаги функции проверки сертификата (прототипа VerifyPolicyParam):

vcert.FLAG_POLICY_NOTIMECHECK = (1 « 2)

Не проверять времена действия сертификата/СОС в цепочке.

vcert.FLAG_POLICY_VERIFY_CRL = (1 « 5)

Выполнять построение цепочки и проверку политики СОС, а не сертификата.

vcert.FLAG_POLICY_NOKEYTIMECHECK = (1 « 6)

Не проверять времена действия закрытого ключа.

Флаги функции создания запроса на получение штампа времени из ЭП формата PKCS#7 (прототипа TSPRequestParam):

vcert.FLAG_TSP_REQUEST_ADD_NONCE = (1 « 1)

Добавлять случайную посылку (NONCE) в запрос на получение штампа времени.

vcert.FLAG_TSP_REQUEST_CERT_REQUEST = (1 « 2)

Требовать включение сертификата авторитетного источника (Time Stamp Authority) в запросе на получение штампа времени.

Флаги функции вычисления ЭП запроса на получение штампа времени (прототипа TSPResponseParam):

vcert.FLAG_TSP_RESPONSE_ADD_TSA_NAME = (1 « 1)

Добавлять имя авторитетного источника (Time Stamp Authority) при вычислении ЭП запроса на получение штампа времени.

Флаги функции проверки ЭП запроса на получение штампа времени (прототипа TSPVerifyParam):

vcert.FLAG_TSP_VERIFY_NOATTACHEDCERT = (1 « 1)

Не использовать сертификат из сообщения для проверки ЭП.

Флаги функции формирования закрытого ключа и XML запроса:

vcert.FLAG_XML_REQUEST_ETOKEN_GOST = (1 « 1)

Использовать для генерации закрытого ключа запроса СКЗИ КриптоПро eToken CSP.

vcert.FLAG_XML_REQUEST_DO_NOT_GENERATE = (1 « 2)

Создать XML запрос без генерации нового закрытого ключа.

vcert.FLAG_XML_REQUEST_RUTOKEN_GOST = (1 « 3)

Использовать для генерации закрытого ключа запроса СКЗИ КриптоПро Rutoken CSP.

vcert.FLAG_XML_REQUEST_GOST_R_34_10_2012 = (1 « 5)

Генерировать закрытый ключ запроса для квалифицированного сертификата по ГОСТ Р 34.10-2012.

Флаги функции создания профиля по заданному пути к справочникам:

vcert.FLAG_CREATE_REGISTRY_PROFILE_OVER = (1 « 1)

Разрешить перезаписывать существующий профиль в Реестре новым путем к справочникам.

1.4.2 Прототипы, уникально идентифицирующие сертификат

function IssuerAndSerial (issuer, serialNumber)

Издатель и серийный номер сертификата.

Состав:

– IssuerAndSerial.prototype.issuer

Издатель (в виде строки Distinguished Name (DN) LDAP, например, CN=Users, DC=x509, DC=ru).

– IssuerAndSerial.prototype.serialNumber

Серийный номер сертификата (в виде hex-строки , например, 40:00:00:01).

1.4.3 Прототипы, описывающие сертификат

function AltName()

Альтернативное имя (издателя или владельца сертификата).

Состав:

– AltName.prototype.emailAddress

RFC 822 Email адрес.

– AltName.prototype.DNS

DNS адрес.

– AltName.prototype.URI

URI адрес.

– AltName.prototype.IP

IP адрес.

– AltName.prototype.organizationName

Наименование организации.

– AltName.prototype.registredAddress

Зарегистрированный адрес.

– AltName.prototype.surname

Ф.И.О владельца сертификата.

– AltName.prototype.businessCategory

Должность.

– AltName.prototype.telephoneNumber

Номер телефона.

– AltName.prototype.description

Описание.

– AltName.prototype.account_number

Номер расчетного счета.

– AltName.prototype.bank_id

Банковский идентификационный код.

– AltName.prototype.physicalDelivery

Почтовый адрес

– AltName.prototype.exchange_address

Адрес Microsoft Exchange.

function ExtKeyUsage(oid)

Расширенное использование ключа сертификата.

Состав:

– ExtKeyUsage.prototype.oid

OID, идентифицирующий применение ключа сертификата.

function Extension(oid,type,critical,data)

Дополнение сертификата.

Состав:

– Extension.prototype.oid

OID, идентифицирующий дополнение.

– Extension.prototype.type

Тип дополнения. Число

– Extension.prototype.critical

Флаг критичности.

– Extension.prototype.data

Base64-строка расширения в DER-кодировке.

function Policy(oid,org_name = null,text = null)

Политика использования сертификата.

Состав:

– Policy.prototype.oid

OID, идентифицирующий политику.

– Policy.prototype.org_name

Название организации.

– Policy.prototype.text

Текст политики использования сертификата.

function Certificate(fields)

Структура сертификата.

Состав:

– Certificate.prototype.fields

Содержит побитовое ИЛИ тех и только тех констант FIELD_XXX, для которых в соответствующих им полях структуры заданы значения (например, при задании FIELD_SUBJECT | FIELD_CERTHASH заполнены только поля subject и certHash).

– Certificate.prototype.issuer

X.500 имя издателя (в виде строки DN LDAP). При задании FIELD_ISSUER в поле fields.

– Certificate.prototype.serialNumber

Серийный номер сертификата (в виде hex-строки). При задании FIELD_SERIALNUMBER в поле fields.

– Certificate.prototype.subject

X.500 имя владельца (в виде строки DN LDAP). При задании FIELD_SUBJECT в поле fields. При заполнении шаблона сертификата для выполнения поиска (см. описание функции vcert.findCert()) имя владельца можно задавать частично или использовать подстановочные знаки-маски «» и «?»:*

- *при выполнении поиска сертификатов в кэше или в локальном справочнике сертификатов, расположенном в хранилище GDBM, допускается подавать строку, заполненную частично с подстановочными знаками-масками «*» и «?» (например, «*,OU=123456789,*»). В этом случае будет производиться перебор всех найденных сертификатов на соответствие имени владельца заданному регулярному выражению;*
- *при выполнении поиска сертификатов в локальном справочнике сертификатов, расположенном в хранилище ODBC, в кэше или в сетевом справочнике сертификатов, допускается подавать строку, заполненную частично (например, «OU=7395399001»). В этом случае, при задании флага поиска FLAG_FIND_PARTIAL_SUBJECT, все сертификаты, находящиеся в справочнике, будут проверяться на соответствие заданному частично имени владельца;*
- *при выполнении поиска сертификатов в сетевом справочнике сертификатов допускается подавать строку, заполненную частично и, возможно, с подстановочным знаком-маской «*» (например, «INN=770200040698» или «INN=7702000406*»). В этом случае будет производиться поиск сертификатов во всех контейнерах директории LDAP, содержащих в своем Relative DN (RDN) заданное частично имя владельца, с учетом, при его наличии, подстановочного знака-маски «*».*

– Certificate.prototype.algorithm

Алгоритм открытого ключа в виде строки. При задании vcert.FIELD_ALGORITHM в поле fields.

– Certificate.prototype.notBefore

Время начала действия сертификата (в миллисекундах, прошедших с 1 января 1970 года GMT+0). При задании vcert.FIELD_NOTBEFORE в поле fields.

– Certificate.prototype.notAfter

Время окончания действия сертификата (в миллисекундах, прошедших с 1 января 1970 года GMT+0). При задании vcert.FIELD_NOTAFTER в поле fields.

– *Certificate.prototype.keyUsage*

Применение ключа (значение битов vcert.KEYUSAGE_XXX). При задании vcert.FIELD_KEYUSAGE в поле fields.

- Certificate.prototype.notBeforePrivate

  Время начала действия закрытого ключа (в миллисекундах, прошедших с 1 января 1970 года GMT+0). При задании vcert.FIELD_NOTBEFOREPRIVATE в поле fields.

- Certificate.prototype.notAfterPrivate

  Время окончания действия закрытого ключа (в миллисекундах, прошедших с 1 января 1970 года GMT+0). При задании vcert.FIELD_NOTAFTERPRIVATE в поле fields.

- Certificate.prototype.issuerAltName

  Альтернативное имя издателя (см. прототип vcert.AltName). При задании vcert.FIELD_ISSUERALTNAME в поле fields.

- Certificate.prototype.subjectAltName

  Альтернативное имя владельца (см. прототип vcert.AltName). При задании vcert.FIELD_SUBJECTALTNAME в поле fields.

- Certificate.prototype.keyId

  Строка идентификатор ключа, соответствующего сертификату. При задании vcert.FIELD_KEYID в поле fields.

- Certificate.prototype.policies

  Массив политик использования (см. прототип vcert.Policy). При задании vcert.FIELD_POLICY в поле fields.

- Certificate.prototype.extKeyUsage

  Массив расширенных использований ключа (см. прототип vcert.ExtKeyUsage). При задании vcert.FIELD_EXTKEYUSAGE в поле fields.

- Certificate.prototype.extensions

  Массив расширений (см. прототип vcert.Extension). При задании vcert.FIELD_EXTENSIONS в поле fields.

- Certificate.prototype.ertEncoded

  Сертификат в DER-кодировке в base64 строке. При задании vcert.FIELD_CERTENCODED в поле fields.

- Certificate.prototype.certHash

  Хэш сертификата в base64 кодировке. При задании vcert.FIELD_CERTHASH в поле fields.

1.4.4. Параметры выполнения функции

function SignParam(flag = 0)

Параметры вычисления ЭП.

Состав:

SignParam.prototype.flag

Флаги (могут использоваться vcert.FLAG_PKCS7, vcert.FLAG_DETACHED, vcert.FLAG_SIGN_XXX).

function VerifyParam(flag = 0, keyUsage = 0, policies = null, extKeyUsage = null, nSignToDelete = 0, minsigns = 0, info = 0)

Параметры проверки ЭП.

Состав:

- VerifyParam.prototype.flag

  Флаги (могут использоваться vcert.FLAG_PKCS7, vcert.FLAG_DETACHED, vcert.FLAG_VERIFY_XXX).
- VerifyParam.prototype.keyUsage

  Область применения ключа ЭП (используется только при задании vcert.FLAG_VERIFY_KEYUSAGE). Сертификат(ы), на котором(ых) выполнена(ны) ЭП, будет(ут) проверяться на наличие заданной области применения.
- VerifyParam.prototype.policies

  Массив политик использования (используется только при задании vcert.FLAG_VERIFY_POLICY). Сертификат(ы), на котором(ых) выполнена(ны) ЭП, будет(ут) проверяться на наличие заданных политик использования (см. прототип vcert.Policy).
- VerifyParam.prototype.extKeyUsage

  Массив расширенных использований ключа (используется только при задании vcert.FLAG_VERIFY_EXTKEYUSAGE). Сертификат(ы), на котором(ых) выполнена(ны) ЭП, будет(ут) проверяться на наличие заданных расширенных использований ключа (см. прототип vcert.ExtKeyUsage).
- VerifyParam.prototype.nSignToDelete

  Количество ЭП с конца, которые необходимо удалить (используется только при задании vcert.FLAG_VERIFY_DELSIGN). Для удаления всех ЭП следует установить равным vcert.DELETE_ALL_SIGNS.
- VerifyParam.prototype.minsigns

  Минимальное количество ЭП (используется только при задании vcert.FLAG_VERIFY_MINSIGNS).
- VerifyParam.prototype.info

Требуемая возвращаемая информация о сертификате(ах), на котором(ых) выполнена(ны) ЭП.

function DecryptParam(flag = 0, info = 0)

Параметры расшифрования.

Состав:

– DecryptParam.prototype.flag

Флаги (могут использоваться vcert.FLAG_PKCS7, vcert.FLAG_DECRYPT_XXX).

– DecryptParam.prototype.info

Требуемая возвращаемая информация о сертификате отправителя (только при использовании неанонимного шифрования).

function EncryptParam(recievers, flag = 0)

Параметры зашифрования.

Состав:

– EncryptParam.prototype.flag

Флаги (могут использоваться vcert.FLAG_PKCS7, vcert.FLAG_ENCRYPT_XXX).

– EncryptParam.prototype.receivers

Массив шаблонов сертификатов получателей (см. прототип vcert.Certificate). Для каждого шаблона данного массива в локальном справочнике сертификатов выполняется поиск (если указан флаг vcert.FLAG_ENCRYPT_REMOTE, то поиск производится также в сетевом справочнике), после которого все найденные сертификаты проверяются по срокам и возможности использования ключа для шифрования.

function FindParam(certTemplate, flag = 0, certInfo = 0)

Параметры поиска сертификата(ов) по заданному шаблону (см. прототип vcert.Certificate). Самый быстрый поиск возможен при задании уникального идентификатора сертификата – в этом случае поиск в справочнике выполняется по уникальному ключу. Поиск сертификата(ов) без задания уникального идентификатора может быть длительным и потребовать полного перебора объектов справочника.

Состав:

– FindParam.prototype.flag

Флаги (могут использоваться vcert.FLAG_FIND_MY для поиска своего рабочего сертификата, vcert.FLAG_FIND_REMOTE для поиска в сетевом справочнике, vcert.FLAG_FIND_XXX).

– FindParam.prototype.certTemplate

Шаблон сертификата для поиска (в зависимости от заданных полей шаблона производится поиск и сравнение найденных сертификатов).

– FindParam.prototype.info

Требуемая возвращаемая информация о найденном(ых) сертификате(ах).

function ImportParam(flag = 0)

Параметры импортирования объекта или объектов.

Состав:

– ImportParam.prototype.flag

Флаги (могут использоваться vcert.FLAG_IMPORT_XXX).

function ExportParam(flag = 0)

Параметры экспорта запросов (только при использовании локального СКЗИ).

Состав:

– ExportParam.prototype.flag

Флаги (могут использоваться vcert.FLAG_EXPORT_XXX).

function VerifyPolicyParam(flag = 0, check_time = 0, keyusage = 0, extKeyUsage = null, policies = null)

Параметры построения цепочки и проверки политики сертификата или СОС.

Состав:

– VerifyPolicyParam.prototype.size

Размер структуры в байтах.

– VerifyPolicyParam.prototype.flag

Флаги (могут использоваться FLAG_POLICY_XXX).

– VerifyPolicyParam.prototype.check_time

Момент времени, используемый для проверки сертификата (если не задан, то используется текущее время).

– VerifyPolicyParam.prototype.keyusage

Область применения ключа сертификата. Сертификат будет проверяться на наличие заданной области применения.

– VerifyPolicyParam.prototype.extkeyusages

Массив расширенных использований ключа. Сертификат будет проверяться на наличие заданных расширенных использований ключа (см. прототип vcert.ExtKeyUsage).

– VerifyPolicyParam.prototype.policies

Массив политик использования. Сертификат будет проверяться на наличие заданных политик использования (см. прототип vcert.Policy).

function TSPRequestParam(index,flag = 0)

Параметры создания запроса на получение штампа времени.

Состав:

– TSPRequestParam.prototype.flag

Флаги (могут использоваться vcert.FLAG_TSP_REQUEST_XXX).

– TSPRequestParam.prototype.index

Индекс (порядковый номер, начиная с «0») ЭП, для которой создается запрос.

function TSPResponseParam(flag= 0)

Параметры вычисления ЭП запроса на получение штампа времени.

Состав:

– TSPResponseParam.prototype.flag

Флаги (могут использоваться vcert.FLAG_TSP_RESPONSE_XXX).

function TSPVerifyParam(index, fields = 0,flag = 0)

Параметры проверки ЭП запроса на получение штампа времени.

Состав:

– TSPVerifyParam.prototype.flag

Флаги (могут использоваться vcert.FLAG_TSP_VERIFY_XXX).

– TSPVerifyParam.prototype.index

Индекс (порядковый номер, начиная с «0») ЭП, для которой создается запрос.

– TSPVerifyParam.prototype.info

Требуемая возвращаемая информация о сертификате, на котором выполнена ЭП.

function OCSPRequestParam(flag = 0)

Параметры создания запроса на получение статуса сертификата.

Состав:

– OCSPRequestParam.prototype.flag

Зарезервировано. Должно быть 0

function OCSPResponseParam(flag = 0)

Параметры вычисления ЭП запроса на получение статуса сертификата.

Состав:

– OCSPResponseParam.prototype.flag

Зарезервировано. Должно быть 0

function OCSPVerifyParam(flag = 0, info)

Параметры проверки ЭП запроса на получение статуса сертификата.

Состав:

– OCSPVerifyParam.prototype.info

Требуемая возвращаемая информация о сертификате, на котором выполнена ЭП.

– OCSPVerifyParam.prototype.flag

Зарезервировано. Должно быть 0

1.5 Описание функций

1.5.1 Общий механизм работы с библиотекой

Для использования ПК Web Plugin необходимо подключить файл библиотеки «library.js» до выполнения другого Javascript кода. После загрузки и установки соединения с плагином функционал библиотеки доступен через глобальную переменную vcert.

Все функции библиотеки используют асинхронное выполнение, если не указано обратное, с использованием стандартных объектов Promise. Для получения результата выполнения функции необходимо указать обработчики resolve и reject с помощью Promise.prototype.then().

Примечания

1 В дальнейшем под возвращаемым значением функции будет пониматься список полей объекта передаваемого в качестве аргумента обработчикам resolve и reject, если не указано обратное.

2 В силу особенностей работы Promise рекомендуется использовать оператор delete для ручного удаления результатов операций, чтобы избежать утечки памяти.

3 В ПК Web Plugin используется WebExtension Native Messaging API для организации доступа к функциям библиотеки. Сообщения между браузером и процессом приложения используют кодировку UTF-8, поэтому для передачи бинарных данных, их необходимо закодировать в base64.

4 Большинство функций для работы с данными принимаю дополнительные аргументы, контролирующие тип входных/выходных данных. Если необходимо обработать текстовые данные то необходимо установить соответствующие параметры в функциях.

1.5.2 Функции инициализации и деинициализации

vcert.**initialize** (profile = «MY» , flag = 0)

Функция инициализации контекста библиотеки

Возвращаемые значения:

В случае успеха:

- ***status***: «OK».

В случае ошибки:

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код, если есть.

Аргументы:

- ***profile*** Имя профиля, может быть равно «MY», что позволяет использовать настройки профилей Справочника сертификатов и диалоговое окно выбора профиля. При задании флага инициализации vcert.FLAG_INIT_REGISTRY позволяет указать требуемый профиль Справочника сертификатов по имени;
- ***flag*** флаги инициализации, могут быть vcert.FLAG_INIT_XXX.

  Примечания:

  1 На каждую страницу где была вызвана функция инициализации создается один контекст.

  2 Контекст деинициализируется при закрытии страницы пользователем или при вызове функции vcert.deinitialize().

  3 При доступе к локальному справочнику, расположенному в хранилище GDBM, следует иметь ввиду, что оно не обеспечивает синхронизации доступа, т.е. к данному конкретному локальному справочнику разрешено иметь доступ только посредством одного контекста библиотеки. При необходимости одновременного доступа к локальному справочнику из нескольких страниц следует пользоваться локальным справочником, расположенным в хранилище ODBC.

Пример 1 – Вызов функции инициализации

```
vcert.initialize()
    .then((val) => console.log(val.status+": Инициализация завершена"),
          (val) => console.log(val.text))
    .then(()    => /*какие-нибудь операции ...*/)
    .then(()    => vcert.deinitialize())
```

Vcert.**deinitialize** ()

Функция инициализации контекста библиотеки

Возвращаемые значения:

В случае успеха:

- ***status***: «OK»

В случае ошибки:

- ***status***: «error»

1.5.3 Функции работы с блоками памяти (блочные функции)

vcert.**hash** (data, dataf = 0)

Функция хэширования блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK».
- ***hash***: Строка с хэш - функцией блока данных в base64 кодировке.

В случае ошибки

- ***status***: «error»
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***data*** блок данных в виде строки или в base64 кодировке в зависимости от значения dataf;
- ***dataf*** флаг указывающий тип данных, если равен 0 то блок данных должен быть закодирован в base64, иначе блок данных представляет собой обычный текст.

vcert.**sign** (sp, data, sign, df = 0)

Функции вычисления ЭП блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»
- ***sign_out***: Строка с результатом вычисления ЭП блока данных в base64 кодировке.

В случае ошибки:

- ***status:*** «error»
- ***text:*** Текст ошибки;
- ***code:*** Не нулевой код ошибки, если есть.

Аргументы:

- ***sp*** параметры вычисления ЭП (см. прототип vcert.SignParam);
- ***data*** блок данных в виде строки или в base64 кодировке в зависимости от значения dataf;
- ***sign*** base64 строка содержащая блок данных с существующими ЭП (в случае добавления ЭП);
- ***dataf*** флаг указывающий тип данных параметра data, если равен 0 то блок данных должен быть закодирован в base64, иначе блок данных представляет собой обычный текст.

Примечание - Вычисление ЭП возможно, только если в рабочем сертификате установлен бит использования ключа vcert.KEYUSAGE_DIGITAL_SIGNATURE (то есть сертификат можно использовать для выполнения ЭП).

Данные всегда подаются в блоке данных data, в том числе и при вычислении первой ЭП в присоединенном формате. Если при вызове задан аргумент sign, то выполняется добавление ЭП к уже существующим.

Пример 2 – Вычисление присоединенной ЭП в PKCS#7 формате:

```
var signParam =  new vcert.SignParam();
signParam.flag = vcert.FLAG_PKCS7;
vcert.sign(signParam,"Строку, которую хотим подписать",null, 1)
        .then((val) => console.log(val.sign_out),
                (err) => console.log(err.text));
```

vcert.**verify** (vp, data, signin, dataf = 0, outf = 0)

Функции проверки ЭП блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***verify_result***: Объект содержащий информацию о результатах проверки подписи;
- ***sign_out***: Блок с оставшимися ЭП, в случае удаления ЭП. Если удаляются все подписи, то возможно вернуть данные ввиде текста используя параметр outf;
- ***out_type***: Тип блока данных sign_out, если равен 0 то блок данных закодирован в base64, иначе был передан простой текст.

В случае ошибки

- ***status*** : «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть;
- ***verify_result***: В случае ошибки проверки хотя бы одной ЭП, необходимо анализировать verify_result.

Аргументы:

- ***vp*** параметры проверки (см. прототип vcert.VerifyParam);
- ***data*** блок данных, в случае проверки отсоединенной ЭП, тип блока данных контролируется параметром dataf;
- ***sign_in*** блок с существующими ЭП для проверки;
- ***dataf*** флаг указывающий тип данных параметра data, если равен 0 то блок данных должен быть закодирован в base64, иначе блок данных представляет собой обычный текст;

– ***outf*** флаг указывающий тип возвращаемых данных, если равен 0 то блок данных будет закодирован в base64, иначе блок данных представляет собой обычный текст.

Примечания

1 Если результат удаления подписей не является UTF-8 строкой, данные возвращаемые функцией будут закодированы в base64 независимо от параметра outf.

2 При проверке ЭП по умолчанию не используется дата отзыва сертификата, на котором была выполнена ЭП, в СОС. Для проверки ЭП в доверенных архивах с использованием даты отзыва сертификата необходимо использовать флаг FLAG_VERIFY_USEREVTIME.

Пример 3 - Проверка ЭП в PKCS#7 формате из файла:

```
var fReader =  new FileReader();
//функция чтения файла, выбранного пользователем
function getBase64FromFile(){
      return  new Promise(function(resolve, reject){
         //<input type="file" id="fileinput">
         var file = document.getElementById("fileinput").files[0];
         fReader.onloadend = function (evt) {
            if(evt.target.readyState == FileReader.DONE){
               var header = ";base64,";
               var fileData = evt.target.result;
               var headpos = fileData.indexOf(header)+header.length;
               var base64Data = fileData.substr(headpos);
               resolve(base64Data);
               delete evt.target.result;
            }
         }
         fReader.readAsDataURL(file);
      });
}
var vp =  new vcert.VerifyParam();
vp.info = vcert.FIELD_ALL;
vp.flag = vcert.FLAG_PKCS7;
getBase64FromFile()
   .then((val) =>{
```

```
        return vcert.verify(vp,null,val);})
    .then((v)   => console.log(JSON.stringify(v.verify_result)),
        (err) => console.log(err.text));
```

vcert.**verifyAndSign** (vp,sp, data, signin, dataf = 0)

Функции проверки и добавления ЭП блока данных в случае успеха проверки

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***verify_result***: Объект содержащий информацию о результатах проверки подписи;
- ***sign_out***: блок с существующими и вновь вычисленной ЭП;

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть;
- ***verify_result***: В случае ошибки проверки хотя бы одной ЭП, необходимо анализировать verify_result.

Аргументы:

- ***vp*** параметры проверки ЭП (см. прототип vcert.VerifyParam);
- ***sp*** параметры вычисления ЭП (см. прототип vcert.SignParam);
- ***data*** блок данных, в случае проверки отсоединенной ЭП, тип блока данных контролируется параметром dataf;
- **sign_in** блок с существующими ЭП для проверки;
- ***dataf*** флаг указывающий тип данных параметра data, если равен 0 то блок данных должен быть закодирован в base64, иначе блок данных представляет собой обычный текст.

Пример 4 – Проверка существующих ЭП, их удаление и добавление собственной ЭП:

```
//функция сохранения файла используя диалог загрузки
function saveToFile(base64data){
     var dataURI = "data:application/octet-stream;base64,";
     dataURI = dataURI.concat(base64data);
   var link = document.createElement("a");
   var file = document.getElementById("file").files[0];
   link.download = file.name.concat(".p7s");
```

```
    link.href = dataURI;
    document.body.appendChild(link);
    link.click();
    document.body.removeChild(link);
    delete link.href;
}
var vp =  new vcert.VerifyParam();
vp.nSignToDelete = -1;
vp.flag = vcert.FLAG_VERIFY_DELSIGN;
vp.info = vcert.FIELD_ALL;
var sp = new vcert.SignParam();
getBase64FromFile().then((b64) => vcert.verifyAndSign(vp,sp,null,b64))
                .then((res) => saveToFile(res.sign_out),
                    (err) => console.log(err.text));
```

vcert.**signedInfo** (data)

Функция получения информации об ЭП блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***sign_info***: информация об ЭП блока данных.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***data*** блок с существующими ЭП.

vcert.**encrypt** (ep, data, dataf = 0)

Функция зашифрования блока данны

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***data_out***: блок с шифрованными данными.

В случае ошибки

- ***status***: «error»;

- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***ep*** параметры выполнения зашифрования (см. прототип vcert.EncryptParam);
- ***data*** блок с открытыми данными, тип блока данных контролируется параметром dataf;
- ***dataf*** флаг указывающий тип данных параметра data, если равен 0 то блок данных должен быть закодирован в base64, иначе блок данных представляет собой обычный текст.

Примечания

1 Выполнение зашифрования возможно, только если в сертификатах отправителя (рабочем сертификате) и получателей установлен хотя-бы один из битов использования ключа vcert.KEYUSAGE_DATA_ENCIPHERMENT, vcert.KEYUSAGE_KEY_ENCIPHERMENT или vcert.KEYUSAGE_KEY_AGREEMENT - то есть, если эти сертификаты разрешено использовать для выполнения шифрования.

2 Следует иметь ввиду, что данная функция не обеспечивает контроль целостности зашифрованных данных и аутентификацию источника данных. В случае необходимости контроля целостности данных и аутентификации их источника следует использовать функцию зашифрования блока данных после вычисления ЭП этого блока.

vcert.**decrypt** (dp, data, outf = 0)

Функция зашифрования блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***data_out***: блок с открытыми данными, тип блока данных контролируется параметром outf.
- ***out_type***: Тип блока данных data_out, если равен 0 то блок данных закодирован в base64, иначе был передан простой текст.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***dp*** параметры выполнения расшифрования (см. прототип vcert.DecryptParam);

- ***data*** блок с шифрованными данными;
- ***outf*** флаг указывающий тип данных параметра data, если равен 0 то блок данных должен быть закодирован в base64, иначе блок данных представляет собой обычный текст.

Примечание - Если результат расшифрования не является UTF-8 строкой, данные возвращаемые функцией будут закодированы в base64 независимо от параметра outf.

Пример 5 – Шифрование на все сертификаты с заданным в альтернативном имени адресом эл. почты с последующим расшифрованием:

```
var reciever = new vcert.Certificate(vcert.FIELD_SUBJECTALTNAME);
reciever.subjectAltName = new vcert.AltName();
reciever.subjectAltName.emailAddress = "ivanov@x509.ru";
var ep = new vcert.EncryptParam([reciever]);
ep.flag = vcert.FLAG_PKCS7;

var dp = new vcert.DecryptParam(vcert.FLAG_PKCS7);

vcert.encrypt(ep, "Сообщение", 1)
        .then((enc) => vcert.decrypt(dp, enc.data_out,1))
        .then((dec) => console.log(dec.data_out));
```

vcert.**encryptedInfo** (data)

Функция получения информации о зашифрованном блоке данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***encrypted_info***: информация о блоке с шифрованными данными.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***data*** блок с шифрованными данными;

vcert.**signAndEncrypt** (sp, ep, data, dataf = 0)

Функция вычисления ЭП и шифрования блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***data_out***: блок с шифрованными данными.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***sp*** параметры вычисления ЭП (см. прототип vcert.SignParam);
- ***ep*** параметры выполнения зашифрования (см. прототип vcert.EncryptParam);
- ***data*** блок с открытыми данными, тип блока данных контролируется параметром dataf;
- ***dataf*** флаг указывающий тип данных параметра data, если равен 0 то блок данных должен быть закодирован в base64, иначе блок данных представляет собой обычный текст.

vcert.**decryptAndVerify** (dp,vp, data, outf = 0)

Функция выполнения расшифрования и проверки ЭП блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***data_out***: блок с открытыми данными, тип блока данных контролируется параметром outf;
- ***out_type***: Тип блока данных data_out, если равен 0 то блок данных закодирован в base64, иначе был передан простой текст;
- ***decrypt_result***: результат расшифрования, содержащий сертификат отправителя зашифрованного сообщения (только для систем с неанонимным шифрованием);
- ***verify_result***: результат проверки ЭП;

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть;
- ***verify_result***: В случае ошибки проверки хотя бы одной ЭП, необходимо анализировать verify_result.

Аргументы:

- ***dp*** параметры выполнения расшифрования (см. прототип vcert.DecryptParam);
- ***vp*** параметры проверки ЭП (см. прототип vcert.VerifyParam);
- ***data*** блок с шифрованными данными;
- ***outf*** флаг указывающий тип возвращаемых данных, если равен 0 то блок данных будет закодирован в base64, иначе блок данных представляет собой обычный текст.

1.5.4 Функции работы с блоками памяти (потоковые функции)

vcert.**streamHashInit** (algorithm)

Функция инициализации потокового хэширования блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***id***: контекст выполнения потоковой операции.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть;

Аргументы:

- ***algorithm*** объектный идентификатор (OID) алгоритма хэширования (для ГОСТ Р 34.11-94 - 1.3.6.1.4.1.3670.1.1 (СКАД «Сигнатура») или 1.2.643.2.2.9 (CCERT PKI), для ГОСТ Р 34.11-2012 - 1.2.643.7.1.1.2.2).

vcert.**streamHashUpdate** (id, data)

Функция продолжения потокового хэширования блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK».

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть;

Аргументы:

- ***id*** контекст выполнения потоковой операции;
- ***data*** блок данных.

Примечание - В случае, если данная функция завершается с ошибкой, контекст выполнения потоковой операции освобождается и не может быть повторно использован.

vcert.**streamHashFinal** (id)

Функция финализации потокового хэширования блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***data***: хэш-функция данных.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть;

Аргументы:

- ***id*** контекст выполнения потоковой операции;
- ***data*** блок данных.

Примечание – В случае, если данная функция завершается с ошибкой, контекст выполнения потоковой операции освобождается и не может быть повторно использован.

vcert.**streamSignInit** (sp, sign_in = null)

Функция инициализации выполнения потоковой ЭП блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***id***: контекст выполнения потоковой операции.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть;

Аргументы:

- ***sp*** параметры вычисления ЭП (см. прототип vcert.SignParam);
- ***sign_in*** блок с существующими ЭП в base64 кодировке (в случае добавления ЭП).

Примечания

1 Вычисление ЭП возможно, только если в рабочем сертификате установлен бит использования ключа vcert.KEYUSAGE_DIGITAL_SIGNATURE (то есть сертификат можно использовать для выполнения ЭП).

2 Данная функция поддерживает вычисление отсоединенной(ых) ЭП только в формате PKCS#7, соответственно, при заполнении параметра pSignPara необходимо указывать флаги vcert.FLAG_DETACHED и vcert.FLAG_PKCS7.

Если при вызове указан блок ЭП sign_in, то выполняется добавление ЭП к уже существующим.

vcert.**streamSignUpdate** (id, sp, data)

Функция продолжения выполнения потоковой ЭП блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK».

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть;

Аргументы:

- ***id*** контекст выполнения потоковой операции.
- ***sp*** параметры вычисления ЭП (см. прототип vcert.SignParam);
- ***data*** блок данных в base64 кодировке.

Примечания

1 В качестве sp необходимо использовать те же параметры вычисления ЭП, которые использовались при создании контекста выполнения потоковой операции.

2 В случае, если данная функция завершается с ошибкой, контекст выполнения потоковой операции освобождается и не может быть повторно использован.

vcert.**streamSignFinal** (id, sp)

Функция финализации выполнения потоковой ЭП блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***sign_out***: результат вычисления ЭП в base64 кодировке.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть;

Аргументы:

– **id** контекст выполнения потоковой операции.

– ***sp*** параметры вычисления ЭП (см. прототип vcert.SignParam);

– ***data*** блок данных в base64 кодировке.

Примечания

1 В качестве sp необходимо использовать те же параметры вычисления ЭП, которые использовались при создании контекста выполнения потоковой операции.

2 По завершении данной функции контекст выполнения потоковой операции освобождается и не может быть повторно использован.

3 Результат выполнения данной функции по формату подписанных данных совместим с результатом выполнения функции vcert.sign(), при условии использования последней для вычисления отсоединенной(ых) ЭП в формате PKCS#7.

vcert.**streamVerifyInit** (vp, sign)

Функция инициализации проверки потоковой ЭП блока данных

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

– ***id***: контекст выполнения потоковой операции.

В случае ошибки

– ***status***: «error»;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть;

Аргументы:

– ***vp*** параметры проверки ЭП (см. прототип vcert.VerifyParam);

– ***sign*** блок с существующими ЭП для проверки.

Примечания

1 При проверке ЭП по умолчанию не используется дата отзыва сертификата, на котором была выполнена ЭП, в COC. Для проверки ЭП в доверенных архивах с использованием даты отзыва сертификата необходимо использовать флаг vcert.FLAG_VERIFY_USEREVTIME.

2 Данная функция поддерживает проверку отсоединенной(ых) ЭП только в формате PKCS#7, соответственно, при заполнении параметра vp необходимо указывать флаги vcert.FLAG_DETACHED и vcert.FLAG_PKCS7.

vcert.**streamVerifyUpdate** (id, vp, data)

Функция продолжения проверки потоковой ЭП блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK».

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть;

Аргументы:

- **id** контекст выполнения потоковой операции;
- ***vp*** параметры проверки ЭП (см. прототип vcert.VerifyParam);
- ***data*** блок данными в base64 кодировке.

Примечания

1 В качестве vp необходимо использовать те же параметры проверки ЭП, которые использовались при создании контекста выполнения потоковой операции.

2 В случае, если данная функция завершается с ошибкой, контекст выполнения потоковой операции освобождается и не может быть повторно использован.

vcert.**streamVerifyFinal** (id, vp)

Функция финализации проверки потоковой ЭП блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK».
- ***verify_result***: результат проверки ЭП;
- ***sign_out***: блок с оставшимися ЭП, в случае удаления ЭП в base64 кодировке.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть;
- ***verify_result***: в случае ошибки проверки хотя бы одной ЭП (для обработки результатов проверки необходимо анализировать verify_result.

Аргументы:

- ***id*** контекст выполнения потоковой операции;
- ***vp*** параметры проверки ЭП (см. прототип vcert.VerifyParam);

Примечания

1 В качестве vp необходимо использовать те же параметры проверки ЭП, которые использовались при создании контекста выполнения потоковой операции.

2 По завершении данной функции контекст выполнения потоковой операции освобождается и не может быть повторно использован.

vcert.**streamEncryptInit** (ep)

Функция инициализации анонимного потокового зашифрования блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK».
- ***id***: контекст выполнения потоковой операции.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть;

Аргументы:

- ***ep*** параметры выполнения зашифрования (см. прототип vcert.EncryptParam);

Примечания

1 Выполнение зашифрования возможно, только если в сертификатах отправителя (рабочем сертификате) и получателей установлен хотя-бы один из битов использования ключа vcert.KEYUSAGE_DATA_ENCIPHERMENT, vcert.KEYUSAGE_KEY_ENCIPHERMENT или vcert.KEYUSAGE_KEY_AGREEMENT - то есть, если эти сертификаты разрешено использовать для выполнения шифрования.

2 Следует иметь ввиду, что данная функция не обеспечивает контроль целостности зашифрованных данных и аутентификацию источника данных. В случае необходимости контроля целостности данных и аутентификации их источника следует использовать функцию зашифрования файла после вычисления ЭП этого файла.

3 Данная функция поддерживает зашифрование данных в формате PKCS#7, соответственно, при заполнении параметра ep необходимо указывать флаг vcert.FLAG_PKCS7.

vcert.**streamEncryptUpdate** (id, ep, data)

Функция продолжения анонимного потокового зашифрования блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***data_out***: блок с шифрованными данными.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***id*** контекст выполнения потоковой операции;
- ***ep*** параметры выполнения зашифрования (см. прототип vcetr.EncryptParam);
- ***data*** блок с открытыми данными в base64 кодировке.

Примечания

1 В качестве ep необходимо использовать те же параметры выполнения зашифрования, которые использовались при создании контекста выполнения потоковой операции.

2 В случае, если данная функция завершается с ошибкой, контекст выполнения потоковой операции освобождается и не может быть повторно использован.

vcert.**streamEncryptFinal** (id, ep)

Функция финализации анонимного потокового зашифрования блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***data_out***: блок с шифрованными данными в base64 кодировке.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***id*** контекст выполнения потоковой операции;
- ***ep*** параметры выполнения зашифрования (см. прототип vcert.EncryptParam);

Примечания

1 В качестве ep необходимо использовать те же параметры выполнения зашифрования, которые использовались при создании контекста выполнения потоковой операции.

2 По завершении данной функции контекст выполнения потоковой операции освобождается и не может быть повторно использован.

3 Результат выполнения данной функции по формату зашифрованных данных совместим с результатом выполнения функции vcert.encrypt(), при условии использования последней для зашифрования данных в формате PKCS#7.

vcert.**streamDecryptInit** (dp, data)

Функция инициализации анонимного потокового расшифрования блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***left***: длина необработанных данных. Поскольку обработка данных производится блоками, то последний блок, присутствующий только частично, обработан не будет;
- ***id***: контекст выполнения потоковой операции.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***dp*** параметры выполнения расшифрования (см. прототип vcert.DecryptParam);
- ***data*** блок данных в base64 кодировке.

Примечание - Данная функция поддерживает расшифрование данных в формате PKCS#7, соответственно, при заполнении параметра dp необходимо указывать флаг FLAG_PKCS7.

vcert.**streamDecryptUpdate** (id, dp, data)

Функция продолжения анонимного потокового расшифрования блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***left***: длина необработанных данных. Поскольку обработка данных производится блоками, то последний блок, присутствующий только частично, обработан не будет;
- ***data_out***: блок с открытыми данными в base64 кодировке.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- **id** контекст выполнения потоковой операции;
- ***dp*** параметры выполнения расшифрования (см. прототип vcert.DecryptParam);
- ***data*** блок данных в base64 кодировке.

Примечания

1 В качестве dp необходимо использовать те же параметры выполнения расшифрования, которые использовались при создании контекста выполнения потоковой операции.

2 В случае, если данная функция завершается с ошибкой, контекст выполнения потоковой операции освобождается и не может быть повторно использован.

vcert.**streamDecryptFinal** (id, dp)

Функция финализации анонимного потокового расшифрования блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***decrypt_result***: результат расшифрования, содержащий сертификат отправителя зашифрованного сообщения (только для систем с неанонимным шифрованием).

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***id*** контекст выполнения потоковой операции;
- ***dp*** параметры выполнения расшифрования (см. прототип vcert.DecryptParam);

Примечания

1 В качестве dp необходимо использовать те же параметры выполнения расшифрования, которые использовались при создании контекста выполнения потоковой операции.

2 По завершении данной функции контекст выполнения потоковой операции освобождается и не может быть повторно использован.

vcert.**streamEncryptInfo** (data)

Функция потокового получения информации о зашифрованном блоке данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***encrypted_info***: информация о блоке с шифрованными данными.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***data*** блок данных в base64 кодировке.

Примечание - При получении кода ошибки vcert.VCERT_E_DECRYPT_FORMAT следует увеличить размер подаваемого блока с шифрованными данными и повторно вызвать функцию vcert.streamEncryptedInfo().

1.5.5 Низкоуровневые функции

vcert.**signHash** (data)

Функция вычисления ЭП хэш-функции данных

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

– ***sign_out***: информация о блоке с шифрованными данными.

В случае ошибки

– ***status***: «error»;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***data*** блок данных в base64 кодировке.

vcert.**verifyHash** (sender, hash, sign)

Функция проверки ЭП хэш-функции данных

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

В случае ошибки

– ***status***: «error»;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***sender*** сертификат, на котором была выполнена ЭП, в DER-кодировке;

– ***hash*** блок с хэш-функцией данных;

– ***sign*** блок данных в base64 кодировке;

vcert.**verifyCert** (vp, data)

Функция построения цепочки и проверки действительности сертификата

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***cert***: раскодированный и разобранный сертификат.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***vp*** параметры проверки сертификата (см. прототип vcert.VerifyParam);
- ***data*** проверяемый сертификат в DER-кодировке;.

vcert.**verifyCertificatePolicy** (vp, data)

Функция построения цепочки и проверки политики сертификата

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***cert***: раскодированный и разобранный сертификат.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***vp*** параметры проверки сертификата (см. прототип vcert.VerifyPolicyParam);
- ***data*** проверяемый сертификат в DER-кодировке, в base64 строке.

vcert.**verifyCrlPolicy** (vp, data)

Функция построения цепочки и проверки политики СОС

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***vp*** параметры проверки СОС (необходимо установить флаг проверки политики СОС vcert.FLAG_POLICY_VERIFY_CRL);
- ***data*** проверяемый сертификат в DER-кодировке, в base64 строке.

1.5.6 Функции преобразования форматов

vcert.**detachSign** (data, outf = 0)

Функция отсоединения ЭП от подписанного блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***data_out***: блок данных, отделенный от ЭП;
- ***out_type***: Тип блока данных data_out, если равен 0 то блок данных закодирован в base64, иначе был передан простой текст.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***data*** блок данных, отделенный от ЭП в base64 кодировке;
- ***outf*** флаг указывающий тип возвращаемых данных, если равен 0 то блок данных будет закодирован в base64, иначе блок данных представляет собой обычный текст.

vcert.**atachSign** (data, sign, dataf =0)

Функция присоединения ЭП к блоку данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***sign_out***: блок данных с присоединенной(ыми) ЭП.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***data*** блок данных, отделенный от ЭП в base64 кодировке;

- ***sign*** ЭП в отсоединенном формате в base64 кодировке;
- ***dataf*** флаг указывающий тип данных параметра data, если равен 0 то блок данных должен быть закодирован в base64, иначе блок данных представляет собой обычный текст.

1.5.7 Функции поиска, импорта и экспорта

vcert.**findCert** (fp)

Функция поиска сертификата(ов) в справочнике(ах) по заданному шаблону

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***find_result***: результат поиска - массив найденных сертификатов.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***fp*** параметры поиска и шаблон сертификата (см. прототип vcert.FindParam).

Примечания

1 Если не указаны флаги vcert.FLAG_FIND_NOVERIFY, vcert.FLAG_FIND_NOTIMECHECK или vcert.FLAG_FIND_NOKEYTIMECHECK, то данная функция возвращает все найденные действительные (по построению цепочки, по сроку действия, по сроку действия закрытого ключа) сертификаты.

2 При указании флага vcert.FLAG_FIND_MY производится поиск рабочего сертификата (для которого есть соответствующий закрытый ключ).

3 Стандартная очередность поиска сертификатов по справочникам такова:

- *в начале, если не задан флаг vcert.FLAG_FIND_IGNORE_CACHED, поиск сертификата производится в кэше среди уже проверенных и кэшированных объектов;*
- *далее, если не задан флаг vcert.FLAG_FIND_IGNORE_LOCAL, поиск сертификата продолжается в локальном справочнике сертификатов;*
- *и в конце, если задан флаг vcert.FLAG_FIND_REMOTE и настроено подключение к сетевому справочнику сертификатов, поиск сертификата завершается в последнем. Необходимо отметить, что если поиск производится по уникальному идентификатору сертификата, то для поиска в сетевом справочнике установка флага*

vcert.FLAG_FIND_REMOTE не требуется, достаточно лишь наличие подключения к сетевому справочнику сертификатов;

- *если поиск выполняется по уникальному идентификатору сертификата, то, при нахождении первого сертификата, удовлетворяющего шаблону, процесс поиска обрывается;*
- *если поиск производится не по уникальному идентификатору сертификата, то процесс поиска не завершится до полного просмотра всех доступных справочников сертификатов и нахождения всех сертификатов, удовлетворяющих шаблону.*

Пример 6 – Поиск всех сертификатов по X.500 имени владельца и заданному в альтернативном имени адресом эл. почты с последующим диалогом выбора сертификата и с поиском в сетевом справочнике:

```
var fields = vcert.FIELD_SUBJECT | vcert.FIELD_SUBJECTALTNAME
var cert = new vcert.Certificate(fields);
cert.subject = "cn=ivanov,dc=x509,dc=ru";
cert.subjectAltName = new vcert.AltName();
cert.subjectAltName.emailAddress = "ivanov@x509.ru"
var fp = new vcert.FindParam(cert);
fp.flag = vcert.FLAG_FIND_REMOTE | vcert.FLAG_FIND_SELECTUI;
fp.info = vcert.FIELD_SUBJECT | vcert.FIELD_KEYID |
vcert.FIELD_CERTHASH;
vcert.findCert(fp).then((val) => console.log(val));
```

Пример 7 – Поиск всех рабочих сертификатов (для которых есть закрытый ключ):

```
 var cert = {fields : 0};
var fp = new vcert.FindParam(cert, vcert.FLAG_MY, null, vcert.FIELD_ALL);
vcert.findCert(fp).then((val) => console.log(val));
```

vcert.**import** (ip, data)

Функция добавления объекта/объектов из блока данных

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;

В случае ошибки

- ***status***: «error»;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***ip*** параметры выполнения импорта (см. прототип vcert.ImportParam);

– ***data*** импортируемый блок данных.

Примечание - Данная функция позволяет импортировать в локальный справочник сертификатов сертификаты и СОС в DER-кодировке, обновления, заверенные ЦС или ЦР, а также устанавливать новый рабочий сертификат (только для локального СКЗИ).

vcert.**export** (ep)

Функция создания и экспорта запросов в блок данных

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

– ***data_out***: блок данных с экспортированным запросом в base64 кодировке.

В случае ошибки

– ***status***: «error»;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***ep*** параметры выполнения экспорта (см. прототип vcert.ExportParam);

1.5.8 Вспомогательные функции

vcert.**base64ToUInt8Array** (base64data)

Функция конвертации base64 строки в Uint8Array

Возвращаемое значение (Функция не является асинхронной):

Типизированный массив Uint8Array.

Аргументы:

– ***base64data*** строка бинарных данных закодированных в base64.

vcert.**base64ToBlob** (base64data)

Функция конвертации base64 строки в Blob

Возвращаемое значение (Функция не является асинхронной):

Возвращает переменную типа Blob.

Аргументы:

– ***base64data*** строка бинарных данных закодированных в base64.

Пример 8 - Вызов диалога загрузки браузера для сохранения подписанных данных в файл

```
vcert.sign(signParam,base64Data)
    .then((val) =>{
      var base64res = val.sign_out;
      var blob = base64toBlob(base64res);
      var url = window.URL.createObjectURL(blob);
      var link = document.createElement("a");
      link.download = "filename.p7s";
      link.href = url;
      document.body.appendChild(link);
      link.click();
      setTimeout(function(){
          document.body.removeChild(link);
          delete link;
          link.href = "";
          window.URL.revokeObjectURL(url);
          delete val.sign_out;
      }, 100);
       });
```

vcert.**parseCert** (flag, data)

Функция разбора и получения информации о сертификате

Возвращаемые значения:

В случае успеха

– ***status*** : «OK»;

– ***cert*** : структура разобранного сертификата.

В случае ошибки

– ***status***: «error»;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***flag*** требуемая информация о сертификате (константы vcert.FIELD_XXX);

– ***data*** сертификат в DER-кодировке.

vcert.**parseCrl** (flag, data)

Функция разбора и получения информации о СОС

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

– ***crl***: структура разобранного СОС.

В случае ошибки

– ***status***: «error»;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***flag*** требуемая информация о СОС (константы vcert.FIELD_CRL_XXX);

– ***data*** СОС в DER-кодировке.

vcert.**parseAltnameEx** (data)

Функция разбора и получения массива других имен альтернативного имени

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

– ***altnames***: структура разобранного альтернативного имени.

В случае ошибки

– ***status***: «error»;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***data*** альтернативное имя в DER-кодировке.

Примечание - Для получения альтернативного имени в DER-кодировке следует найти в массиве расширений сертификата альтернативное имя издателя (OID 2.5.29.18) или владельца (OID 2.5.29.17).

vcert.**parseBasicConstraintsEx** (data)

Функция разбора и получения информации о базовых ограничениях сертификата

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

– ***constraints***: структура разобранных базовых ограничений.

В случае ошибки

– ***status***: «error»;

– ***text***: Текст ошибки;

– **code**: Не нулевой код ошибки, если есть.

Аргументы:

– ***data*** базовые ограничения в DER-кодировке;

Примечание - Для получения базовых ограничений в DER-кодировке следует найти в массиве расширений сертификата базовые ограничения (OID 2.5.29.19).

vcert.**getPKCS7Info** (data)

Функция получения информации о блоке данных в формате PKCS#7

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

– ***info***: информация о блоке данных в формате PKCS#7.

В случае ошибки

– ***status***: «error»;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***data*** блок данных в формате PKCS#7;

vcert.**CheckExpiringCertKey** (days)

Функция проверки истечения срока действия закрытого ключа в заданный период времени

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

– ***days***: период истечения срока действия ключа в днях;

– ***keyid***: идентификатор закрытого ключа сформированного запроса на получение рабочего сертификата, для которого рабочий сертификат еще не был получен.

В случае ошибки

– ***status***: «error»;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***days*** проверяемый период времени в днях.

vcert.**showCertificate** (data)

Функция отображения диалогового окна с информацией о сертификате

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

В случае ошибки

– ***status***: «error»;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***data*** сертификат в DER-кодировке.

vcert.**showCrl** (data)

Функция отображения диалогового окна с информацией о СОС

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

В случае ошибки

– ***status***: «error»;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***data*** СОС в DER-кодировке.

vcert.**updateCRLs** ()

Функция обновления найденных в локальном справочнике СОС из их точек распространения

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

В случае ошибки

– ***status***: «error»;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

1.5.9 Функции работы со справочниками и профилями

vcert.**createXmlRequest** (flag, cert)

Функция формирования закрытого ключа и XML запроса с парным открытым ключом

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***xml***: сформированный XML запрос, содержащий парный открытый ключ.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***flag*** флаги (могут использоваться vcert.FLAG_XML_REQUEST_XXX);
- ***cert*** аблон сертификата, на основании которого создается XML запрос (имеют значение следующие поля шаблона сертификата: X.500 имя владельца, альтернативное имя владельца, политики использования, расширенные использования ключа, расширения и применение ключа).

vcert.**issueCertificateStore** (data, localstore)

Функция формирования справочников сертификатов пользователя

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***psestore***: путь к создаваемому ПСП, в который будут перенесены все сертификаты корневых ЦС, входящие в ПСП контекста библиотеки;
- ***localstore***: путь к ЛСП, который был сформирован ЦР при выгрузке нового рабочего сертификата пользователя.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***data*** новый рабочий сертификат для выдаваемых справочников (данный сертификат будет использован для подписания ПСП, и он также будет добавлен в качестве рабочего сертификата в ЛСП);

- ***localstore*** путь к ЛСП, который был сформирован ЦР при выгрузке нового рабочего сертификата пользователя.

vcert.**createCertificateStore** (certs, crls)

Функция формирования справочников сертификатов пользователя из сертификатов и СОС

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***psestore***: путь к создаваемому ПСП, в который будут перенесены все сертификаты корневых ЦС;
- ***localstore***: путь к создаваемому ЛСП, в который будут перенесены все остальные объекты;

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***certs*** массив сертификатов (первый элемент данного массива должен быть рабочим сертификатом пользователя);
- ***crls*** массив СОС.

vcert.**createRegistryProfile** (profile, psestore, flag)

Функция создания нового профиля пользователя по заданному пути к ПСП и ЛСП

Возвращаемые значения:

В случае успеха

- ***status***: «OK».

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***profile*** имя создаваемого профиля, например, «По умолчанию»;
- ***psestore*** путь к ПСП и ЛСП. Для ПСП и ЛСП, находящихся в системном хранилище ОС Windows, следует использовать идентификатор ключа владельца рабочего сертификата

– ***flag*** флаги (могут использоваться vcert.FLAG_CREATE_REGISTRY_PROFILE_XXX).

vcert.**queryRegistryProfile** (index)

Функция создания нового профиля пользователя по заданному пути к ПСП и ЛСП

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

– ***profile***: имя профиля.

В случае ошибки

– ***status***: «error»;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***index*** индекс профиля.

1.5.10 Функции работы со штампом времени

vcert.**tspRequestFromPkcs7** (rp, data)

Функция создания запроса на получение штампа времени из ЭП блока данных формата PKCS#7

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

– ***request***: созданный запрос на получение штампа времени.

В случае ошибки

– ***status***: «error»;

– text: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***rp*** параметры создания запроса (см. прототип vcert.TSPRequestParam);

– ***data*** подписанный блок данных в формате PKCS#7;

vcert.**tspSignResponse** (rp, data)

Функция вычисления ЭП запроса на получение штампа времени

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

– ***response***: подписанный запрос на получение штампа времени.

В случае ошибки

– ***status***: error;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***rp*** параметры вычисления ЭП запроса (см. прототип vcert.TSPResponseParam);

– ***data*** запрос на получение штампа времени;

Примечание - Для вычисления ЭП запроса на получение штампа времени используемый для этого рабочий сертификат должен удовлетворять следующим условиям:

– *расширение сертификата **X509v3 Область Применения Ключа** должно быть помечено как **критичное** и должно содержать исключительно применения **Электронная Подпись** и **Неотрекаемый**;*

– *расширение сертификата **X509v3 Расширенная Область Применения Ключа** должно быть помечено как **критичное** и должно содержать исключительно расширенную область применения **Отметка времени** (1.3.6.1.5.5.7.3.8);*

– *расширение сертификата **X509v3 Основные Ограничения** должно быть помечено как **критичное** и должно содержать признак сертификата ЦС равный **CA:FALSE**.*

vcert.**TspVerifyResponse** (vp, data)

Функция проверки ЭП запроса на получение штампа времени

Возвращаемые значения:

В случае успеха

– ***status***: «OK»;

– ***verify_result***: подписанный запрос на получение штампа времени.

В случае ошибки

– ***status***: «error»;

– ***text***: Текст ошибки;

– ***code***: Не нулевой код ошибки, если есть.

Аргументы:

– ***vp*** параметры проверки ЭП запроса (см. прототип vcert.TSPVerifyParam);

– ***data*** подписанный запрос на получение штампа времени.

vcert.**TspUrlStampPkcs7** (rp, data, url)

Функция проверки ЭП запроса на получение штампа времени

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***sign_out***: подписанный блок данных в формате PKCS#7 со штампом времени.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***rp*** параметры создания запроса (см. прототип vcert.TSPRequestParam);
- ***data*** подписанный блок данных в формате PKCS#7;
- ***url*** адрес сервера штампов времени (например, http://pki.x509.ru/tsp/);

vcert.**TspVerifyPkcs7** (vp, data)

Функция проверки ЭП запроса на получение штампа времени из ЭП блока данных формата PKCS#7

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***verify_result***: результат проверки ЭП запроса.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***vp*** параметры проверки ЭП запроса (см. прототип vcert.TSPVerifyParam);
- ***data*** подписанный блок данных в формате PKCS#7 со штампом времени.

1.5.11 Функции определения статуса сертификатов

vcert.**ocspCreateRequest** (rp, data)

Функция создания запроса на получение статуса сертификата

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***request***: созданный запрос на получение статуса сертификата.

В случае ошибки

- ***status***: «error»;

- **text**: Текст ошибки;
- **code**: Не нулевой код ошибки, если есть.

Аргументы:

- ***rp*** параметры создания запроса (см. прототип vcert.OCSPRequestParam);
- ***data*** проверяемый сертификат в DER-кодировке.

vcert.**ocspSignResponse** (rp, data)

Функция вычисления ЭП запроса на получение статуса сертификата

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***response***: подписанный запрос на получение статуса сертификата.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***rp*** параметры вычисления ЭП запроса (см. прототип vcert.OCSPResponseParam);
- **data** запрос на получение статуса сертификата.

Примечание - Для вычисления ЭП запроса на получение статуса сертификата используемый для этого рабочий сертификат должен удовлетворять следующим условиям:

- *расширение сертификата **X509v3 Область Применения Ключа** должно быть помечено как **критичное** и должно содержать применения **Электронная Подпись** и **Неотрекаемый**;*
- *расширение сертификата **X509v3 Расширенная Область Применения Ключа** должно содержать расширенную область применения **Подпись OCSP** (1.3.6.1.5.5.7.3.9);*
- *сертификат должен содержать пустое расширение сертификата (расширение нулевой длины) **Без проверки OCSP** (1.3.6.1.5.5.7.3.9.5);*
- *расширение сертификата **X509v3 Основные Ограничения** должно быть помечено как **критичное** и должно содержать признак сертификата ЦС равный **CA:FALS**.*
- *сертификат и соответствующий ему СОС должны содержать хотя бы одну функционирующую точку распространения СОС для возможности автоматического обновления СОС.*

vcert.**OcspUrlObtainResponse** (rp, data, url)

Функция вычисления ЭП запроса на получение статуса сертификата на указанном сервере

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***response***: подписанный запрос на получение статуса сертификата.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***rp*** параметры создания запроса (см. прототип vcert.OCSPRequestParam);
- ***url*** адрес сервера получения статуса сертификатов (например, http://pki.x509.ru/ocsp/). Если значение равно null, то производится попытка извлечения адреса сервера из расширения Информация Доступа к Центру проверяемого сертификата;
- ***data*** проверяемый сертификат в DER-кодировке.

vcert.**OcspVerifyResponse** (vp, data)

Функция проверки ЭП запроса на получение статуса сертификата

Возвращаемые значения:

В случае успеха

- ***status***: «OK»;
- ***verify_result***: результат проверки ЭП запроса.

В случае ошибки

- ***status***: «error»;
- ***text***: Текст ошибки;
- ***code***: Не нулевой код ошибки, если есть.

Аргументы:

- ***rp*** параметры проверки ЭП запроса (см. прототип vcert.OCSPVerifyParam);
- **data** подписанный запрос на получение статуса сертификата.

2 ОПИСАНИЕ ОШИБОК

Ниже (Таблица 1) приведено описание ошибок, возникающих при использовании библиотеки прикладного программного интерфейса с сертификатами ключей для операционной системы Windows и библиотек прикладного программного интерфейса КС.

В левой колонке указано символьное имя ошибки и шестнадцатеричное значение соответствующего кода возврата.

В средней колонке указывается место возможного возникновения ошибки:

- **LOCAL** – функция(и) библиотеки (язык C\Java, платформа WIN32) прикладного программного интерфейса работы с сертификатами для платформы WIN32.
- **REMOTE** – функция(и) библиотеки интерфейса (язык C\платформа WIN32, язык Java) прикладного программного интерфейса КС. В дополнение к описанным ниже кодам возврата, функция библиотеки интерфейса могут возвращать код ошибок ОС Microsoft Windows, например, ERRO_INVALID_PARAMETER (0x00000057) или RPC_S_SERVER_UNAVAILABLE (0x000006BA). Детальное описание кодов ошибок ОС Microsoft Windows приведено в статье по ссылке http://http://msdn.microsoft.com/en-us/library/windows/desktop/ms681386\%28v=vs.85\%29.aspx;
- **CS** - исполняемый модуль КС, т.е. данная ошибка может возникнуть только в процессе выполнения КС, и она не является результатом вызова функции(й) библиотек интерфейса КС.

В правой колонке приведено детальное описание и причина возникновения ошибки.

Таблица 1 – Список ошибок

Код возврата	Место возникновения ошибки	Описание и причина возникновения ошибки
VCERT_OK (0x00000000)	LOCAL REMOTE CS	Успешное завершение функции
VCERT_E_GENERIC (0xE0700001)	LOCAL REMOTE CS	Данный код ошибки указывает на возможную внутреннюю ошибку, некорректные параметры настройки или на неправильные параметры, переданные в функции
VCERT_E_INVALID_PARAMETER (0xE0700002)	LOCAL REMOTE	В функцию был передан некорректный параметр. Возникает в случае использования неправильного контекста библиотеки или других параметров, неподдерживаемых флагов или на некорректную длину массива

Код возврата	Место возникновения ошибки	Описание и причина возникновения ошибки
VCERT_E_INVALID_CONTEXT (0xE0700003)	LOCAL REMOTE CS	Произошла попытка использования неправильного контекста библиотеки или контекста выполнения потоковой операции, нарушение параметров профилей в Реестре (в случае локальной библиотеки), ошибка в синтаксисе файла pki1.conf (в случае библиотеки КС) или ошибка при поиске объектов по запросу от АРМ УКС
VCERT_E_OPERATION_NOT_SUPPORTED (0xE0700004)	LOCAL REMOTE CS	Неправильные значения в структурах local_param_t или crsrv_param_t, вызов неподдерживаемой функции или нераспознанная команда с АРМ УКС
VCERT_E_INVALID_FLAG (0xE0700005)	LOCAL REMOTE	Некорректное использование флагов FLAG_DETACHED или FLAG_PKCS7
VCERT_E_NO_MEMORY (0xE0700006)	LOCAL REMOTE CS	Произошла ошибка при попытке выделения блока памяти необходимого размера
VCERT_E_HASH (0xE0700007)	LOCAL REMOTE	Произошла внутренняя ошибка СКЗИ при хэшировании (по ГОСТ Р 34.11-94 или ГОСТ Р 34.11-2012) данных из-за ошибки самотестирования криптографических процедур
VCERT_E_CERT_USAGE (0xE0700008)	LOCAL REMOTE CS	Отсутствует необходимое использование (шифрование или вычисление ЭП) или расширенное использование закрытого ключа сертификат
VCERT_E_CERT_FIND_PRIVATE_KEY (0xE0700009)	LOCAL REMOTE CS	Произошла ошибка при загрузке закрытого ключа во время инициализации контекста библиотеки работы с сертификатами или при запуске КС
VCERT_E_PKCS7_SET_TYPE (0xE070000A)	LOCAL REMOTE	Возникла нехватка ресурсов (оперативной памяти) при установке типа сообщения в PKCS#7 формате
VCERT_E_PKCS7_ADD_SIGNATURE (0xE070000C)	LOCAL REMOTE CS	Возникла нехватка ресурсов (оперативной памяти) при вычислении ЭП сообщения в PKCS#7 формате
VCERT_E_PKCS7_ADD_CERTIFICATE (0xE070000D)	LOCAL REMOTE	Возникла нехватка ресурсов (оперативной памяти) при добавлении сертификата в подписанное сообщение в PKCS#7 формате
VCERT_E_PKCS7_CONTENT_NEW (0xE070000E)	LOCAL REMOTE	Возникла нехватка ресурсов (оперативной памяти) при создании нового сообщения в PKCS#7 формате

Код возврата	Место возникновения ошибки	Описание и причина возникновения ошибки
VCERT_E_PKCS7_D2I (0xE070000F)	LOCAL REMOTE	Ошибка раскодирования сообщения из DER-кодировки в PKCS#7 формате из-за его повреждения или нехватки ресурсов
VCERT_E_PKCS7_I2D (0xE0700010)	LOCAL REMOTE	Возникла нехватка ресурсов (оперативной памяти) при закодировании в DER-кодировку сообщения в PKCS#7 формате
VCERT_E_PKEY_NOT_GOST (0xE0700011)	LOCAL REMOTE	Тип закрытого ключа СКЗИ не соответствует требуемому или у закрытого ключа неверный или нераспознанный идентификатор алгоритма
VCERT_E_SIGN (0xE0700012)	LOCAL REMOTE	Произошла внутренняя ошибка СКЗИ при вычислении ЭП из-за выгрузки закрытого ключа, ошибки самотестирования криптографических процедур или нехватки ресурсов
VCERT_E_VERIFY_POLICY (0xE0700013)	LOCAL REMOTE	Произошла ошибка при проверке наличия регламента в сертификате из-за его отсутствия в сертификате или нехватки ресурсов
VCERT_E_VERIFY_EXTKEYUSAGE (0xE0700014)	LOCAL REMOTE	Произошла ошибка при проверке наличия расширенного использования ключа в сертификате из-за его отсутствия в сертификате или нехватки ресурсов
VCERT_E_PKCS7_READ (0xE0700015)	LOCAL REMOTE CS	Произошла ошибка при попытке раскодирования из DER-кодировки подписанного сообщения в PKCS#7 формате из-за его искажения или нехватки ресурсов
VCERT_E_OVERFLOW (0xE0700016)	LOCAL REMOTE	Произошла ошибка при преобразовании данных из-за превышения допустимого размера
VCERT_E_REQ_DAMAGED (0xE0700017)	LOCAL REMOTE	Произошла ошибка при попытке раскодировать PKCS#10 запрос из DER-кодировки
VCERT_E_REVREQ_DAMAGED (0xE0700018)	LOCAL REMOTE	Произошла ошибка при попытке раскодировать запрос на отзыв сертификата из DER-кодировки
VCERT_E_VERIFY (0xE0700019)	LOCAL REMOTE	Произошла общая ошибка проверки ЭП документа
VCERT_E_VERIFY_FORMAT (0xE070001A)	LOCAL REMOTE CS	Выявлено нарушение формата подписанного сообщения или сообщение не является подписанным

Код возврата	Место возникновения ошибки	Описание и причина возникновения ошибки
VCERT_E_DELETE_SIGN (0xE070001B)	LOCAL REMOTE	Произошла ошибка удаления ЭП из подписанного сообщения, связанная с нехваткой ресурсов
VCERT_E_PKCS7_SET_CIPHER (0xE070001D)	LOCAL REMOTE	Возникла ошибка установки типа зашифрованного сообщения в PKCS#7 формате из-за нехватки ресурсов
VCERT_E_PKCS7_SET_CIPHER_INFO (0xE070001E)	LOCAL REMOTE	Возникла ошибка установки сертификата отправителя зашифрованного сообщения в PKCS#7 формате из-за нехватки ресурсов
VCERT_E_PKCS7_ADD_RECIPIENT (0xE070001F)	LOCAL REMOTE	Возникла ошибка добавления сертификата получателя зашифрованного сообщения в PKCS#7 формате из-за нехватки ресурсов
VCERT_E_PKCS7_ENCRYPT (0xE0700020)	LOCAL REMOTE	Возникла ошибка финализации формирования и закодирования зашифрованного сообщения в PKCS#7 формате
VCERT_E_ENCRYPT (0xE0700021)	LOCAL REMOTE	Возникла внутренняя ошибка СКЗИ при зашифровании сообщения из-за выгрузки закрытого ключа, ошибки самотестирования криптографических процедур или нехватки ресурсов
VCERT_E_PKCS7_WRONG_TYPE (0xE0700022)	LOCAL REMOTE CS	Попытка использования флага FLAG_DETACHED, не соответствующая подписанному сообщению, или обнаружение нераспознанного идентификатора, зашифрованного или подписанного сообщения в PKCS#7 формате
VCERT_E_EVP_GET_CIPHER (0xE0700023)	LOCAL REMOTE	Не был распознан идентификатор алгоритма шифрования при разборе зашифрованного сообщения в формате PKCS#7
VCERT_E_DECRYPT_NO_RECIPIENTS (0xE0700024)	LOCAL REMOTE	Не найден список идентификаторов сертификатов получателей зашифрованного сообщения в формате PKCS#7
VCERT_E_DECRYPT_NO_SENDER (0xE0700025)	LOCAL REMOTE	Не найден сертификат отправителя зашифрованного сообщения
VCERT_E_DECRYPT_NO_RECIPIENT (0xE0700026)	LOCAL REMOTE	Произведена попытка расшифровать сообщение в PKCS#7 формате на сертификате, идентификатор которого отсутствует в списке идентификаторов сертификатов получателей

Код возврата	Место возникновения ошибки	Описание и причина возникновения ошибки
VCERT_E_PKCS7_DECRYPT (0xE0700027)	LOCAL REMOTE	Произошла ошибка расшифрования сообщения в PKCS#7 формате из-за его искажения, нехватки ресурсов или внутренней ошибки СКЗИ
VCERT_E_DECRYPT (0xE0700028)	LOCAL REMOTE CS	Возникла внутренняя ошибка СКЗИ при расшифровании сообщения из-за выгрузки закрытого ключа, ошибки самотестирования криптографических процедур или нехватки ресурсов
VCERT_E_RANDOM (0xE0700029)	LOCAL REMOTE	Произошла внутренняя ошибка СКЗИ при формировании случайной последовательности из-за ошибки самотестирования криптографических процедур или нехватки ресурсов
VCERT_E_OPEN_CONFIG (0xE071002A)	LOCAL REMOTE	Произошла ошибка при открытии конфигурационного файла с параметрами профилей pki1.conf
VCERT_E_READ_CONFIG (0xE071002B)	LOCAL REMOTE	Произошла ошибка при чтении конфигурационного файла с параметрами профилей pki1.conf
VCERT_E_NO_DEFAULT_CONFIG (0xE071002C)	LOCAL REMOTE	При инициализации контекста библиотеки не было указано имя профиля и отсутствует профиль по умолчанию в конфигурационном файле с параметрами профилей pki1.conf
VCERT_E_OPEN_PSE (0xE070002D)	LOCAL CS	Возникла ошибка открытия, чтения или записи файла ПСП при инициализации контекста локальной библиотеки или загрузке КС, ошибка выполнения экспорта запроса на получение нового сертификата или запроса на отзыв рабочего сертификата в PKCS#7 формате, или ошибка импорта файла с обновлениями
VCERT_E_OPEN_LOCALSTORE (0xE070002E)	LOCAL CS	Возникла ошибка открытия, чтения или записи файла или БД ЛСП при инициализации контекста локальной библиотеки или загрузке КС
VCERT_E_VERIFY_STORE_USAGE (0xE070002F)	LOCAL CS	У файла ПСП или файла с обновлениями в формате PKCS#7 отсутствует или установлен некорректный идентификатор использования

Код возврата	Место возникновения ошибки	Описание и причина возникновения ошибки
VCERT_E_VERIFY_STORE (0xE0700030)	LOCAL CS	Нарушена целостность файла ПСП или файла с обновлениями в формате PKCS#7
VCERT_E_OPEN_LDAPSTORE (0xE0700031)	LOCAL CS	Произошла ошибка подключения к сетевому справочнику сертификатов
VCERT_E_NO_STREAM_DATA (0xE0700032)	LOCAL REMOTE	При попытке выполнения потокового расшифрования данных в формате PKCS#7 не была найдена ASN.1 последовательность шифрованного содержимого неопределенной длины
VCERT_E_BAD_STREAM_EOC (0xE0700033)	LOCAL REMOTE	Была произведена попытка расшифрования искаженных потоковых данных в формате PKCS#7 (не найдена метка окончания данных)
VCERT_E_VERIFY_CERT (0xE0700034)	LOCAL REMOTE CS	Произошла ошибка при построении или проверке цепочки сертификата
VCERT_E_CERT_MISSING (0xE0700035)	LOCAL REMOTE CS	Произошла ошибка при построении или проверке цепочки сертификата или СОС из-за отсутствия в справочниках необходимого сертификата издателя
VCERT_E_CERT_EXPIRED (0xE0700036)	LOCAL REMOTE CS	Произошла ошибка при построении или проверке цепочки сертификата или СОС из-за присутствия в цепочке сертификата с истекшим сроком действия
VCERT_E_CERT_DAMAGED (0xE0700037)	LOCAL REMOTE CS	Произошла ошибка при попытке раскодировать сертификат из DER-кодировки или проверить его ЭП
VCERT_E_CERT_BROKEN_CONSTRAINT (0xE0700038)	LOCAL REMOTE CS	Произошла ошибка при построении или проверке цепочки сертификата или СОС из-за нарушенных ограничений цепочки
VCERT_E_CERT_REVOKED (0xE0700039)	LOCAL REMOTE CS	Произошла ошибка при построении или проверке цепочки сертификата или СОС из-за наличия в цепочке отозванного сертификата
VCERT_E_CERT_UNTRUSTED (0xE070003A)	LOCAL REMOTE CS	Произошла ошибка при построении или проверке цепочки сертификата или СОС из-за невозможности построения полной цепочки, или произошла ошибка при импорте обновления из-за нераспознанного сертификата, подписавшего это обновление

Код возврата	Место возникновения ошибки	Описание и причина возникновения ошибки
VCERT_E_CRL_MISSING (0xE070003B)	LOCAL REMOTE CS	Произошла ошибка при построении или проверке цепочки сертификата или СОС из-за отсутствия в справочниках необходимого СОС издателя
VCERT_E_CRL_EXPIRED (0xE070003C)	LOCAL REMOTE CS	Произошла ошибка при построении или проверке цепочки сертификата или СОС из-за присутствия в цепочке СОС с истекшим сроком действия
VCERT_E_CRL_DAMAGED (0xE070003D)	LOCAL REMOTE CS	При попытке раскодировать СОС из DER-кодировки или проверить его ЭП произошла ошибка
VCERT_E_CERT_BROKEN_HIERARCHY (0xE070003E)	LOCAL REMOTE CS	Произошла ошибка при построении или проверке цепочки сертификата или СОС из-за нарушенных ограничений иерархии
VCERT_E_CHAIN_ERROR (0xE070003F)	LOCAL REMOTE CS	Произошла общая ошибка при построении или проверке цепочки сертификата или СОС из-за нехватки ресурсов
VCERT_E_TOO_MUCH_CERT (0xE0700040)	LOCAL REMOTE CS	При поиске сертификата по уникальному критерию (Идентификатору ключа или Издателю и Серийному номеру) было найдено более одного сертификата
VCERT_E_INVALID_USAGE (0xE0700041)	LOCAL REMOTE CS	При построении цепочки сертификата или СОС или при выполнении криптографической операции обнаружено, что сертификат не соответствует требуемому использованию (отсутствует необходимый регламент, использование или расширенное использование ключа)
VCERT_E_INVALID_SIGNATURE (0xE0700042)	LOCAL REMOTE CS	Произошла криптографическая ошибка СКЗИ при проверке ЭП из-за неверной ЭП
VCERT_E_OPENKEY_NOT_FOUND (0xE0700043)	LOCAL REMOTE CS	При построении цепочки сертификата или СОС произошла ошибка получения открытого ключа издателя для проверки ЭП
VCERT_E_SELFCHECK (0xE0700044)	LOCAL REMOTE CS	Произошла ошибка при выполнении самотестирования криптографических функций
VCERT_E_UPDATECRL (0xE0700045)	LOCAL REMOTE CS	Произошла ошибка при поиске СОС для обновления в ЛСП

Код возврата	Место возникновения ошибки	Описание и причина возникновения ошибки
VCERT_E_CERT_NOT_FOUND (0xE0700046)	LOCAL REMOTE CS	Необходимый сертификат для выполнения проверки ЭП или для выполнения расшифрования сообщения не был найден в доступных справочниках сертификатов
VCERT_E_CERT_NOT_YET_VALID (0xE0700047)	LOCAL REMOTE CS	Произошла ошибка при построении или проверке цепочки сертификата или СОС из-за присутствия в цепочке еще не действительного по времени сертификата
VCERT_E_REQ_NOT_FOUND (0xE0700048)	LOCAL REMOTE CS	Произошла ошибка при поиске запроса на получение нового рабочего сертификата в ЛСП из-за отсутствия ресурсов
VCERT_E_PKCS7_WRONG_ALGORITHM (0xE0700049)	LOCAL REMOTE	Была произведена попытка расшифрования данных в формате PKCS#7, зашифрованных блоком, с помощью потоковых функций расшифрования, или зашифрованное сообщение содержит нераспознанный идентификатор алгоритма
VCERT_E_KEY_EXPIRED (0xE070004C)	LOCAL REMOTE CS	Произошла ошибка из-за попытки использования закрытого ключа с истекшим сроком действия
VCERT_E_KEY_NOT_YET_VALID (0xE070004D)	LOCAL REMOTE CS	Произошла ошибка из-за попытки использования закрытого ключа с еще не наступившим сроком действия
VCERT_E_CRL_NOT_YET_VALID (0xE070004E)	LOCAL REMOTE CS	Произошла ошибка при построении или проверке цепочки сертификата или СОС из-за присутствия в цепочке еще не действительного по времени СОС
VCERT_E_INIT_CSP (0xE070004F)	LOCAL CS	При попытке инициализации контекста локальной библиотеки или запуске КС произошла внутренняя ошибка инициализации СКЗИ
VCERT_E_SIGN_INIT (0xE0700053)	LOCAL CS	При попытке инициализации контекста локальной библиотеки или запуске КС произошла ошибка инициализации модуля поддержки ЭП СКЗИ
VCERT_E_CRYPTO_INIT (0xE0700054)	LOCAL CS	При попытке инициализации контекста локальной библиотеки или запуске КС произошла ошибка инициализации модуля поддержки шифрования СКЗИ

Код возврата	Место возникновения ошибки	Описание и причина возникновения ошибки
VCERT_E_OPEN_PROFILES_KEY (0xE070005F)	LOCAL	При попытке инициализации контекста локальной библиотеки произошла ошибка чтения конфигурации профилей Справочника сертификатов из Реестра
VCERT_E_FIND_SESSION (0xE0700081)	REMOTE CS	Сессия КС, указанная в параметре функции, не найдена по ее идентификатору
VCERT_E_SIGNLEN (0xE0700082)	LOCAL REMOTE	При добавлении или проверке ЭП произошла ошибка контроля длины или искажение формата подписанного сообщения
VCERT_E_DECRYPT_FORMAT (0xE0700083)	LOCAL REMOTE	Ошибка при попытке декодирования зашифрованного сообщения в формате PKCS#7 из-за его искажения или его недостаточной длины
VCERT_E_ADD_OBJECT (0xE0700087)	LOCAL REMOTE CS	Произошла ошибка при добавлении объекта (сертификата, СОС или запроса на получение нового или на отзыв рабочего сертификата) в ПСП (для сертификатов корневого ЦС), ЛСП (для остальных объектов) или в файл в формате PKCS#7 для отправки в ЦР
VCERT_E_TOO_MANY_CERTS_FOUND (0xE0720089)	LOCAL REMOTE CS	При выполнении криптографической операции найдено более одного сертификата по уникальному критерию поиска
VCERT_E_USER_CANCEL (0xE072008A)	LOCAL	Операция прервана пользователем
VCERT_E_OPEN_INFILE (0xE070008B)	LOCAL REMOTE	Произошла ошибка открытия входного файла
VCERT_E_OPEN_OUTFILE (0xE070008C)	LOCAL REMOTE	Произошла ошибка открытия выходного файла
VCERT_E_READ_FILE (0xE070008D)	LOCAL REMOTE	Произошла ошибка чтения или отображения в память входного файла
VCERT_E_WRITE_FILE (0xE070008E	LOCAL REMOTE	Произошла ошибка записи или отображения в память выходного файла
VCERT_E_FILE_LENGTH (0xE070008F)	LOCAL REMOTE	Произошла ошибка из-за некорректной длины входного файла (нулевой или более 2 Гбайт)

Код возврата	Место возникновения ошибки	Описание и причина возникновения ошибки
VCERT_E_FILE_MAPPING (0xE0700090)	LOCAL REMOTE	Произошла ошибка отображения файла в оперативную память из-за нехватки ресурсов
VCERT_E_DELETE_OBJECT (0xE0700091)	CS	Произошла ошибка удаления объекта из ЛСП из-за нехватки ресурсов или искажения формата ЛСП
VCERT_E_INSUFFICIENT_SIGNS (0xE0700092)	LOCAL REMOTE	Количество обнаруженных ЭП подписанного сообщения меньше, чем требуемое
VCERT_E_VERIFY_CRL (0xE0700093)	LOCAL REMOTE CS	Произошла ошибка при построении или проверке цепочки СОС
VCERT_E_GET_PUBKEY (0xE0700094)	LOCAL REMOTE	Произошла ошибка при получении открытого ключа из сертификата подписавшего, возникшая при проверке ЭП хэш-функции данных
VCERT_E_REQ_NEW (0xE0700098)	LOCAL	Произошла ошибка при создании запроса на получение нового рабочего сертификата из-за ошибки формирования закрытого ключа или нехватки ресурсо
VCERT_E_REQ_SIGN (0xE070009A)	LOCAL	Произошла ошибка при вычислении ЭП запроса на получение нового рабочего сертификата из-за нехватки ресурсов
VCERT_E_MAKE_REVREQ (0xE070009B)	LOCAL	Произошла ошибка при создании запроса на отзыв рабочего сертификата из-за нехватки ресурсов
VCERT_E_SIGN_REVREQ (0xE070009C)	LOCAL	Произошла ошибка при вычислении ЭП запроса на отзыв рабочего сертификата из-за нехватки ресурсов
VCERT_E_LOAD_PRIVATE_KEY (0xE070009D)	LOCAL CS	Произошла ошибка при загрузке закрытого ключа с АРМ УКС или при установке нового рабочего сертификата
VCERT_E_ADD_SIGNER (0xE070009E)	LOCAL CS	Произошла ошибка при вычислении ЭП ПСП во время импорта обновления или при вычислении ЭП запроса
VCERT_E_PKCS7_DATA_INIT (0xE070009F)	LOCAL REMOTE	Ошибка при формировании зашифрованного сообщения в формате PKCS#7 из-за отсутствия ресурсов
VCERT_E_BIO_WRITE (0xE07000A0)	LOCAL REMOTE	Ошибка при формировании блока данных зашифрованного сообщения в формате PKCS#7 из-за отсутствия ресурсов

Код возврата	Место возникновения ошибки	Описание и причина возникновения ошибки
VCERT_E_OPEN_IDP (0xE07000A1)	LOCAL REMOTE CS	Произошла ошибка при попытке подключения к точке распространения СОС во время выполнения операции обновления
VCERT_E_READ_IDP (0xE07000A2)	LOCAL REMOTE CS	Произошла ошибка при чтении данных из точки распространения СОС во время выполнения операции обновления
VCERT_E_SESSIONCORRUPTED (0xE07000A3)	LOCAL	Произошла ошибка из-за неверной длины полученных данных контроля целостности во время проведения сеанса установления или при использовании шифрованного канала
VCERT_E_MAC (0xE07000A4)	LOCAL	Произошла ошибка из-за несоответствия вычисленной контрольной суммы, требуемой во время проведения сеанса установления, или при использовании шифрованного канала
VCERT_E_CRYPT (0xE07000A5)	LOCAL	Произошла ошибка при шифровании данных во время проведения сеанса установления или при использовании шифрованного канала из-за нехватки ресурсов или ошибки самотестирования криптографических процедур
VCERT_E_SPOOFING (0xE07000A6)	LOCAL	Обнаружено искажение типа передаваемого сообщения или данных сообщения во время проведения сеанса установления шифрованного канала
VCERT_E_CERTDONOTMATCH (0xE07000A7)	LOCAL	Обнаружено несоответствие сертификата ответной стороны заданному шаблону при обработке данных во время проведения сеанса установления шифрованного канала
VCERT_E_XMLINIT (0xE07000A8)	LOCAL REMOTE	Произошла ошибка при инициализации библиотеки разбора XML документов
VCERT_E_INVALID_CREDENTIALS (0xE02000A8)	REMOTE	Обнаружены аутентификационные данные с отрицательной или нулевой длиной, переданные в сессию КС
VCERT_E_ACCESS_DENIED (0xE02000A9)	REMOTE	Доступ к сессии КС запрещен из-за неверных аутентификационных данных
VCERT_E_SESSION_BLOCKED (0xA02000AA)	REMOTE	Сессия КС заблокирована или ожидает загрузки закрытого ключа

Код возврата	Место возникновения ошибки	Описание и причина возникновения ошибки
VCERT_E_CLIENT_INFO (0xE02000AB)	REMOTE	Произошла ошибка получения данных о клиенте уровня сетевого протокола DCE-RPC
VCERT_E_UNSECURE_CREDENTIALS (0xE02000AC)	REMOTE	Длина аутентификационных данных меньше минимальной требуемой длины в восемь (8) символов
VCERT_E_RESIGN_PROHIBIT (0xE02000AD)	REMOTE	Выполнение операции переподписания между указанными сессиями КС запрещено
VCERT_E_TSP_HASH_LENGTH (0xE0700100)	LOCAL	Неверная длина хэша при создании запроса на получение штампа времени
VCERT_E_TSP_HASH_ALGORITHM (0xE0700101)	LOCAL	Неизвестный или неверный алгоритм хэша при создании запроса на получение штампа времени
VCERT_E_TSP_CERT_PURPOSE (0xE0700102)	LOCAL	Сертификат не может быть использован для вычисления ЭП запросов на получение штампа времени
VCERT_E_TSP_SIGN_FAILED (0xE0700103)	LOCAL	Произошла ошибка при вычислении ЭП запроса на получение штампа времени
VCERT_E_TSP_NO_DIGEST (0xE0700104)	LOCAL	В списке атрибутов отсутствует хэш данных и/или ЭП с указанным индексом
VCERT_E_TSP_INVALID_SIGNER_NUM (0xE0700105)	LOCAL	Штамп времени содержит неверное (не равное 1) количество подписантов
VCERT_E_TSP_NO_TST_INFO (0xE0700106)	LOCAL	Произошла ошибка при получении информационного блока штампа времени
VCERT_E_TSP_RESP_D2I (0xE0700107)	LOCAL	Произошла ошибка при выполнении раскодирования подписанного штампа времени из DER-кодировки
VCERT_E_TSP_RESP_NOT_ISSUED (0xE0700108)	LOCAL	Штамп времени не был выдан авторитетным источником, то есть значение статуса результата отличается от 0 (granted) и 1 (grantedWithMods)
VCERT_E_TSP_DIGEST_MISMATCH (0xE0700109)	LOCAL	Полученный подписанный штамп времени содержит хэш данных, отличный от посланного в запросе хэша ЭП сообщения

Код возврата	Место возникновения ошибки	Описание и причина возникновения ошибки
VCERT_E_OCSP_CERT_PURPOSE (0xE0700140)	LOCAL	Сертификат не может быть использован для вычисления ЭП запросов на получение статуса сертификата
VCERT_E_OCSP_SIGN_FAILED (0xE0700141)	LOCAL	Произошла ошибка при вычислении ЭП запроса на получение статуса сертификата
VCERT_E_OCSP_RESP_D2I (0xE0700142)	LOCAL	Произошла ошибка при выполнении раскодирования подписанного запроса на получение статуса сертификата из DER-кодировки
VCERT_E_OCSP_RESP_NOT_ISSUED (0xE0700143)	LOCAL	Запрос на получение статуса сертификата не был подписан сетевым ответчиком, то есть значение статуса результата отличается от 0 (successful)
VCERT_E_OCSP_NOT_BASICRESP (0xE0700144)	LOCAL	Тип подписанного запроса на получение статуса сертификата не является Базовым
VCERT_E_OCSP_CERTID_MISMATCH (0xE0700145)	LOCAL	Идентификатор проверяемого сертификата из подписанного запроса на получение статуса сертификата отличается от посланного в запросе
VCERT_E_OCSP_ISSUER_MISMATCH (0xE0700146)	LOCAL	Имя издателя сертификата сетевого ответчика не соответствует имени издателя проверяемого сертификата

ПЕРЕЧЕНЬ ТАБЛИЦ

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (стр.) в документе	№ документа	Входящий № сопроводительного документа	Подпись	Дата
	измененных	замененных	новых	аннулированных					