Средство Криптографической Защиты Информации

OnixCrypt Iola COM

Версия 3.1

Руководство программиста

Содержание

1. Аннотация

OnixCrypt Iola COM версия 3.1 является дальнейшим развитием средства криптографической защиты информации (СКЗИ) OnixCrypt Iola COM, разработанного ТОО "Onix Group".

СКЗИ OnixCrypt Iola COM реализует казахстанские и зарубежные криптографические алгоритмы и разработано в виде ActiveX библиотеки, реализующей COM-объект (Component Object Model).

Данный документ описывает программный интерфейс СКЗИ OnixCrypt Iola COM, реализованный для следующих операционных систем:

- Windows 2000;
- Windows XP;
- Windows 2003.

2. Область применения

Реализация OnixCrypt Iola COM в виде COM-объекта позволяет легко встраивать поддержку казахстанских и зарубежных криптографических алгоритмов в различное прикладное программное обеспечение.

3. Основные функции

Основные функции, реализуемые OnixCrypt Iola CSP:

- генерация ключей ГОСТ 34.310-2004 для ЭЦП и обмена ключами;
- генерация ключей RSA для ЭЦП и шифрования;
- формирование закрытых ключей с возможностью защиты (шифрования ключевых контейнеров) на следующие типы носителей:
    - дискета 3,5";
    - файловая система;
    - реестр Windows;
    - сменный носитель с интерфейсом USB;
    - токены и смарт-карты Jemalto;
    - токены и смарт-карты Aladdin;
    - токены Pi-Card.
- возможность хранения сертификатов открытых ключей в ключевом контейнере;
- возможность генерации ключей с различными параметрами в соответствии с ГОСТ 34.310-2004;
- хеширование данных в соответствии с ГОСТ 34.311-1995;
- хеширование данных в соответствии с FIPS PUB 180-2;
- шифрование данных во всех режимах, определенных ГОСТ 28147-89;
- имитозащита данных в соответствии с ГОСТ 28147-89;
- формирование электронной цифровой подписи в соответствии с ГОСТ 34.310-2004 и RSA.
- шифрование данных во всех режимах, определенных RFC2268;
- шифрование данных во всех режимах, определенных IETF Draft "A Stream Cipher Encryption Algorithm "Arcfour"";
- шифрование данных во всех режимах, определенных FIPS PUB 46-3;
- опциональное использование пароля (пин-кода) для дополнительной защиты ключевой информации;
- реализация мер защиты от НСД ключевой информации пользователя.

4. Операционные системы

СКЗИ OnixCrypt Iola CSP версии 3.1 функционирует в следующих операционных системах (ОС):

- Windows 2000;
- Windows XP;
- Windows 2003.

5. Интерфейс криптопровайдера

5.1 Типы и идентификаторы

Данный раздел содержит определения идентификаторов и параметров, используемых в криптопровайдере "OnixCrypt Iola COM".

IOLACOM_STORETYPE. Перечислимый тип, определяющий способ хранения ключей/сертификатов (вид хранилища/носителя), используется для установки значений свойств KeyType/CertType.

IOLACOM_STORETYPE = TOleEnum;

Значения:

Значение IOLACOM_STORETYPE		Описание
ST_IOLA_FILE	1	Ключи/сертификаты хранится в файлах
ST_IOLA_STORE	2	Ключи хранятся в ключевых контейнерах в файле формата IolaStore, сертификаты хранятся в системном хранилище Windows
ST_IOLA_REGISTRY	3	Ключи хранятся в системном реестре, сертификаты хранятся в системном хранилище Windows
ST_IOLA_ETOKEN	4	Ключи/сертификаты хранятся на токенах/смарт-картах (Jemalto/Aladdin)
ST_IOLA_PCARD	5	Ключи/сертификаты хранятся на токенах Pi-Card
ST_IOLA_EXTERNAL	15	Ключи/сертификаты хранятся во внешнем хранилище, для взаимодействия используется внешняя библиотека

IOLACOM_DOCTYPE. Перечислимый тип, определяющий документы (данные) над которыми производятся операции, используется как параметр в функциях ExportDoc, InfoDoc.

IOLACOM_DOCTYPE = TOleEnum;

Значения:

Значение IOLACOM_DOCTYPE		Описание
DT_REQ	1	Запрос на сертификат в формате PKCS#10
DT_REQ_KEY	2	Запрос на сертификат в формате PKCS#10 и блоб закрытого ключа
DT_CERT	3	Сертификат в формате X.509
DT_CERT_KEY	4	Сертификат в формате X.509 и блоб закрытого ключа
DT_KEY	5	Блоб закрытого ключа
DT_PKCS7	6	Закрытый ключ/сертификат в формате PKCS#7

IOLACOM_KEYTYPE. Перечислимый тип, определяющий назначение (спецификацию) ключа при генерации ключей, запроса или самоподписанного сертификата, используется как параметр в функциях GenAndSaveKey, GenAndExportKey, GenSignedRequest, GenClearRequest, GenSelfSignedCert.

IOLACOM_KEYTYPE = TOleEnum;

Значения:

Значение IOLACOM_KEYTYPE		Описание
KT_KEYX	1	Ключ обмена
KT_SIGN	2	Ключ подписи
KT_CA	3	Ключ центра сертификации
KT_ROOT	4	Ключ корневого центра сертификации

IOLACOM_RNGTYPE. Перечислимый тип, определяющий способ генерации псевдослучайных чисел, используется для установки значений свойства RngType.

IOLACOM_RNGTYPE = TOleEnum;

Значения:

Значение IOLACOM_RNGTYPE		Описание
RT_IOLA_REGISTRY	1	Встроенный программный генератор
RT_IOLA_ETOKEN	2	Генератор на токенах/смарт-картах Jemalto/Aladdin
RT_IOLA_PCARD	3	Генератор на токенах Pi-Card
RT_IOLA_EXTERNAL	20	Генератор из внешней библиотеки

IOLACOM_LICTYPE. Перечислимый тип, определяющий источник данных, содержащий регистрационную лицензию, используется как параметр в функции ImportLic.

IOLACOM_LICTYPE = TOleEnum;

Значения:

Значение IOLACOM_LICTYPE		Описание
LT_DATA	1	Лицензия представлена в виде строки
LT_FILE	2	Лицензия содержится в файле

IOLACOM_CERTPROPS. Перечислимый тип, определяющий значение поля/расширения в запросе/сертификате, используется как параметр в функции PropDoc.

IOLACOM_CERTPROPS = TOleEnum;

Значения:

Значение IOLACOM_CERTPROPS		Описание
PROP_UNIQUE_NAME	0	Идентификатор открытого ключа
PROP_SUBJECT_STR	1	Общепринятое имя (CN) субъекта
PROP_ISSUER_STR	2	Общепринятое имя (CN) издателя
PROP_VALID_BEFORE	3	Срок (после) годности
PROP_VALID_AFTER	4	Срок (до) годности
PROP_PUBKEY_ALG	5	Алгоритм открытого ключа
PROP_KEY_USAGE	6	Использование ключа
PROP_SERIAL_NUMBER	7	Серийный номер
PROP_FILE_NAME	8	Идентификатор открытого ключа
PROP_SUBJECT_NAME	9	RDN субъекта
PROP_ISSUER_NAME	10	RDN издателя

IOLACOM_ERRORS. Перечислимый тип, определяющий подробные коды ошибок времени выполнения. Код ошибки может быть получен с помощью функции GetLastError.

IOLACOM_ERRORS = TOleEnum;

Значения:

Значение IOLACOM_ERRORS		Описание
IOLACOM_E_OK	0	
IOLACOM_E_DATA_EXISTS	1	
IOLACOM_E_FILE_READ	2	
IOLACOM_E_FILE_WRITE	3	
IOLACOM_E_DATA_PREPARE	4	
IOLACOM_E_PROV_ACQUIRE	5	
IOLACOM_E_HASH_CREATE	6	
IOLACOM_E_HASH_DATA	7	
IOLACOM_E_HASH_PARAM_GET	8	
IOLACOM_E_CERT_SEEK	9	
IOLACOM_E_SECKEY_IMPORT	10	
IOLACOM_E_SECKEY_EXPORT	11	
IOLACOM_E_SECKEY_SEEK	12	
IOLACOM_E_SECKEY_PREPARE	13	
IOLACOM_E_CERT_IMPORT	14	
IOLACOM_E_CERT_EXPORT	15	
IOLACOM_E_USER_CANCEL	16	
IOLACOM_E_REQ_SEEK	17	
IOLACOM_E_REQ_IMPORT	18	
IOLACOM_E_REQ_EXPORT	19	
IOLACOM_E_DATA_INVALID	20	
IOLACOM_E_CERT_REMOVE	21	
IOLACOM_E_REQ_REMOVE	22	
IOLACOM_E_SECKEY_REMOVE	23	
IOLACOM_E_SECKEY_GENERATE	24	
IOLACOM_E_PUBKEY_PREPARE	25	
IOLACOM_E_PUBKEY_IMPORT	26	
IOLACOM_E_PUBKEY_EXPORT	27	
IOLACOM_E_REQ_PREPARE	28	
IOLACOM_E_SIGN_SIGN	29	
IOLACOM_E_SIGN_VERIFY	30	
IOLACOM_E_LIC_INVALID	31	
IOLACOM_E_SIGN_EXISTS	32	
IOLACOM_E_SIGN_ATTRIBUTES	33	
IOLACOM_E_SIGN_CONTENTTYPE	34	
IOLACOM_E_SIGN_MESSAGEDIGEST	35	
IOLACOM_E_SIGN_NOT_EXISTS	36	
IOLACOM_E_SECKEY_SET_PARAM	37	
IOLACOM_E_SECKEY_GET_PARAM	38	
IOLACOM_E_RANDOM_GENERATE	39	
IOLACOM_E_ENCRYPT	40	
IOLACOM_E_DECRYPT	41	
IOLACOM_E_LIC_WRONG	42	
IOLACOM_E_CODE_WRONG	43	
IOLACOM_E_REQ_CONVERT	44	

IOLACOM_PRIVATEFLAGS. Перечислимый тип, определяющий формат экспорта закрытого ключа. Используется как параметр в функции ExportDoc().

IOLACOM_PRIVATEFLAGS = TOleEnum;

Значения:

Значение IOLACOM_PRIVATEFLAGS		Описание
PRF_ENCRYPT_KEY	65536	Зашифрованный ключевой блоб
PRF_BLOB_KEY	131072	Ключевой блоб
PRF_PVK_KEY	262144	Ключ в формате PVK
PRF_PKCS1_KEY	524288	Ключ в формате PKCS#1
PRF_PKCS8_KEY	1048576	Ключ в формате PKCS#8

IOLACOM_OPERFLAGS. Перечислимый тип, определяющий дополнительные условия выполнения операций, используется как параметр в функциях.

IOLACOM_OPERFLAGS = TOleEnum;

Значения:

Значение IOLACOM_OPERFLAGS		Описание
CF_WITH_FILES	1	Операция осуществляется над файлами. Значениями входных/выходных параметров являются имена файлов
CF_DRAFT_DATA	2	Входные данные и выходные данные являются raw-данными
CF_IN_BASE64	4	Входные данные в кодировке Base64
CF_IN2_BASE64	8	Дополнительные входные данные в кодировке Base64
CF_OUT_BASE64	16	Выходные данные в кодировке Base64
CF_MOVE_FILE	32	Удалять файлы после выполнения операции (в случае CF_WITH_FILES).
SF_DETACHED_DATA	64	Подпись храниться отдельно от данных
SF_WITH_TIME	128	Добавлять метку времени в подпись
SF_WITH_CERTS	256	Добавлять сертификат в подпись
SF_CHAIN_CERTS	512	Добавлять всю цепочку сертификатов в подпись
SF_WITH_CRLS	1024	Добавлять список отозванных сертификатов в подпись
VF_CERT_IMPORT	2048	Устанавливать сертификат (-ы) в хранилище в процессе выполнения операции
VF_CERT_SHOW	4096	Показывать сертификат (-ы) в процессе выполнения операции
CRF_PASS_STR	8192	Параметр Additional, содержит значение ключа для шифрования/расшифрования
CRF_PASS_FILE	16384	Параметр Additional, содержит имя файла содержащего ключ для шифрования/расшифрования
CRF_PASS_KEYX	32768	Параметр Additional, содержит идентификаторы ключей (серийные номера сертификатов) для получателей шифрования/расшифрования
IMF_CERT_AUTO	65536	Автоматически устанавливать сертификаты в хранилище
IMF_CERT_BROWSE	131072	Отображать диалог выбора сертификатов для установки
IMF_CHECK_REQ	262144	Проверять наличие запроса на сертификат в хранилище
IMF_WITH_KEYID	524288	Использовать идентификатор ключа в качестве имени контейнера

5.2 Свойства криптопровайдера

Свойство LangID. Определяет язык сообщений, генерируемых в процессе выполнения операций.

property LangID: Integer

Значения:

Значение LangID		Описание
LANG_ENGLISH	0	Английский
LANG_RUSSIAN	1	Русский

Свойство ParentWnd. Определяет дескриптор родительского окна, при выводе диалоговых окон.

property ParentWnd: Integer

Свойство ProvType. Определяет тип криптопровайдера.

property ProvType: Integer

Значения:

Значение ProvType	Описание
75	ГОСТ-криптопровайдер
1	RSA-криптопровайдер

Свойство ProvName. Определяет имя криптопровайдера.

property ProvName: WideString

Значения:

Значение ProvName	Описание
“IOLA GOST R 34.10-2001 L2 Cryptographic Service Provider”	ГОСТ-криптопровайдер
“IOLA RSA L2 Cryptographic Service Provider”	RSA-криптопровайдер

Свойство UserKeys. Определяет профайл хранения ключей.

property UserKeys: WordBool

Значения:

Значение UserKeys	Описание
True	Личные ключи (значение по умолчанию)
False	Машинные ключи

Свойство KeyType. Определяет способ хранения ключей (вид хранилища/носителя).

property KeyType: Integer

Значения: из набора IOLACOM_STORETYPE.

Свойство KeyPath. Определяет путь к ключевому хранилищу/носителю.

property KeyPath: WideString

Значения:

При KeyType	Описание
ST_IOLA_FILE	Путь к каталогу
ST_IOLA_STORE	Путь к каталогу
ST_IOLA_REGISTRY	Ключ реестра
ST_IOLA_ETOKEN	Имя считывателя
ST_IOLA_PCARD	Имя устройства

Свойство KeyDll. Определяет динамическую библиотеку, реализующую интерфейс доступа к ключевому хранилищу/носителю.

property KeyDll: WideString

Значения:

При KeyType	Описание
ST_IOLA_EXTERNAL	Имя файла

Свойство CertType. Определяет способ хранения сертификатов/запросов (вид хранилища/носителя).

property CertType: Integer

Значения: из набора IOLACOM_STORETYPE.

Свойство CertPath. Определяет путь к хранилищу/носителю сертификатов/запросов.

property CertPath: WideString

Значения:

При CertType	Описание
ST_IOLA_FILE	Путь к каталогу
ST_IOLA_STORE	Путь к каталогу
ST_IOLA_REGISTRY	
ST_IOLA_ETOKEN	Имя считывателя
ST_IOLA_PCARD	Имя устройства

Свойство CertDll. Определяет динамическую библиотеку, реализующую интерфейс доступа хранилищу/носителю сертификатов/запросов.

property CertDll: WideString

Значения:

При CertType	Описание

ST_IOLA_EXTERNAL	Имя файла

Свойство CertStore. Определяет раздел хранилища сертификатов/запросов.

property CertStore: WideString

Значения:

CertStore	Описание
“REQUEST”	Запросы на сертификат
“MY”	Личные сертификаты
“CA”	Промежуточные центры сертификации
“ROOT”	Доверенные корневые центры сертификации

Свойство RngType. Определяет тип датчика случайных чисел (тип генератора).

property RngType: Integer

Значения: из набора IOLACOM_RNGTYPE.

Свойство RngPath. Определяет путь к датчику.

property RngPath: WideString

Значения:

При RngType	Описание
RT_IOLA_REGISTRY	
RT_IOLA_ETOKEN	Имя считывателя
RT_IOLA_PCARD	Имя устройства

Свойство RngDll. Определяет динамическую библиотеку, реализующую интерфейс доступа к датчику случайных чисел.

property RngDll: WideString

Значения:

При RngType	Описание
RT_IOLA_EXTERNAL	Имя файла

Свойство OidHashSet. Определяет параметры алгоритма хеширования ГОСТ 34.311-1995.

property OidHashSet: WideString

Значения:

Значение OidHashSet	Описание
“1.2.643.2.2.30.1”	Параметры по умолчанию

Свойство OidSignSet. Определяет параметры алгоритма подписи ГОСТ 34.310-2004.

property OidSignSet: WideString

Значения:

Значение OidSignSet	Описание
“1.2.643.2.2.35.1”	Параметры подписи по умолчанию
“1.2.643.2.2.35.2”	Параметры Оскар 2.x
“1.2.643.2.2.35.3”	Параметры подписи 1

Свойство OidKeyXSet. Определяет параметры алгоритма обмена ключами ГОСТ 34.310-2004.

property OidKeyXSet: WideString

Значения:

Значение OidKeyXSet	Описание
“1.2.643.2.2.36.0”	Параметры обмена по умолчанию
“1.2.643.2.2.36.1”	Параметры обмена 1

Свойство OidCipherSet. Определяет параметры алгоритма шифрования ГОСТ 28147-89.

property OidCipherSet: WideString

Значения:

Значение OidCipherSet	Описание
“1.2.643.2.2.31.1”	Параметры по умолчанию
“1.2.643.2.2.31.2”	Параметры шифрования 1
“1.2.643.2.2.31.3”	Параметры шифрования 2
“1.2.643.2.2.31.4”	Параметры шифрования 3
“1.2.643.2.2.31.5”	Оскар 1.1
“1.2.643.2.2.31.6”	Оскар 1.0
“1.2.643.2.2.31.7”	РИК 1

5.3 Функции работы с ключами

Функция GenAndSaveKey(). Обеспечивает генерацию ключей и сохранение их в хранилище. Хранилище должно быть определено свойствами KeyType, KeyPath (KeyDll).
function GenAndSaveKey(AProvType, KeySpec, KeySize, AHashAlg: Integer; const AProvName: WideString): WideString; safecall;

Параметры:

AProvType [in] Тип криптопровайдера;

KeySpec [in] Спецификация/тип ключа;

KeySize [in] Длина ключа;

AHashAlg [in] Алгоритм хеширования;

AProvName [in] Имя криптопровайдера.

Результат:

При успешном завершении функция возвращает имя ключевого контейнера (уникальный идентификатор ключа), в противном случае возвращается пустая строка и генерируется соответствующее исключение. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция GenAndExportKey(). Обеспечивает генерацию ключей и возврат их значений.

function GenAndExportKey(AProvType, KeySpec, KeySize, AHashAlg: Integer; const AProvName: WideString; var PublicKeyData: WideString; var PrivateKeyData: WideString): WideString; safecall;

Параметры:

AProvType [in] Тип криптопровайдера;

KeySpec [in] Спецификация/тип ключа;

KeySize [in] Длина ключа;

AHashAlg [in] Алгоритм хеширования;

AProvName [in] Имя криптопровайдера;

PublicKeyData [out] Открытый ключ;

PrivateKeyData [out] Закрытый ключ.

Результат:

При успешном завершении функция возвращает имя ключевого контейнера (уникальный идентификатор ключа), параметр PublicKeyData содержит открытый ключ, параметр PrivateKeyData содержит закрытый ключ; в противном случае возвращается пустая строка и генерируется соответствующее исключение. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция SelectKey(). Обеспечивает вызов диалогового окна выбора ключа. Исходное хранилище определяется свойствами KeyType, KeyPath (KeyDll).

function SelectKey(const InfoCaption, InfoString1, InfoString2: WideString): WideString; safecall;

Параметры:

InfoCaption [in] Заголовок окна;

InfoString1 [in] Текст сообщения;

InfoString2 [in] Текст сообщения.

Результат:

При успешном завершении функция возвращает имя выбранного ключевого контейнера (уникальный идентификатор ключа) и устанавливает значения свойств KeyType, KeyPath (KeyDll) согласно диалогу, в противном случае возвращается пустая строка и

генерируется соответствующее исключение. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

5.4 Функции работы с запросами/сертификатами

Функция GenSignedRequest(). Обеспечивает генерацию и подпись запроса на сертификат PKCS#10 и сохранение его в хранилище. Ключевое хранилище должно быть определено свойствами KeyType, KeyPath (KeyDll), а хранилище запросов/сертификатов должно быть определено свойствами CertType, CertPath (CertDll).

function GenSignedRequest(const UserID: WideString; AProvType: Integer; KeySpec: Integer; AHashAlg: Integer; const AProvName: WideString; const CommonName, CountryName, StateOrProvinceName, LocalityName, OrganizationName, OrganizationalUnitName, EmailAddress: WideString; UseForEmail: WordBool): WideString; safecall;

Параметры:

UserID [in] Уникальный идентификатор ключа (имя ключевого контейнера).

AProvType [in] Тип криптопровайдера;

KeySpec [in] Спецификация/тип ключа;

KeySize [in] Длина ключа;

AHashAlg [in] Алгоритм хеширования;

AProvName [in] Имя криптопровайдера;

CommonName [in] RDN-имя субъекта;

CountryName [in] RDN-имя субъекта;

StateOrProvinceName [in] RDN-имя субъекта;

LocalityName [in] RDN-имя субъекта;

OrganizationName [in] RDN-имя субъекта;

OrganizationalUnitName [in] RDN-имя субъекта;

EmailAddress [in] RDN-имя субъекта;

UseForEmail [in] Добавляет расширение, обеспечивающее возможность использования сертификата при обмене электронными сообщениями.

Результат:

При успешном завершении функция возвращает значение запроса на сертификат PKCS#10 в Base64-кодировке, в противном случае возвращается пустая строка и генерируется соответствующее исключение. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция ConvertReq2Cert(). Обеспечивает преобразование запроса на сертификат PKCS#10 в неподписанный сертификат X.509 (выпуск неподписанного сертификата).

function ConvertReq2Cert(const CertRequest: WideString; var Certificate: WideString): WordBool; safecall;

Параметры:

CertRequest [in] Запрос на сертификат PKCS#10;

Certificate [out] Неподписанный сертификат X.509.

Результат:

При успешном завершении функция возвращает True, параметр Certificate содержит неподписанный сертификат; в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция SignDoc(). Обеспечивает подпись сертификата/запроса субъекта ключом издателя. Ключевое хранилище должно быть определено свойствами KeyType, KeyPath (KeyDll), а хранилище запросов/сертификатов должно быть определено свойствами CertType, CertPath (CertDll).

function SignDoc(const IssuerID: WideString; const PrivateData: WideString; var DocData: WideString): WordBool; safecall;

Параметры:

IssuerID [in] Уникальный идентификатор ключа издателя;

PrivateData [in] Закрытый ключ издателя, в случае если отсутствует в хранилище;

Certificate [in, out] Сертификат/запрос субъекта.

Результат:

При успешном завершении функция возвращает True, параметр Certificate содержит подписанный сертификат/запрос; в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция VerifyDoc(). Обеспечивает проверку подписи издателя под сертификатом/запросом. Ключевое хранилище должно быть определено свойствами KeyType, KeyPath (KeyDll), а хранилище запросов/сертификатов должно быть определено свойствами CertType, CertPath (CertDll).

function VerifyDoc(const UserID: WideString; const PublicData: WideString; const DocData: WideString): WordBool; safecall;

Параметры:

UserID [in] Уникальный идентификатор ключа издателя;

PublicData [in] Открытый ключ издателя, в случае если сертификат отсутствует в хранилище;

DocData [in] Сертификат/запрос субъекта.

Результат:

При успешном завершении функция возвращает True, в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция KeyPairDoc(). Обеспечивает проверку соответствия закрытого ключа субъекта сертификату/запросу.

function KeyPairDoc(const PrivateData: WideString; const DocData: WideString): WordBool; safecall;

Параметры:

PrivateData [in] Закрытый ключ;

DocData [in] Сертификат/запрос.

Результат:

При успешном завершении функция возвращает True, в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция GenSelfSignedCert (). Обеспечивает генерацию самоподписанного сертификата X.509 и сохранение его в хранилище. Ключевое хранилище должно быть определено свойствами KeyType, KeyPath (KeyDll), а хранилище запросов/сертификатов должно быть определено свойствами CertType, CertPath (CertDll).

function GenSelfSignedCert(const UserID: WideString; AProvType: Integer; KeySpec: Integer; AHashAlg: Integer; const AProvName: WideString; const CommonName, CountryName, StateOrProvinceName, LocalityName, OrganizationName, OrganizationalUnitName, EmailAddress: WideString; UseForEmail: WordBool): WordBool; safecall;

Параметры:

UserID [in] Уникальный идентификатор ключа (имя ключевого контейнера;

AProvType [in] Тип криптопровайдера;

KeySpec [in] Спецификация/тип ключа;

AHashAlg [in] Алгоритм хеширования;

AProvName [in] Имя криптопровайдера;

CommonName [in] RDN-имя субъекта;

CountryName [in] RDN-имя субъекта;

StateOrProvinceName [in] RDN-имя субъекта;

LocalityName [in] RDN-имя субъекта;

OrganizationName [in] RDN-имя субъекта;

OrganizationalUnitName [in] RDN-имя субъекта;

EmailAddress [in] RDN-имя субъекта;

UseForEmail [in] Добавляет расширение, обеспечивающее возможность использования сертификата при обмене электронными сообщениями.

Результат:

При успешном завершении функция возвращает True, в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция PropDoc(). Обеспечивает получение значений полей/расширений из сертификата/запроса, определенных в наборе IOLACOM_CERTPROPS. Хранилище запросов/сертификатов должно быть определено свойствами CertType, CertPath (CertDll). См. также PropDoc3.

function PropDoc(const UserID: WideString; PropID: Integer): WideString; safecall;

Параметры:

UserID [in] Уникальный идентификатор ключа;

PropID [in] см. **IOLACOM_CERTPROPS**.

Результат:

При успешном завершении функция возвращает значение поля/расширения, в противном случае возвращается пустая строка и генерируется соответствующее исключение. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция ProvDoc(). Обеспечивает автоматическую установку свойств ProvType и ProvName по указанному сертификату/запросу. Хранилище запросов/сертификатов должно быть определено свойствами CertType, CertPath (CertDll).

function ProvDoc(const UserID: WideString): WordBool; safecall;

Параметры:

UserID [in] Уникальный идентификатор ключа.

Результат:

При успешном завершении функция возвращает True, в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция ViewDoc(). Обеспечивает просмотр сертификата, с помощью системного диалогового окна. Хранилище сертификатов должно быть определено свойствами CertType, CertPath (CertDll).

procedure ViewDoc(const UserID: WideString); safecall;

Параметры:

UserID [in] Уникальный идентификатор ключа.

Результат:

При успешном выполнении будет отображено окно просмотра сертификата, в противном случае генерируется соответствующее исключение. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

5.5 Функции работы с ключами/запросами/сертификатами

Функция RemoveDoc(). Обеспечивает удаление ключа/запроса/сертификата из хранилища. Ключевое хранилище должно быть определено свойствами KeyType, KeyPath (KeyDll), а хранилище запросов/сертификатов должно быть определено свойствами CertType, CertPath (CertDll).

function RemoveDoc(const UserID: WideString; RemoveType: Integer): WordBool; safecall;

Параметры:

UserID [in] Уникальный идентификатор ключа;

RemoveType [in] см. **IOLACOM_DOCTYPE**.

Результат:

При успешном завершении функция возвращает True, в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция CheckDoc(). Проверяет наличие ключа/запроса/сертификата в хранилище. Ключевое хранилище должно быть определено свойствами KeyType, KeyPath (KeyDll), а хранилище запросов/сертификатов должно быть определено свойствами CertType, CertPath (CertDll).

function CheckDoc(const UserID: WideString): Integer; safecall;

Параметры:

UserID [in] Уникальный идентификатор ключа;

Результат:

При успешном завершении функция возвращает значение, определяющее доступные элементы, см. IOLACOM_DOCTYPE, в противном случае генерируется соответствующее исключение. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

.

Функция InfoDoc(). Обеспечивает получение информации о размещении и размере ключа/запроса/сертификата. Ключевое хранилище должно быть определено свойствами KeyType, KeyPath (KeyDll), а хранилище запросов/сертификатов должно быть определено свойствами CertType, CertPath (CertDll).

function InfoDoc(const UserID: WideString; InfoType: Integer; var DataInfo: WideString; var SizeInfo: Integer): WordBool; safecall;

Параметры:

UserID [in] Уникальный идентификатор ключа;

InfoType [in] см. **IOLACOM_DOCTYPE**;

DataInfo [out] Информация о размещении ключа/запроса/сертификата;

SizeInfo [out] Информация размере ключа/запроса/сертификата.

Результат:

При успешном завершении функция возвращает True, параметр DataInfo содержит информацию о размещении элемента, параметр SizeInfo содержит информацию о размере элемента; в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

5.6 Функции импорта/экспорта

Функция ImportDoc(). Обеспечивает установку ключа/запроса/сертификата в хранилище. Ключевое хранилище должно быть определено свойствами KeyType, KeyPath (KeyDll), а хранилище запросов/сертификатов должно быть определено свойствами CertType, CertPath (CertDll).

function ImportDoc(const DocData, PrivateData: WideString; OperFlags: Integer; var ImportInfo: WideString): WordBool; safecall;

Параметры:

DocData [in] Сертификат или запрос на сертификат, если присутствует;

PrivateData [in] Закрытый ключ, если присутствует;

OperFlags [in] см. **IOLACOM_OPERFLAGS;**

ImportInfo [out] Информация о выполненных действиях.

Результат:

При успешном завершении функция возвращает True и параметр ImportInfo содержит информацию о выполненных действиях, в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция ImportWin(). Обеспечивает установку ключа в криптопровайдер (установленный в системе), а сертификата в системное хранилище. Ключевое хранилище должно быть определено свойствами KeyType, KeyPath (KeyDll), а хранилище запросов/сертификатов должно быть определено свойствами CertType, CertPath (CertDll).

function ImportWin(const UserID, WinStores, WinProvName: WideString; WinProvType: Integer; AssignKey: WordBool): WordBool; safecall;

Параметры:

UserID [in] Уникальный идентификатор ключа;

WinStores [in] Системное хранилище: “MY”, “CA”, “ROOT”;

WinProvName [in] Имя криптопровайдера;

WinProvType [in] Тип криптопровайдера;

AssignKey [in] Определяет, необходимо ли связывать закрытый ключ с сертификатом.

Результат:

При успешном завершении функция возвращает True, в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция ExportDoc(). Обеспечивает получение значений ключа/запроса/сертификата из хранилища. Ключевое хранилище должно быть определено свойствами KeyType, KeyPath (KeyDll), а хранилище запросов/сертификатов должно быть определено свойствами CertType, CertPath (CertDll).

function ExportDoc(const UserID: WideString; ExportType: Integer; OperFlags: Integer; var DocData: WideString; PrivateFalgs: Integer; var PrivateData: WideString): WordBool; safecall;

Параметры:

UserID [in] Уникальный идентификатор ключа;

ExportType [in] см. **IOLACOM_DOCTYPE**;

OperFlags [in] см. **IOLACOM_OPERFLAGS**;

DocData [out] Запрос/сертификат;

PrivateFalgs [in] см. **IOLACOM_PRIVATEFLAGS;**

PrivateData [out] Закрытый ключ.

Результат:

При успешном завершении функция возвращает True, в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

5.7 Функции хеширования

Функция HashData(). Обеспечивает вычисление значения хеш-функции над данными.

function HashData(OperFlags, HashAlg: Integer; const InData: WideString; var OutData: WideString): WordBool; safecall;

Параметры:

OperFlags [in] см. **IOLACOM_OPERFLAGS;**

HashAlg [in] Алгоритм хеширования;

InData [in] Исходные данные;

OutData [out] Значение хеш-функции.

Результат:

При успешном завершении функция возвращает True, параметр OutData содержит значение хеш-функции; в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

5.8 Функции установки/проверки подписи

Функция SignData(). Обеспечивает вычисление значения подписи над данными.

function SignData(const SignerID: WideString; OperFlags: Integer; const InData, InSign: WideString; var OutData: WideString): WordBool; safecall;

Параметры:

SignerID [in] Уникальный идентификатор ключа подписывающего;

OperFlags [in] см **IOLACOM_OPERFLAGS**;

InData [in] Исходные данные;

InSign [in] Существующие подписи, если присутствуют;

OutData [out] Значение подписи (-ей).

Результат:

При успешном завершении функция возвращает True, параметр OutData содержит значение подписи; в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция VerifyData(). Обеспечивает проверку значения подписи над данными.

function VerifyData(const SignerID: WideString; OperFlags: Integer; const InData, InSign: WideString; var OutData, OutSign, ImportInfo, VerifyInfo: WideString): WordBool; safecall;

Параметры:

SignerID [in] Уникальный идентификатор ключа проверяемого. Если отсутствует, то проверяются все подписи.

OperFlags [in] см **IOLACOM_OPERFLAGS;**

InData [in] Исходные данные (в случае **SF_DETACHED_DATA**) или PKCS#7;

InSign [in] Значения подписи (-ей) (в случае **SF_DETACHED_DATA**);

OutData [out] Подписанные данные, отдельно от подписи (-ей);

OutSign [out] Значения подписи (-ей), отдельно от данных;

VerifyInfo [out] Информация о результате проверки подписи (-ей).

Результат:

При успешном завершении функция возвращает True, параметр OutData содержит подписанные данные, параметр OutSign содержит значение подписи (-ей), параметр VerifyInfo информацию о результате проверки подписи (-ей); в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

5.9 Функции шифрования/расшифрования данных

Функция EncryptData(). Обеспечивает шифрование данных.

function EncryptData(const UserID, Additional: WideString; OperFlags, Algorithm, Size, Mode: Integer; const InData: WideString; var EncryptInfo, OutData: WideString): WordBool; safecall;

Параметры:

UserID [in] Уникальный идентификатор ключа отправителя;

OperFlags [in] см **IOLACOM_OPERFLAGS**;

Algorithm [in] Алгоритм шифрования;

Size [in] Размер блока шифрования;

Mode [in] Режим шифрования;

InData [in] Исходные данные;

EncryptionInfo [out] Информация о результате шифрования;

OutData [out] Зашифрованные данные.

Результат:

При успешном завершении функция возвращает True, параметр OutData содержит зашифрованные данные, параметр EncryptionInfo содержит информацию о результате шифрования; в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция DecryptData(). Обеспечивает расшифрование данных.

function DecryptData(const UserID, Additional: WideString; OperFlags, Algorithm, Size, Mode: Integer; const InData: WideString; var OutData: WideString): WordBool; safecall;

Параметры:

UserID [in] Уникальный идентификатор ключа получателя.

OperFlags [in] см **IOLACOM_OPERFLAGS;**

Algorithm [in] Алгоритм расшифрования;

Size [in] Размер блока расшифрования;

Mode [in] Режим расшифрования;

InData [in] Зашифрованные данные;

OutData [out] Расшифрованные данные.

Результат:

При успешном завершении функция возвращает True, параметр OutData содержит расшифрованные данные; в противном случае False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

5.10 Вспомогательные функции

Функция ImportLic(). Обеспечивает ввод лицензии (данных о регистрации) на использование IolaCOM.

function ImportLic(LicType: Integer; const LicData: WideString): WordBool; safecall;

Параметры:

LicType [in] см. **IOLACOM_LICTYPE**;

LicData [in] Данные о лицензии.

Результат:

При успешном завершении функция возвращает True, в противном случае возвращает False. Подробный код ошибки (см. IOLACOM_ERRORS) может быть получен с помощью функции GetLastError().

Функция GetLastError(). Обеспечивает получение подробного кода ошибки, возникшей в процессе выполнения операций, см. IOLACOM_ERRORS.

function GetLastError: Integer; safecall;