**СКЗИ «KALKAN-CRYPT (PHP)» v1.0.0.2**

**Общее описание СКЗИ и архитектуры построения**

**Константы криптографической библиотеки KalkanCrypt**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Константы, определяющие способ хранения ключей/сертификатов (вид хранилища/носителя).**  **Значения:**   |  |  |  | | --- | --- | --- | | **Значение** | | **Описание** | | **KCST\_PKCS12** | 0x00000001 | Файловая система (небезопасный способ хранения ключей) | | **KCST\_KZIDCARD** | 0x00000002 | Удостоверение личности гражданина РК | | **KCST\_KAZTOKEN** | 0x00000004 | Казтокен | | **KCST\_ETOKEN72K** | 0x00000008 | eToken 72k | | **KCST\_JACARTA** | 0x00000010 | JaCarta | | **KCST\_X509CERT** | 0x00000020 | Сертификат X509 | | **KCST\_AKEY** | 0x00000040 | aKey | |
| **Константы, определяющие принадлежность сертификата.**  **Значения:**   |  |  |  | | --- | --- | --- | | **Значения** | | **Описание** | | **KC\_CERT\_CA** | 0x00000201 | Корневой сертификат УЦ | | **KC\_CERT\_INTERMEDIATE** | 0x00000202 | Сертификат промежуточного УЦ | | **KC\_CERT\_USER** | 0x00000204 | Сертификат пользователя | |
| **Константы, определяющие тип кодировки.**  **Значения:**   |  |  |  | | --- | --- | --- | | **Значения** | | **Описание** | | **KC\_CERT\_DER** | 0x00000101 | Кодировка DER | | **KC\_CERT\_PEM** | 0x00000102 | Кодировка PEM | | **KC\_CERT\_B64** | 0x00000104 | Кодировка Base64 | |
| **Константы, определяющие тип валидации.**  **Значения:**   |  |  |  | | --- | --- | --- | | **Значения** | | **Описание** | | **KC\_USE\_NOTHING** | 0x00000401 | Не делать проверок | | **KC\_USE\_CRL** | 0x00000402 | Проверка сертификата по списку отозванных сертификатов | | **KC\_USE\_OCSP** | 0x00000404 | Проверка сертификата посредством сервиса OCSP | |
| **Константы, определяющие значение поля/расширения в запросе/сертификате.**  **Значения:**   |  |  |  | | --- | --- | --- | | **Значения** | | **Описание** | | **KC\_CERTPROP\_ISSUER\_COUNTRYNAME** | 0x00000801 | Страна издателя | | **KC\_CERTPROP\_ISSUER\_SOPN** | 0x00000802 | Название штата или провинции издателя | | **KC\_CERTPROP\_ISSUER\_LOCALITYNAME** | 0x00000803 | Населённый пункт издателя | | **KC\_CERTPROP\_ISSUER\_ORG\_NAME** | 0x00000804 | Наименование организации издателя | | **KC\_CERTPROP\_ISSUER\_ORGUNIT\_NAME** | 0x00000805 | Название организационного подразделения издателя | | **KC\_CERTPROP\_ISSUER\_COMMONNAME** | 0x00000806 | Имя Фамилия издателя | | **KC\_CERTPROP\_SUBJECT\_COUNTRYNAME** | 0x00000807 | Страна субъекта | | **KC\_CERTPROP\_SUBJECT\_SOPN** | 0x00000808 | Название штата или провинции субъекта | | **KC\_CERTPROP\_SUBJECT\_LOCALITYNAME** | 0x00000809 | Населенный пункт субъекта | | **KC\_CERTPROP\_SUBJECT\_COMMONNAME** | 0x0000080a | Общее имя субъекта | | **KC\_CERTPROP\_SUBJECT\_GIVENNAME** | 0x0000080b | Имя субъекта | | **KC\_CERTPROP\_SUBJECT\_SURNAME** | 0x0000080c | Фамилия субъекта | | **KC\_CERTPROP\_SUBJECT\_SERIALNUMBER** | 0x0000080d | Серийный номер субъекта | | **KC\_CERTPROP\_SUBJECT\_EMAIL** | 0x0000080e | e-mail субъекта | | **KC\_CERTPROP\_SUBJECT\_ORG\_NAME** | 0x0000080f | Наименование организации субъекта | | **KC\_CERTPROP\_SUBJECT\_ORGUNIT\_NAME** | 0x00000810 | Название организационного подразделения субъекта | | **KC\_CERTPROP\_SUBJECT\_BC** | 0x00000811 | Бизнес категория субъекта | | **KC\_CERTPROP\_SUBJECT\_DC** | 0x00000812 | Доменный компонент субъекта | | **KC\_CERTPROP\_NOTBEFORE** | 0x00000813 | Дата действителен с | | **KC\_CERTPROP\_NOTAFTER** | 0x00000814 | Дата действителен по | | **KC\_CERTPROP\_KEY\_USAGE** | 0x00000815 | Использование ключа | | **KC\_CERTPROP\_EXT\_KEY\_USAGE** | 0x00000816 | Расширенное использование ключа | | **KC\_CERTPROP\_AUTH\_KEY\_ID** | 0x00000817 | Идентификатор ключа центра сертификации | | **KC\_CERTPROP\_SUBJ\_KEY\_ID** | 0x00000818 | Идентификатор ключа субъекта | | **KC\_CERTPROP\_CERT\_SN** | 0x00000819 | Серийный номер серификата | | **KC\_CERTPROP\_ISSUER\_DN** | 0x0000081a | Отличительное имя издателя | | **KC\_CERTPROP\_SUBJECT\_DN** | 0x0000081b | Отличительное имя субъекта | | **KC\_CERTPROP\_SIGNATURE\_ALG** | 0x0000081c | Алгоритм подписи | |
| **Константы, определяющие дополнительные условия выполнения операций. Используется как параметр в функциях.**  **Значения:**   |  |  |  | | --- | --- | --- | | **KC\_SIGN\_DRAFT** | 0x00000001 | Сырая подпись (draft sign) | | **KC\_SIGN\_CMS** | 0x00000002 | Подпись в формате CMS | | **KC\_IN\_PEM** | 0x00000004 | Входные данные в формате PEM | | **KC\_IN\_DER** | 0x00000008 | Входные данные в кодировке DER | | **KC\_IN\_BASE64** | 0x00000010 | Входные данные в кодировке BASE64 | | **KC\_IN2\_BASE64** | 0x00000020 | Дополнительные входные данные в кодировке BASE64 | | **KC\_DETACHED\_DATA** | 0x00000040 | Отсоединенная подпись | | **KC\_WITH\_CERT** | 0x00000080 | Вложить сертификат в подпись | | **KC\_WITH\_TIMESTAMP** | 0x00000100 | Добавить в подпись метку времени (не используется в текущей версии) | | **KC\_OUT\_PEM** | 0x00000200 | Выходные данные в формате PEM | | **KC\_OUT\_DER** | 0x00000400 | Выходные данные в кодировке DER | | **KC\_OUT\_BASE64** | 0x00000800 | Выходные данные в кодировке BASE64 | | **KC\_PROXY\_OFF** | 0x00001000 | Отключить использование прокси-сервера и стереть настройки. | | **KC\_PROXY\_ON** | 0x00002000 | Включить и установить настройки прокси-сервера (адрес и порт) | | **KC\_PROXY\_AUTH** | 0x00004000 | Прокси-сервер требует авторизацию (логин/пароль) | |
| **Константы, определяющие подробные коды ошибок времени выполнения. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString .**  **Значения:**   |  |  |  | | --- | --- | --- | | **KCR\_OK** | 0x00000000 | Нет ошибки | | **KCR\_ERROR\_READ\_PKCS12** | 0x08F00002 | Невозможно прочитать файл формата pkcs#12 | | **KCR\_ERROR\_OPEN\_PKCS12** | 0x08F00003 | Невозможно открыть файл формата pkcs12 | | **KCR\_INVALID\_PROPID** | 0x08F00004 | Недопустимый идентификатор расширения сертификата | | **KCR\_BUFFER\_TOO\_SMALL** | 0x08F00005 | Размер буфера слишком мал | | **KCR\_CERT\_PARSE\_ERROR** | 0x08F00006 | Невозможно разобрать (распарсить) сертификат | | **KCR\_INVALID\_FLAG** | 0x08F00007 | Недопустимый флаг | | **KCR\_OPENFILEERR** | 0x08F00008 | Невозможно открыть файл | | **KCR\_INVALIDPASSWORD** | 0x08F00009 | Неправильный пароль | | **KCR\_MEMORY\_ERROR** | 0x08F0000d | Невозможно выделить память | | **KCR\_CHECKCHAINERROR** | 0x08F0000e | Не найден сертификат УЦ или сертификат пользователя при проверки цепочки | | **KCR\_VALIDTYPEERROR** | 0x08F00010 | Недопустимый тип валидации сертификата | | **KCR\_BADCRLFORMAT** | 0x08F00011 | Некорректный формат CRL | | **KCR\_LOADCRLERROR** | 0x08F00012 | Невозможно загрузить CRL | | **KCR\_LOADCRLSERROR** | 0x08F00013 | Невозможно загрузить CRL-ы | | **KCR\_UNKNOWN\_ALG** | 0x08F00015 | Неизвестный алгоритм подписи | | **KCR\_KEYNOTFOUND** | 0x08F00016 | Не найден приватный ключ пользователя | | **KCR\_SIGN\_INIT\_ERROR** | 0x08F00017 | Невозможно инициализировать менеджера подписи | | **KCR\_SIGN\_ERROR** | 0x08F00018 | Не удалось сгенерировать цифровую подпись | | **KCR\_ENCODE\_ERROR** | 0x08F00019 | Ошибка шифрования | | **KCR\_INVALID\_FLAGS** | 0x08F0001a | Недопустимые флаги | | **KCR\_CERTNOTFOUND** | 0x08F0001b | Не найден сертификат пользователя | | **KCR\_VERIFYSIGNERROR** | 0x08F0001c | Ошибка верификации подписи xml | | **KCR\_BASE64\_DECODE\_ERROR** | 0x08F0001d | Ошибка дешифровки из Base 64 | | **KCR\_UNKNOWN\_CMS\_FORMAT** | 0x08F0001e | Неизвестный формат CMS | | **KCR\_CA\_CERT\_NOT\_FOUND** | 0x08F00020 | Не найден сертификат УЦ | | **KCR\_XMLSECINIT\_ERROR** | 0x08F00021 | Ошибка инициализации xmlsec | | **KCR\_LOADTRUSTEDCERTSERR** | 0x08F00022 | Ошибка загрузки доверенных сертификатов | | **KCR\_SIGN\_INVALID** | 0x08F00023 | Недопустимая подпись xml | | **KCR\_NOSIGNFOUND** | 0x08F00024 | Не найдена подпись во входных данных | | **KCR\_DECODE\_ERROR** | 0x08F00025 | Ошибка дешифрования | | **KCR\_XMLPARSEERROR** | 0x08F00026 | Невозможно разобрать (распарсить) xml | | **KCR\_XMLADDIDERROR** | 0x08F00027 | Не удалось добавить атрибут ID | | **KCR\_XMLINTERNALERROR** | 0x08F00028 | Ошибка при работе с xml | | **KCR\_XMLSETSIGNERROR** | 0x08F00029 | Не удалось подписать xml | | **KCR\_OPENSSLERROR** | 0x08F0002a | Ошибка openssl | | **KCR\_NOTOKENFOUND** | 0x08F0002c | Не найден токен | | **KCR\_OCSP\_ADDCERTERR** | 0x08F0002d | Не удалось добавить сертификат в ocsp | | **KCR\_OCSP\_PARSEURLERR** | 0x08F0002e | Не удалось разобрать url | | **KCR\_OCSP\_ADDHOSTERR** | 0x08F0002f | Не удалось добавить хост | | **KCR\_OCSP\_REQERR** | 0x08F00030 | Не удалось добавить текущее время в запрос | | **KCR\_OCSP\_CONNECTIONERR** | 0x08F00031 | Ошибка подключения к OCSP респондеру | | **KCR\_VERIFY\_NODATA** | 0x08F00032 | Нет входных данных для верификации | | **KCR\_IDATTR\_NOTFOUND** | 0x08F00033 | Не найден атрибут ID | | **KCR\_IDRANGE** | 0x08F00034 | Некорректный идентификатор | | **KCR\_READERNOTFOUND** | 0x08F00037 | Не найден ридер | | **KCR\_GETCERTPROPERR** | 0x08F00038 | Не удалось получить значение атрибута | | **KCR\_SIGNFORMMAT** | 0x08F00039 | Неизвестный формат подписи | | **KCR\_INDATAFORMAT** | 0x08F0003a | Неизвестный формат входных данных | | **KCR\_OUTDATAFORMAT** | 0x08F0003b | Неизвестный формат выходных данных | | **KCR\_VERIFY\_INIT\_ERROR** | 0x08F0003c | Невозможно инициализировать менеджера верификации подписи | | **KCR\_VERIFY\_ERROR** | 0x08F0003d | Не удалось верифицировать цифровую подпись | | **KCR\_HASH\_ERROR** | 0x08F0003e | Не удалось хэшировать данные | | **KCR\_SIGNHASH\_ERROR** | 0x08F0003f | Не удалось подписать хэшированные данные | | **KCR\_CACERTNOTFOUND** | 0x08F00040 | Не найден сертификат УЦ в хранилище сертификатов | | **KCR\_CERTTIMEINVALID** | 0x08F00042 | Срок действия сертификата истек либо еще не наступил | | **KCR\_CONVERTERROR** | 0x08F00043 | Ошибка записи сертификата в структуру X509 | | **KCR\_TSACREATEQUERY** | 0x08F00044 | Ошибка генерации запроса timestamp | | **KCR\_CREATEOBJ** | 0x08F00045 | Ошибка записи OID в ASN1 структуру | | **KCR\_CREATENONCE** | 0x08F00046 | Ошибка генерации уникального числа | | **KCR\_HTTPERROR** | 0x08F00047 | Ошибка протокола http | | **KCR\_CADESBES\_FAILED** | 0x08F00048 | Ошибка проверки расширения CADESBES в CMS | | **KCR\_CADEST\_FAILED** | 0x08F00049 | Ошибка проверки подписи токена TSA | | **KCR\_NOTSATOKEN** | 0x08F0004a | В подписи не присутствует метка TSA | | **KCR\_INVALID\_DIGEST\_LEN** | 0x08F0004b | Неправильная длина хэша | | **KCR\_GENRANDERROR** | 0x08F0004c | Ошибка генерации случайного числа | | **KCR\_SOAPNSERROR** | 0x08F0004d | Не найдены заголовки SOAP-сообщений | | **KCR\_GETPUBKEY** | 0x08F0004e | Ошибка экспорта публичного ключа | | **KCR\_GETCERTINFO** | 0x08F0004f | Ошибка получения информации о сертификате | | **KCR\_FILEREADERROR** | 0x08F00050 | Ошибка чтения файла | | **KCR\_CHECKERROR** | 0x08F00051 | Хэш не совпадает | | **KCR\_ZIPEXTRACTERR** | 0x08F00052 | Невозможно открыть архив | | **KCR\_NOMANIFESTFILE** | 0x08F00053 | Не найден MANIFEST | | **KCR\_LIBRARYNOTINITIALIZED** | 0x08F00101 | Библиотека не инициализирована | | **KCR\_ENGINELOADERR** | 0x08F00200 | Ошибка подключения (загрузки) модуля (engine) | | **KCR\_PARAM\_ERROR** | 0x08F00300 | Некорректные входные данные | | **KCR\_CERT\_STATUS\_OK** | 0x08F00400 | Статус сертификата – валидный. Используется при проверке сертификата по OCSP. (не является ошибкой, делается запись в лог) | | **KCR\_CERT\_STATUS\_REVOKED** | 0x08F00401 | Статус сертификата – отозван. Используется при проверке сертификата по OCSP. | | **KCR\_CERT\_STATUS\_UNKNOWN** | 0x08F00402 | Статус сертификата – неизвестен. Используется при проверке сертификата по OCSP. Например, не удалось установить издателя сертификата. | |

**Описание Функций криптографической библиотеки KalkanCrypt**

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Обзор** | **Описание** | **Возвращаемые значения** |
| **KalkanCrypt\_Init ().**  Инициализация библиотеки. | PHP\_FUNCTION KalkanCrypt\_Init (); | – | При успешном завершении возвращает 0. |
| **KalkanCrypt\_GetTokens ().**  Обеспечивает получение указателя на строку подключенных устройств типа storage и их количество. | PHP\_FUNCTION KalkanCrypt\_GetTokens ([in] INT storage, [out] CHAR\* tokens, [out] INT\* tCount); | Параметры:  [in] INT storage — тип хранилища;  [out] CHAR\* tokens — указатель на строку подключенных устройств;  [out] INT\* tCount — количество подключенных устройств. | При успешном завершении возвращает 0, в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_GetCertificatesList ().**  Обеспечивает получение списка сертификатов в виде строки и их количество. | PHP\_FUNCTION KalkanCrypt\_GetCertificatesList([out] CHAR\* certAlias, [out] INT\* count); | Параметры:  [out] CHAR\* certAlias — указатель на строку сертификатов;  [out] INT\* count — количество сертификатов. | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_LoadKeyStore ().**  Загрузка ключей/сертификата их хранилища. | PHP\_FUNCTION KalkanCrypt\_LoadKeyStore ([in] INT storage, [in] CHAR password, [in] CHAR container, [in] CHAR alias); | Параметры:  [in] INT storage — тип хранилища;  [in] CHAR password — пароль к хранилищу;  [in] CHAR container — название хранилища;  [in] CHAR alias — label (alias) сертификата. | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_X509LoadCertificateFromFile ().**  Загрузка сертификата из файла для дальнейшей работы с ним. | PHP\_FUNCTION KalkanCrypt\_X509LoadCertificateFromFile ([in] CHAR certPath, [in] INT certType); | Параметры:  [in] CHAR certPath — путь до файла сертификата;  [in] INT certType) — тип сертификата (см. KalkanCrypt Cert Types). | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_X509LoadCertificateFromBuffer ().**  Загрузка сертификата из памяти. | PHP\_FUNCTION KalkanCrypt\_X509LoadCertificateFromBuffer ([in] CHAR inCert, [in] INT flag); | Параметры:  [in] CHAR inCert — сертификат в виде строки;  [in] INT flag — флаги. Указывается кодировка сертификата (см. Кодировка сертификатов). | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_X509ExportCertificateFromStore ().**  Экспорт сертификата из хранилища. | PHP\_FUNCTION KalkanCrypt\_X509ExportCertificateFromStore ([in] CHAR alias, [in] INT flag, [out] CHAR \*outCert); | Параметры:  [in] CHAR alias — label (alias) сертификата;  [in] INT flag — флаги;  [out] CHAR\* outCert — указатель на начало сертификата. | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_X509CertificateGetInfo ().**  Обеспечивает получение значений полей/расширений из сертификата. Сертификат должен быть предварительно загружен с помощью одной из функций: LoadKeyStore(), X509LoadCertificateFromFile(),X509LoadCertificateFromBuffer(). | PHP\_FUNCTION KalkanCrypt\_X509CertificateGetInfo ([in] CHAR inCert, [in] INT propId, [out] CHAR\* outData); | Параметры:  [in] CHAR inCert — сертификат в виде строки;  [in] INT propId — идентификатор полей/расширений сертификата (см. KalkanCrypt CertPropID); [out] CHAR\* outData — указатель на значение указанного поля/расширения. | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_X509ValidateCertificate ().**  Осуществляет проверку сертификата: проверка срока действия, построение цепочки сертификатов, проверка отозванности по OCSP или CRL. | PHP\_FUNCTION KalkanCrypt\_X509ValidateCertificate ([in] CHAR inCert, [in] INT validType, [in] CHAR validPath, [in] \_\_int64 checkTime, [out] CHAR\* outInfo); | Параметры:  [in] CHAR inCert — сертификат в виде строки;  [in] INT validType — тип проверки (OCSP/CRL) (см. KalkanCrypt ValidType);  [in] CHAR validPath — путь до CRL или URL OCSP;  [in] \_\_int64 checkTime — дата и время, на момент которого необходимо провести проверку (зарезервировано, в текущей версии не используется);  [out] CHAR\* outInfo — выходная подробная информация о результате проверки. | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Если сертификат действителен, то GetLastError вернет 0, иначе, если сертификат недействителен, вернет код ошибки 0x00000017 (23). Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_SignData ().**  Подписывает данные. | PHP\_FUNCTION KalkanCrypt\_SignData ([in] CHAR alias, [in] INT flags, [in] CHAR inData, [out] CHAR\* outSign); | Параметры:  [in] CHAR alias — label (alias) сертификата;  [in] INT flags — флаги. Устанавливают формат входных/выходных данных, тип подписи (см. KalkanCrypt flags);  [in] CHAR inData — входные данные;  [out] CHAR\* outSign — выходные данные (подписи). | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_SignXML ().**  Подписывает данные в формате XML. | PHP\_FUNCTION KalkanCrypt\_SignXML ([in] CHAR alias, [in] INT flags, [in] CHAR signNodeId, [in] CHAR parentSignNode, [in] CHAR parentNameSpace, [in] CHAR inData, [out] CHAR\* outSign); | Параметры:  [in] CHAR alias — label (alias) сертификата;  [in] INT flags — флаги;  [in] CHAR signNodeId — идентификатор тэга, который необходимо подписать. Не указывается, если необходимо подписать все содержимое документа;  [in] CHAR parentSignNode — идентификатор тэга, в который необходимо поместить значение подписи;  [in] CHAR parentNameSpace — пространство имен тэга, в который необходимо поместить значение подписи. Если пространство имен есть, но не будет указано - то тег не найдется;  [in] CHAR inData — входные данные;  [out] CHAR\* outSign — выходные данные (подписи). | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_VerifyData ().**  Обеспечивает проверку подписи. | PHP\_FUNCTION KalkanCrypt\_VerifyData ([in] CHAR alias, [in] INT flags, [in] INT inCertID, [in] CHAR inData, [in] CHAR inSign, [out] CHAR\* outData, [out] CHAR\* outVerifyInfo, [out] CHAR\* outCert); | Параметры:  [in] CHAR alias — label (alias) сертификата;  [in] INT flags — флаги;  [in] INT inCertID — идентификатор (порядковый номер) сертификата (начинается с 0);  [in] CHAR inData — входные данные;  [in] CHAR inSign — подписанные входные данные;  [out] CHAR\* outData — выходные данные;  [out] CHAR\* outVerifyInfo — выходная подробная информация о результате проверки подписи;  [out] CHAR\* outCert — указатель на начало сертификата с подробной информацией. | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_VerifyXML ().**  Обеспечивает проверку подписи данных в формате XML. | PHP\_FUNCTION KalkanCrypt\_VerifyXML ([in] CHAR alias, [in] INT flags, [in] CHAR inData, [out] CHAR\* outVerifyInfo); | Параметры:  [in] CHAR alias — label (alias) сертификата;  [in] INT flags — флаги;  [in] CHAR inData — входные данные;  [out] CHAR\* outVerifyInfo — выходная подробная информация о результате проверки подписи. | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_GetLastError ().**  Обеспечивает получение подробного кода ошибки, возникшей в процессе выполнения функций криптопровайдера KalkanCryptCOM. | PHP\_FUNCTION KalkanCrypt\_GetLastError ([out, retval] LONG\* rv); | Параметры:  [out, retval] LONG\* rv — возвращаемый код последней ошибки. | Возвращает 0, если не найдена ошибка, иначе — значение > 0 (см. KalkanCrypt Errors).  При вызове очищается протокол работы криптопровайдера и значение кода последней ошибки. |
| **KalkanCrypt\_GetLastErrorString ().**  Обеспечивает получение подробного протокола работы функций криптопровайдера KalkanCryptCOM. | PHP\_FUNCTION KalkanCrypt\_GetLastErrorString ([out] CHAR\* errorString, [out] LONG\* rv); | Параметры:  [out] CHAR\* errorString — подробный протокол работы функций криптопровайдера;  [out] LONG\* rv — код последней ошибки. | При успешном завершении возвращает 0,в противном случае – 1.  При вызове очищается протокол работы криптопровайдера и значение кода последней ошибки. |
| **KalkanCrypt\_getSigAlgFromXML ().**  Обеспечивает получение алгоритма подписи из XML. | PHP\_FUNCTION KalkanCrypt\_getSigAlgFromXML ([in] CHAR inXML, [out] CHAR\* outSigAlg); | Параметры:  [in] CHAR inXML — входные данные в формате XML;  [out] CHAR\* outSigAlg — алгоритм подписи. | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_getCertFromCMS ().**  Обеспечивает получение сертификата из CMS. | PHP\_FUNCTION KalkanCrypt\_getCertFromCMS([in] CHAR inCMS, [in] INT inSignID, [in] INT flsgs, [out] CHAR\* outCert); | Параметры:  [in] CHAR inCMS — входные данные в формате XML;  [in] INT inSignID — идентификатор сертификата;  [in] INT flags — флаги;  [out] CHAR\* outCert — указатель на начало сертификата. | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_getCertFromXML ().**  Обеспечивает получение сертификата из XML. | PHP\_FUNCTION KalkanCrypt\_getCertFromXML ([in] CHAR inXML, [in] INT inSignID, [out] CHAR\* outCert); | Параметры:  [in] CHAR inXML — входные данные в формате XML;  [in] INT inSignID — идентификатор сертификата;  [out] CHAR\* outCert — указатель на начало сертификата. | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_XMLFinalize ().**  Освобождает память и завершает работу библиотеки с модулями, отвечающие за парсинг, подпись и проверку данных в формате XML. | PHP\_FUNCTION KalkanCrypt\_XMLFinalize (); | – | Не надо вызывать каждый раз при подписи. Можно только один раз после цикла подписания xml файлов. |
| **KalkanCrypt\_Finalize ().**  Освобождает ресурсы криптопровайдера KalkanCryptCOM и завершает работу библиотеки. | PHP\_FUNCTION KalkanCrypt\_Finalize (); | – | — |
| **KalkanCrypt\_HashData ().**  Хэширует данные. | PHP\_FUNCTION KalkanCrypt\_HashData ([in] CHAR algorithm, [in] INT flags, [in] CHAR inData, [out] CHAR\* outData); | [in] CHAR algorithm –алгоритм хэширования (строка “sha256” или “Gost34311\_95”);  [in] INT flags – флаги;  [in] CHAR inData – входные данные;  [out] CHAR\* outData – выходные хэшированные данные. | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_SignHash ().**  Подписывает входные хэшированные данные. | PHP\_FUNCTION KalkanCrypt\_SignHash ([in] CHAR alias, [in] INT flags, [in] CHAR inHash, [out] CHAR\* outSign). | [in] CHAR alias - label (alias) сертификата;  [in] INT flags – флаги;  [in] CHAR inData – входные хэшированные данные;  [out] CHAR\* outData – выходные данные. | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_TSASetUrl ().**  Установка адреса сервиса TSA. | PHP\_FUNCTION KalkanCrypt\_TSASetUrl ([in] CHAR tsaurl); | [in] CHAR tsaurl – адрес сервиса TSA | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString .  Значение по умолчанию http://tsp.pki.gov.kz:80 |
| **KalkanCrypt\_getTimeFromSig ()**  Получить время подписи. | PHP\_FUNCTION KalkanCrypt\_getTimeFromSig ([in] CHAR inData, [in] INT flags, [in] INT sigId, [out] LONGLONG\* outDateTime); | [in] CHAR inData – входные данные (подпись, в текущей версии только формата CAdES);  [in] INT flags – флаги  [in] INT sigId – идентификатор подписи (в текущей версии не используется)  [out] LONGLONG\* outDateTime – выходные данные, время подписи UTC в Unix формате (отсчет от 01/01/1970) | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **SignDataBytes()**  Подписать бинарные данные. | PHP\_FUNCTION SignDataBytes([in] CHAR alias, [in] INT flags, [in] BYTE\* inData, [in] INT inDataLength, [out] CHAR\* outSign); | [in] CHAR alias - label (alias) сертификата;  [in] INT flags – флаги;  [in] BYTE\* inData - входные данные;  [in] INT inDataLength – длина входных данных;  [out] CHAR\* outSign – выходные данные (подпись); | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_SetProxy ()**  Установить настройки прокси-сервера. | PHP\_FUNCTION KalkanCrypt\_SetProxy ([in] INT flags, [in] CHAR inProxyAddr, [in] CHAR inPort, [in] CHAR inUserName, [in] CHAR inUserPass); | [in] INT flags – флаги;  [in] CHAR inProxyAddr – адрес прокси-сервера;  [in] CHAR inPort – порт;  [in] CHAR inUserName – имя пользователя;  [in] CHAR inUserPass – пароль. | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **KalkanCrypt\_SignWSSE ().**  Подписывает данные в формате WSSE. | PHP\_FUNCTION KalkanCrypt\_SignWSSE ([in] CHAR alias, [in] INT flags,  [in] CHAR inData, [out] CHAR\* outSign), [out] CHAR signNodeId; | Параметры:  [in] CHAR alias — label (alias) сертификата;  [in] INT flags — флаги;  [in] CHAR inData — входные данные;  [out] CHAR\* outSign — выходные данные (подписи);  [out] CHAR signNodeId — идентификатор тэга, который необходимо подписать. Не указывается, если необходимо подписать все содержимое документа. | При успешном завершении возвращает 0,в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **ZipConSign()**  Подпись файлов с последующим размещением их в zip-контейнер | PHP\_FUNCTION KalkanCrypt\_ZipConSign ([in] CHAR alias, [in] CHAR inFiles, [in] CHAR zipName, [in] CHAR outDir, [in] INT flags); | [in] CHAR alias - label (alias) сертификата;  [in] CHAR inFiles – файлы, которые необходимо записать.(В конце каждого пути к файлу необходимо вставить вертикальную линию - «|», например: ***inFiles=*****“ test/1.pdf|test/2.txt|”);**  [in] CHAR zipName – имя создаваемого архива;  [in] CHAR outDir – расположение создаваемого архива:  [in]INT flags – флаг (В текущей версии не используется). | При успешном завершении возвращает 0, в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |
| **ZipConVerify()**  Проверка подписи электронных документов (zip-контейнер) | PHP\_FUNCTION KalkanCrypt\_ZipConVerify (  [in] CHAR inZipFile, [in] INT flags,  [out] CHAR \* outVerifyInfo) | [in] CHAR inZipFile – имя ZIP-файла  [in] INT flags – флаги(В текущей версии не используется);  [out] CHAR \* outVerifyInfo - подробная информация о результате проверки подписи. | При успешном завершении возвращает 0, в противном случае – 1. Код ошибки можно получить с помощью функции KalkanCrypt\_GetLastError. Подробный текст ошибки можно получить с помощью функции KalkanCrypt\_GetLastErrorString . |