

ООО «ВАЛИДАТА»

УТВЕРЖДЕН
ВАМБ.00096-06-ЛУ

**СРЕДСТВО КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ
«ВАЛИДАТА КРИПТОСЕРВЕР» ВЕРСИЯ 4**

БИБЛИОТЕКА ПРИКЛАДНОГО ПРОГРАММНОГО ИНТЕРФЕЙСА
КРИПТОГРАФИЧЕСКОГО СЕРВЕРА ДЛЯ ПЛАТФОРМ «JAVA» И «IBM
WEBSPPHERE APPLICATION SERVER»

Руководство программиста

ВАМБ.00096-06 33 02

2020

Аннотация

Данный документ содержит описание библиотеки прикладного программного интерфейса криптографического сервера для платформ «Java» и «IBM WebSphere Application Server» (далее — ППИ Java), входящей в состав программного комплекса (ПК) ВАМБ.00096-06 «Средство криптографической защиты информации «Валидата Криптосервер» версия 4» (далее по тексту — СКЗИ «Валидата Криптосервер»).

Документ предназначен для разработчиков прикладного программного обеспечения (ППО) как руководство по встраиванию и использованию ППИ Java при работе с КС.

При встраивании библиотеки предполагается, что системный программист имеет знания о существующей архитектуре системы сертификатов ключей проверки электронной подписи (ЭП), используемых рекомендациях и стандартах.

Содержание

1	БИБЛИОТЕКА ПРИКЛАДНОГО ПРОГРАММНОГО ИНТЕРФЕЙСА ДЛЯ ПЛАТФОРМ JAVA И IBM WEBSPPHERE APPLICATION SERVER	5
1.1	Назначение	5
1.2	Характеристики библиотеки	6
1.3	Использование библиотеки	6
1.3.1	Условия, необходимые для использования библиотеки	6
1.3.2	Состав библиотеки	7
1.4	Описание библиотеки	8
1.4.1	Типы	8
1.4.2	Использование классов	9
1.4.3	Кодирование и декодирование данных	10
1.4.4	Константы	10
1.4.5	Перечисления	15
1.4.6	Общие классы	16
1.4.7	Классы контекстов потоковых функций	16
1.4.8	Классы, уникально идентифицирующие сертификат	16
1.4.9	Классы, описывающие сертификат	17
1.4.10	Классы, описывающие САС	20
1.4.11	Параметры выполнения функций	21
1.4.12	Результаты выполнения функций	25
1.5	Описание функций	29
1.5.1	Функции инициализации и деинициализации	29
1.5.2	Функции получения описаний ошибок	29
1.5.3	Функции экспорта и импорта объектов СУС	29
1.5.4	Функции разбора и получения информации об объектах СУС	30
1.5.5	Функции построения и проверки цепочек объектов СУС	31
1.5.6	Функции вычисления хэш-значений	32
1.5.7	Функции вычисления и проверки ЭП хэш-значений	33
1.5.8	Функции поиска сертификатов	33
1.5.9	Функции блочного вычисления совмещенной ЭП CMS-сообщений	34
1.5.10	Функции потокового вычисления совмещенной ЭП CMS-сообщений	34
1.5.11	Функции блочного вычисления отделенной ЭП CMS-сообщений	35
1.5.12	Функции потокового вычисления отделенной ЭП CMS-сообщений	35
1.5.13	Функции блочной проверки совмещенных ЭП CMS-сообщений	36
1.5.14	Функции потоковой проверки совмещенных ЭП CMS-сообщений	37
1.5.15	Функции блочной проверки отделенных ЭП CMS-сообщений	38
1.5.16	Функции потоковой проверки отделенных ЭП CMS-сообщений	38
1.5.17	Функции блочного зашифрования CMS-сообщений	39
1.5.18	Функции потокового зашифрования CMS-сообщений	40
1.5.19	Функции блочного расшифрования CMS-сообщений	41
1.5.20	Функции потокового расшифрования CMS-сообщений	41

1.5.21	Функции блочного преобразования отделенных ЭП в совмещенные	42
1.5.22	Функции блочного преобразования совмещенных ЭП в отделенные	43
1.5.23	Функции потокового преобразования совмещенных ЭП в отделенные	43
1.5.24	Функции блочного получения информации о CMS-сообщениях	44
1.5.25	Функции потокового получения информации о CMS-сообщениях	45
1.5.26	Функции блочной простановки и проверки штампов времени .	46
1.5.27	Функции потоковой простановки и проверки штампов времени	47
1.5.28	Функции получения online-статуса сертификата	50
1.5.29	Функции выработки случайного числа заданной длины	51
2	ОПИСАНИЕ ОШИБОЧНЫХ СИТУАЦИЙ	52
	ПЕРЕЧЕНЬ СОКРАЩЕНИЙ	58
	ПЕРЕЧЕНЬ ТАБЛИЦ	60

1 БИБЛИОТЕКА ПРИКЛАДНОГО ПРОГРАММНОГО ИНТЕРФЕЙСА ДЛЯ ПЛАТФОРМ JAVA И IBM WEBSHERE APPLICATION SERVER

1.1 Назначение

Библиотека прикладного программного интерфейса (ППИ) криптографического сервера (КС) для платформ Java и IBM WebSphere Application Server (ППИ Java) предназначена для предоставления программного интерфейса, выполняемого с помощью Java JRE (среда выполнения) версия 1.6, 1.7, 1.8 или более новые, и/или IBM WebSphere Application Server (сервер приложений) версия 8.0, 8.5, 8.5.5 и более новые, к функциям КС.

Библиотека ППИ КС обеспечивает удаленный доступ к функциям аутентификации в соответствии с рекомендациями X.509, формирования электронной подписи (ЭП) по ГОСТ Р 34.10-2012, ГОСТ Р 34.11-2012, проверки ЭП по ГОСТ Р 34.10-2001, ГОСТ Р 34.10-2012, ГОСТ Р 34.11-94, ГОСТ Р 34.11-2012 и шифрования и вычисления имитовставки по ГОСТ Р 34.12-2015 (блочные шифры «Магма» и «Кузнечик»), ГОСТ 28147-89.

Примечания

1 При использовании блочных функций ППИ Java обеспечивает выполнение функций зашифрования/расшифрования и создания/проверки ЭП под файлом или областью памяти объёмом максимум до 2 Гбайт. При этом данные величины могут быть уменьшены в зависимости от текущего использования и гранулированности виртуальной памяти вызывающего процесса.

2 При использовании потоковых функций ППИ Java обеспечивает выполнение функций зашифрования/расшифрования и создания/проверки ЭП под файлом или областью памяти без ограничения общего объема.

Удаленный доступ к функциям КС осуществляется посредством использования протокола DCE-RPC поверх протокола TCP/IP. Реализация функций КС, доступных посредством библиотеки ППИ Java, основана на использовании криптографического ядра ПК ВАНБ.00060-06 «СКЗИ «Валидата CSP» версия 6» (далее — СКЗИ «Валидата CSP»).

Библиотека ППИ Java обеспечивает удаленный доступ к следующим функциям КС:

- зашифрование и расшифрование файла;
- зашифрование и расшифрование области памяти;
- создание ЭП файла;
- создание ЭП области памяти;
- проверка ЭП файла;
- проверка ЭП области памяти;
- удаление ЭП из файла;
- удаление ЭП из области памяти;

- выработка хэш-значения для файла;
- выработка хэш-значения для области памяти;
- преобразование отделенной и совмещенной ЭП;
- выбор и инициализация работы с КС;
- аутентификация при работе КС;
- создание ЭП хэш-функции данных;
- проверка ЭП хэш-функции данных;
- выработка случайного числа заданной длины.

1.2 Характеристики библиотеки

В качестве алгоритма ЭП используется асимметричный вариант ЭП, а именно криптосистема с двумя ключевыми элементами - открытым (общедоступным) и закрытым - по ГОСТ Р 34.10-2001 (только в части проверки ЭП) и ГОСТ Р 34.10-2012. Для формирования хэш-функции сообщения используются алгоритмы по ГОСТ Р 34.11-94 и ГОСТ Р 34.11-2012. Шифрование информации и вычисление имитовставки выполняется по ГОСТ Р 34.12-2015 (блочные шифры «Магма» и «Кузнечик») и ГОСТ 28147-89.

Библиотека ППИ Java предназначена для работы в операционных системах (ОС), указанных в документе ВАМБ.00096-06 30 01 «СКЗИ «Валидата Криптосервер» версия 4. Формуляр».

1.3 Использование библиотеки

В данном документе дано краткое описание методов и классов с указанием соответствующих констант, структур и функций из библиотеки прикладного программного интерфейса криптографического сервера для C/C++.

Для полного описания методов и классов необходимо изучить описание соответствующих констант, структур и функций в документе ВАМБ.00096-06 33 01 «СКЗИ «Валидата Криптосервер» версия 4. Руководство программиста».

1.3.1 Условия, необходимые для использования библиотеки

При использовании библиотеки ППИ Java необходимо соблюдать следующие условия:

- библиотека предназначена для использования в приложениях, написанных на языке программирования Java и собираемых с помощью Java SDK (средство разработки) версия 1.6, 1.7, 1.8 или более новые;
- перед началом использования библиотеки для обращения к функциям КС необходимо запустить КС и загрузить в него криптографические ключи и необходимые сертификаты;
- максимальный объем блоков данных, которые могут передаваться в функции библиотеки для передачи на КС, составляет 2 Гбайт.

1.3.2 Состав библиотеки

В состав библиотеки ППИ Java входят, в том числе, следующие файлы (пути указаны относительно каталога установки ППИ Java):

- **CryptoServerLibrary.jar** - архив, содержащий собственно библиотеку ППИ Java;
- **jarapac/jarapac.jar** - архив, содержащий транспортно-независимую часть библиотеки Java DCE-RPC;
- **jarapac/ncacn_ip_tcp.jar** - архив, содержащий транспортно-зависимую часть библиотеки Java DCE-RPC;
- **jarapac/lib/jcifs-1.1.2.jar** - архив, содержащий библиотеку Java реализации протоколов CIFS/SMB;
- **CRSRVTest/CryptoServerTest.class** - тестовая утилита, предназначенная для циклического последовательного вызова функций библиотеки ППИ Java;
- **CRSRVTest/CryptoServerUtil.class** - командная тестовая утилита;
- **run.cmd** - командная процедура, позволяющая запускать тестовые утилиты в ОС Windows;
- **run.sh** - командная процедура, позволяющая запускать тестовые утилиты в ОС zLinux и в ОС Linux.

Для корректного функционирования библиотеки необходимо добавить полные пути ко всем вышеперечисленным архивам в хранилище CLASSPATH, используемое Java JRE и/или IBM WebSphere Application Server.

Поскольку архивы CryptoServerLibrary.jar, jarapac/jarapac.jar и jarapac/ncacn_ip_tcp.jar используют классы (Java.io.File и другие), осуществляющие доступ к файлам, свойствам и TCP/IP сокетам, при использовании Java 2 Security Manager в IBM WebSphere Application Server необходимо внести данные архивы в список разрешенных к загрузке посредством редактирования двух конфигурационных файлов:

1. Файл политик библиотек IBM WebSphere Application Server `${install_root}/config/cells/${cell_name}/nodes/${node_name}/library.policy`:

```
grant {
    permission java.net.SocketPermission "${validataHost}:1333", "
connect, resolve";
    permission java.io.FilePermission "/${validataRoot}/jarapac/
ncacn_ip_tcp.jar", "read";
    permission java.io.FilePermission "\\${validataTemp}\\-", "read
";
    permission java.util.PropertyPermission "*", "read, write";
};
```

2. Файл политик приложения IBM WebSphere Application Server (имя файла зависит от имени приложения, в нашем случае - CryptoServerTest.ear) `${install_root}/config/cells/${cell_name}/applications/CryptoServerTest.ear/deployments/CryptoServerTest/META-INF/was.policy`:

```
grant codeBase "file:${application}" {
```

```

    permission java.io.FilePermission "/${validataRoot}/jarapac/
ncacn_ip_tcp.jar", "read";
    permission java.io.FilePermission "\\${validataTemp}\\-", "read
";
    permission java.util.PropertyPermission "*", "read, write";
};

```

Переменные `validataRoot` (путь к каталогу установки библиотеки ППИ Java, например, `C:/Validata/Java`), `validataHost` (имя узла КС или NLB-кластера КС, например, `rpc`) и `validataTemp` (путь к каталогу с рабочими файлами, например, `C:\Validata\Temp`) можно определить с помощью Administrative Console (Консоли управления), находящейся в *Servers->Application Servers->\${server_name}->Java and Process Management->Process Definition->Java Virtual Machine->Custom Properties*. Заданные при выполнении настройки значения будут сохранены в конфигурационном файле сервера IBM WebSphere Application Server `${install_root}/config/cells/${cell_name}/nodes/${node_name}/servers/${server_name}/server.xml`.

Данные действия необходимо повторить на каждом используемом сервере. Обратите внимание, что разрешения `java.net.SocketPermission` и `java.io.FilePermission` должны быть указаны для работы приложения (в нашем случае - `CryptoServerTest.ear`), тогда как разрешения `java.util.PropertyPermission` указаны для того, чтобы избежать записи полученных исключительных ситуаций в файл протоколов.

1.4 Описание библиотеки

1.4.1 Типы

При использовании библиотеки ППИ Java необходимо обратить внимание на следующее:

- при использовании функций библиотеки ППИ Java, обеспечивающих удаленный доступ к функциям КС, возникновение ошибочной ситуации может быть показано либо с помощью ненулевого кода возврата, выдаваемого функциями по их завершении, либо при возникновении исключительной ситуации (Java Exception), выбрасываемой во время выполнения функций;
- тип кода возврата, выдаваемого функциями библиотеки ППИ Java после их завершения, соответствует встроенному типу *int* языка программирования Java. Присутствие ненулевого кода возврата обычно обозначает ошибку, произошедшую на КС во время выполнения запрошенной функции, и наиболее часто возникающую при передаче неправильных параметров в запрошенную функцию;
- тип исключительной ситуации (Java Exception) соответствует стандартному типу *IOException* языка программирования Java. Выбрасывание исключительной ситуации обычно означает некорректную работу протоколов DCE-RPC и/или TCP/IP, и требует проверки наличия и работоспособности подсоединения компьютеров сервера прикладного ПО и КС посредством сегмента локальной вычислительной сети. При получении ошибки в виде исключительной ситуации следует прекратить использование существующего объекта (контекста) библиотеки и создать вместо него новый;

- тип всех строковых данных, используемых в функциях библиотеки, соответствует стандартному типу *String* языка программирования Java. Следует обратить внимание на то, что КС всегда принимает и возвращает строковые данные, содержащие символы русского языка, в кодировке Microsoft Windows 1251 (Code Page 1251), которые библиотека ППИ Java получает из или преобразует во внутреннее представление типа *String*;
- тип всех констант-битовых масок, используемых в функциях библиотеки, соответствует встроенному типу *int* языка программирования Java.

1.4.2 Использование классов

Поскольку требования языка программирования Java диктуют использование классов (Java Classes), все предоставляемые библиотекой ППИ Java типы, функции и константы доступны посредством использования основного класса библиотеки *CryptoServerLibrary*, находящегося в пакете (Java Package) *CRSRVLib*. Необходимо обратить внимание, что все составные типы, используемые в функциях библиотеки, также организованы как подклассы основного класса библиотеки *CryptoServerLibrary*. Перед началом использования библиотеки необходимо создать объект (Java Class Instance) основного класса библиотеки, используя предоставленный для этого конструктор класса (Java Class Constructor) *CryptoServerLibrary(String serverName, Properties properties)*. В качестве параметра *serverName* (Имя Сервера) используется строка формата <DNS Name/IP Address>[<TCP Port>]. Параметр *properties* (значение которого может быть *null*) может содержать нижеследующие конфигурационные значения (логические значения передаются как строки "true" или "false", а числовые значения как строки вида "12345"):

- *CRSRVLib.bEnableAPIDebug* (логическое, по умолчанию == *false*) - значение *true* включает режим выдачи отладочных сообщений при вызове функций библиотеки ППИ Java. Не рекомендуется использовать значение *true* при нагрузке ввиду большого объема выдаваемых отладочных сообщений;
- *CRSRVLib.sDebuggingOutput* (строковое, по умолчанию == "") - имя файла, в конец которого будут дописываться отладочные сообщения, выдаваемые библиотекой ППИ Java. Если значение равно "", то отладочные сообщения записываются в стандартный поток вывода (*System.out*);
- *CRSRVLib.bUseNotesAddress* (логическое, по умолчанию == *false*) - значение *false* необходимо для корректного кодирования данных в системах, совместимых со СКАД "Сигнатура". Значение *true* необходимо для корректного кодирования данных в системах, совместимых с ПК "Валидата Клиент";
- *CRSRVLib.iSocketMaximumCalls* (числовое, ≥ 0 , по умолчанию == 0) - значение, указывающее максимальное количество DCE-RPC вызовов, после выполнения которых библиотека автоматически выполняет переподсоединение к серверу. Если значение равно 0, то переподсоединение не производится;
- *CRSRVLib.iSocketInactivityInterval* (числовое, ≥ 0 , по умолчанию == 0) - значение (в секундах) максимального интервала между DCE-RPC вызовами, после истечения которого библиотека автоматически выполняет переподсоединение к серверу. Если значение равно 0, то переподсоединение не производится;
- *rpc.ncasn_ip_tcp.bEnableSocketIODebug* (логическое, по умолчанию ==

false) - значение *true* включает режим выдачи отладочных сообщений при выполнении DCE-RPC вызовов. Не рекомендуется использовать значение *true* при нагрузке ввиду большого объема выдаваемых отладочных сообщений;

– *rpc.ncacn_ip_tcp.bEnableSocketKeepAlive* (логическое, по умолчанию == *true*) - значение *true* включает режим TCP KeepAlive (периодическая посылка данных для контроля состояния) подключения к серверу;

– *rpc.ncacn_ip_tcp.iSocketIOTimeout* (числовое, ≥ 0 , по умолчанию == 0) - значение (в секундах) максимального времени выполнения операции TCP ввода-вывода (подсоединение, чтение и запись данных), после истечения которого будет выброшена исключительная ситуация *SocketTimeoutException*. Если значение равно 0 , то установка максимального времени выполнения не производится.

Например, чтобы подсоединиться к КС, находящемуся на узле с именем *grs* и ожидающему запросов на подсоединение к TCP порту 1333, необходимо создать объект основного класса библиотеки следующим образом: *CRSRVLib.CryptoServerLibrary crsrvLib = new CRSRVLib.CryptoServerLibrary("rpc[1333]", null)*. При создании объекта основного класса библиотеки инициализируется контекст (внутри данного объекта), полностью описывающий данное подключение к КС. При необходимости использования нескольких подключений к КС (или к NLB-кластеру КС) одновременно, следует создать требуемое число объектов (по одному на подключение) основного класса библиотеки. При необходимости использования библиотеки из нескольких потоков приложения необходимо обеспечить эксклюзивное использование каждого контекста конкретным потоком, т.е. не допускается одновременное использование данного контекста несколькими потоками приложения.

1.4.3 Кодирование и декодирование данных

Для обмена данными с КС используется протокол DCE-RPC поверх протокола TCP/IP. Все входные данные, передаваемые в функции и возвращаемые из функций библиотеки ППИ Java, кодируются в соответствии со спецификацией Distributed Computing Environment версия 1.2 (DCE версия 1.2). При заполнении входных данных необходимо выполнить следующее условие, связанное с особенностями процедуры кодирования: все ссылки на классы (и все ссылки из них и всех их подклассов), передаваемые как входные данные в функции библиотеки ППИ Java, должны быть не равны *null* и должны ссылаться на существующие объекты соответствующих классов. Исключение составляют только ссылки на массивы и объекты стандартного типа *String* языка Java. Невыполнение этого требования может привести к выбрасыванию исключительной ситуации *java.lang.NullPointerException*. Например, при заполнении параметра *auth* (класса *CRSRVLib.CryptoServerLibrary.mem_blk_t*), используемого в функции *CRSRVLib.VCERTR_Authorize*, ссылка на объект класса должна быть не равна *null*. В тоже время, ссылка на массив *buf* из данного объекта может быть равна *null*.

1.4.4 Константы

Возможные значения алгоритма открытого ключа сертификата (член *algorithm* класса *CRSRVLib.CryptoServerLibrary.certificate_t*):

String CRSRVLib.CryptoServerLibrary.**ALGORITHM_GOST_R_34_10_12_256** = ("GOST R 34.10-2012 (256 bit)");

Открытый ключ сертификата соответствует ГОСТ Р 34.10-2012 (256 бит).

String CRSRVLib.CryptoServerLibrary.**ALGORITHM_GOST_R_34_10_12_512** = ("GOST R 34.10-2012 (512 bit)");

Открытый ключ сертификата соответствует ГОСТ Р 34.10-2012 (512 бит).

String CRSRVLib.CryptoServerLibrary.**ALGORITHM_GOST_R_34_10_2001** = ("GOST R 34.10-2001");

Открытый ключ сертификата соответствует ГОСТ Р 34.10-2001.

Битовые маски, обозначающие заполненные поля сертификата (член `fields` класса `CRSRVLib.CryptoServerLibrary.certificate_t`), соответствуют флагам `FIELD_XXX` :

int CRSRVLib.CryptoServerLibrary.**FIELD_SERIAL** = (1 << 0);

int CRSRVLib.CryptoServerLibrary.**FIELD_ISSUER** = (1 << 1);

int CRSRVLib.CryptoServerLibrary.**FIELD_SUBJECT** = (1 << 2);

int CRSRVLib.CryptoServerLibrary.**FIELD_NOTBEFORE** = (1 << 5);

int CRSRVLib.CryptoServerLibrary.**FIELD_NOTAFTER** = (1 << 6);

int CRSRVLib.CryptoServerLibrary.**FIELD_KEYUSAGE** = (1 << 7);

int CRSRVLib.CryptoServerLibrary.**FIELD_ISSUERALTNAME** = (1 << 8);

int CRSRVLib.CryptoServerLibrary.**FIELD_SUBJECTALTNAME** = (1 << 9);

int CRSRVLib.CryptoServerLibrary.**FIELD_EXTKEYUSAGE** = (1 << 10);

int CRSRVLib.CryptoServerLibrary.**FIELD_POLICY** = (1 << 11);

int CRSRVLib.CryptoServerLibrary.**FIELD_EXTENSIONS** = (1 << 12);

int CRSRVLib.CryptoServerLibrary.**FIELD_NOTBEFOREPRIVATE** = (1 << 13);

int CRSRVLib.CryptoServerLibrary.**FIELD_NOTAFTERPRIVATE** = (1 << 14);

int CRSRVLib.CryptoServerLibrary.**FIELD_KEYID** = (1 << 15);

int CRSRVLib.CryptoServerLibrary.**FIELD_CERTENCODED** = (1 << 16);

int CRSRVLib.CryptoServerLibrary.**FIELD_CERTHASH** = (1 << 17);

int CRSRVLib.CryptoServerLibrary.**FIELD_ALGORITHM** = (1 << 18);

int CRSRVLib.CryptoServerLibrary.**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);

int CRSRVLib.CryptoServerLibrary.**FIELD_ALGORITHM_OID_FORMAT** = (1 << 31);

Битовые маски, обозначающие заполненные поля САС (член `fields` класса `CRSRVLib.CryptoServerLibrary.crl_t`), соответствуют флагам `FIELD_CRL_XXX` :

int CRSRVLib.CryptoServerLibrary.**FIELD_CRL_ISSUER** = (1 << 0);

int CRSRVLib.CryptoServerLibrary.**FIELD_CRL_LASTUPDATE** = (1 << 1);

int CRSRVLib.CryptoServerLibrary.**FIELD_CRL_NEXTUPDATE** = (1 << 2);

int CRSRVLib.CryptoServerLibrary.**FIELD_CRL_NUMBER** = (1 << 3);

int CRSRVLib.CryptoServerLibrary.**FIELD_CRL_REVOKED** = (1 << 4);

int CRSRVLib.CryptoServerLibrary.**FIELD_CRL_CRL ENCODED** = (1 << 5);

```
int CRSRVLib.CryptoServerLibrary.FIELD_CRL_CRLHASH = (1 « 6);
int CRSRVLib.CryptoServerLibrary.FIELD_CRL_ALL = (FIELD_CRL_ISSUER |
FIELD_CRL_LASTUPDATE | FIELD_CRL_NEXTUPDATE | FIELD_CRL_NUMBER |
FIELD_CRL_REVOKED | FIELD_CRL_CRL ENCODED | FIELD_CRL_CRLHASH);
```

Битовые маски возможных областей использования открытого ключа сертификата (член keyUsage класса CRSRVLib.CryptoServerLibrary.certificate_t), соответствуют флагам KEYUSAGE_XXX :

```
int CRSRVLib.CryptoServerLibrary.KEYUSAGE_DIGITAL_SIGNATURE = (1 « 0);
int CRSRVLib.CryptoServerLibrary.KEYUSAGE_NON_REPUDIATION = (1 « 1);
int CRSRVLib.CryptoServerLibrary.KEYUSAGE_KEY_ENCIPHERMENT = (1 « 2);
int CRSRVLib.CryptoServerLibrary.KEYUSAGE_DATA_ENCIPHERMENT = (1 « 3);
int CRSRVLib.CryptoServerLibrary.KEYUSAGE_KEY_AGREEMENT = (1 « 4);
int CRSRVLib.CryptoServerLibrary.KEYUSAGE_KEY_CERT_SIGN = (1 « 5);
int CRSRVLib.CryptoServerLibrary.KEYUSAGE_CRL_SIGN = (1 « 6);
int CRSRVLib.CryptoServerLibrary.KEYUSAGE_ENCIPHER_ONLY = (1 « 7);
int CRSRVLib.CryptoServerLibrary.KEYUSAGE_DECIPHER_ONLY = (1 « 8);
```

Флаги функций вычисления ЭП (класса CRSRVLib.CryptoServerLibrary.sign_param_t)), соответствуют флагам FLAG_CMS_SIGN_XXX :

```
int CRSRVLib.CryptoServerLibrary.FLAG_CMS_SIGN_ADDSIGNER = (1 « 3);
int CRSRVLib.CryptoServerLibrary.FLAG_CMS_SIGN_SUBJECTKEYID = (1 « 16);
int CRSRVLib.CryptoServerLibrary.FLAG_CMS_SIGN_ENVELOPE = (1 « 17);
int CRSRVLib.CryptoServerLibrary.FLAG_CMS_SIGN_CADESBES = (1 « 18);
```

Флаги функций проверки ЭП (класса CRSRVLib.CryptoServerLibrary.verify_param_t), соответствуют флагам FLAG_CMS_VERIFY_XXX :

```
int CRSRVLib.CryptoServerLibrary.FLAG_CMS_VERIFY_DELETESIGNATURES
= (1 « 2);
int CRSRVLib.CryptoServerLibrary.FLAG_CMS_VERIFY_SIGNERKEYUSAGES =
(1 « 3);
int CRSRVLib.CryptoServerLibrary.FLAG_CMS_VERIFY_SIGNERPOLICIES = (1
« 4);
int CRSRVLib.CryptoServerLibrary.FLAG_CMS_VERIFY_
SIGNEREXTKEYUSAGES = (1 « 5);
int CRSRVLib.CryptoServerLibrary.FLAG_CMS_VERIFY_
MINIMUMSIGNATURES = (1 « 7);
int CRSRVLib.CryptoServerLibrary.FLAG_CMS_VERIFY_DONOTCHECKTIMES
= (1 « 10);
int CRSRVLib.CryptoServerLibrary.FLAG_CMS_VERIFY_DONOTADDSIGNER =
(1 « 11);
int CRSRVLib.CryptoServerLibrary.FLAG_CMS_VERIFY_
IGNOREATTACHEDSIGNER = (1 « 12);
int CRSRVLib.CryptoServerLibrary.FLAG_CMS_VERIFY_USEREVOCATIONTIME
```

```

= (1 « 13);
int          CRSRVLib.CryptoServerLibrary.FLAG_CMS_VERIFY_-
DONOTADDATTACHEDSIGNER = (1 « 14);
int          CRSRVLib.CryptoServerLibrary.FLAG_CMS_VERIFY_-
REQUIREATTACHEDSIGNER = (1 « 15);
int  CRSRVLib.CryptoServerLibrary.FLAG_CMS_VERIFY_USEATTACHEDCHAIN
= (1 « 16);
int          CRSRVLib.CryptoServerLibrary.FLAG_CMS_VERIFY_-
REQUIREATTACHEDCHAIN = (1 « 17);

```

Флаги функций выполнения зашифрования (класса CRSRVLib.CryptoServerLibrary.encrypt_param_t), соответствуют флагам FLAG_CMS_ENCRYPT_XXX :

```

int          CRSRVLib.CryptoServerLibrary.FLAG_CMS_ENCRYPT_-
USEREMOTESEARCH = (1 « 2);
int          CRSRVLib.CryptoServerLibrary.FLAG_CMS_ENCRYPT_-
DONOTCHECKTIMES = (1 « 4);
int          CRSRVLib.CryptoServerLibrary.FLAG_CMS_ENCRYPT_-
DONOTVERIFYCHAIN = (1 « 5);
int          CRSRVLib.CryptoServerLibrary.FLAG_CMS_ENCRYPT_-
ADDREMOTETOLOCAL = (1 « 7);
int          CRSRVLib.CryptoServerLibrary.FLAG_CMS_ENCRYPT_-
DONOTCHECKKEYTIME = (1 « 8);
int  CRSRVLib.CryptoServerLibrary.FLAG_CMS_ENCRYPT_SUBJECTISPARTIAL
= (1 « 9);
int          CRSRVLib.CryptoServerLibrary.FLAG_CMS_ENCRYPT_-
IGNORESTORECACHE = (1 « 10);
int          CRSRVLib.CryptoServerLibrary.FLAG_CMS_ENCRYPT_-
IGNORESTORELOCAL = (1 « 11);
int  CRSRVLib.CryptoServerLibrary.FLAG_CMS_ENCRYPT_SUBJECTATTRIBUTE
= (1 « 12);
int  CRSRVLib.CryptoServerLibrary.FLAG_CMS_ENCRYPT_SUBJKEYID  = (1 «
16);
int          CRSRVLib.CryptoServerLibrary.FLAG_CMS_ENCRYPT_GOST_R_34_12_-
15MG = (1 « 17);
int  CRSRVLib.CryptoServerLibrary.FLAG_CMS_ENCRYPT_GOST_R_34_12_15GH
= (1 « 18);
int          CRSRVLib.CryptoServerLibrary.FLAG_CMS_ENCRYPT_-
ADDOMACATTRIBUTE = (1 « 19);
int          CRSRVLib.CryptoServerLibrary.FLAG_CMS_ENCRYPT_-
RECIPKEYAGREEMENT = (1 « 20);

```

Флаги функций поиска сертификатов (класса CRSRVLib.CryptoServerLibrary.find_param_t), соответствуют флагам FLAG_FIND_XXX :

```

int  CRSRVLib.CryptoServerLibrary.FLAG_FIND_CERTWITHPRIVATE = (1 « 0);
int  CRSRVLib.CryptoServerLibrary.FLAG_FIND_USEREMOTESEARCH = (1 « 2);
int  CRSRVLib.CryptoServerLibrary.FLAG_FIND_DONOTCHECKTIMESK = (1 «

```

```

4);
int CRSSRVLib.CryptoServerLibrary.FLAG_FIND_DONOTVERIFYCHAIN = (1 « 5);
int CRSSRVLib.CryptoServerLibrary.FLAG_FIND_ADDREMOTETOLOCAL = (1 «
7);
int CRSSRVLib.CryptoServerLibrary.FLAG_FIND_DONOTCHECKKEYTIME = (1 «
8);
int CRSSRVLib.CryptoServerLibrary.FLAG_FIND_SUBJECTISPARTIAL = (1 « 9);
int CRSSRVLib.CryptoServerLibrary.FLAG_FIND_IGNORESTORECACHE = (1 «
10);
int CRSSRVLib.CryptoServerLibrary.FLAG_FIND_IGNORESTORELOCAL = (1 «
11);
int CRSSRVLib.CryptoServerLibrary.FLAG_FIND_SUBJECTATTRIBUTE = (1 « 12);

```

Флаги функции импорта (класса CRSSRVLib.CryptoServerLibrary.import_param_t), соответствуют флагам FLAG_IMPORT_XXX :

```

int CRSSRVLib.CryptoServerLibrary.FLAG_IMPORT_CERDERPEMFORMAT = (1
« 1);
int CRSSRVLib.CryptoServerLibrary.FLAG_IMPORT_CRLDERPEMFORMAT = (1 «
2);
int CRSSRVLib.CryptoServerLibrary.FLAG_IMPORT_CERTWITHPRIVATE = (1 «
3);
int CRSSRVLib.CryptoServerLibrary.FLAG_IMPORT_SIGNEDCARAUPDATE = (1 «
4);
int CRSSRVLib.CryptoServerLibrary.FLAG_IMPORT_DONOTVERIFYCHAIN = (1 «
5);
int CRSSRVLib.CryptoServerLibrary.FLAG_IMPORT_FORCEREMOTESTORE = (1
« 6);

```

Флаги функции проверки сертификата (класса CRSSRVLib.CryptoServerLibrary.verify_policy_param_t), соответствуют флагам FLAG_POLICY_XXX :

```

int CRSSRVLib.CryptoServerLibrary.FLAG_POLICY_DONOTCHECKTIMES = (1«
2);
int CRSSRVLib.CryptoServerLibrary.FLAG_POLICY_DOCRLVERIFICATION = (1 «
5);
int CRSSRVLib.CryptoServerLibrary.FLAG_POLICY_DONOTCHECKKEYTIME = (1
« 6);

```

Флаги функции простановки штампа времени (класса CRSSRVLib.CryptoServerLibrary.tsp_request_param_t), соответствуют флагам FLAG_TSP_REQUEST_XXX :

```

int CRSSRVLib.CryptoServerLibrary.FLAG_TSP_REQUEST_INCLUDENONCE = (1
« 1);
int CRSSRVLib.CryptoServerLibrary.FLAG_TSP_REQUEST_ATTACHEDSIGNER =
(1 « 2);

```

Флаги функции вычисления ЭП и формирования штампов времени (класса CRSSRVLib.CryptoServerLibrary.tsp_response_param_t), соответ-

ствуют флагам FLAG_TSP_RESPONSE_XXX :

int CRSRVLib.CryptoServerLibrary.FLAG_TSP_RESPONSE_INCLUDETSANAME
= (1 « 1);

Флаги функции проверки штампа времени CMS-сообщений (класса CRSRVLib.CryptoServerLibrary.tsp_verify_param_t), соответствуют флагам FLAG_TSP_VERIFY_XXX :

int CRSRVLib.CryptoServerLibrary.FLAG_TSP_VERIFY_IGNOREATTACHEDSIGNER = (1 « 1);

Флаги CMS-сообщения (класса CRSRVLib.CryptoServerLibrary.cms_msginf_t), соответствуют флагам FLAG_CMS_MSGINF_FLAG_XXX :

int CRSRVLib.CryptoServerLibrary.FLAG_CMS_MSGINF_FLAG_NDEF = (1 « 0);

int CRSRVLib.CryptoServerLibrary.FLAG_CMS_MSGINF_FLAG_DTCH = (1 « 1);

1.4.5 Перечисления

Выбор идентификации сертификата по уникальному идентификатору (член type класса CRSRVLib.CryptoServerLibrary.certid_t):

int CRSRVLib.CryptoServerLibrary.ID_ISSUER_AND_SERIAL = (0) - по издателю и серийному номеру.

int CRSRVLib.CryptoServerLibrary.ID_KEYID = (1) - по идентификатору закрытого ключа.

int CRSRVLib.CryptoServerLibrary.ID_CERTHASH = (2) - по хэшу сертификата.

int CRSRVLib.CryptoServerLibrary.ID_ALIAS = (3) - по символическому имени (алиасу) рабочей сессии КС.

Тип структуры идентификатора сертификатов CMS-сообщения (член type класса CRSRVLib.CryptoServerLibrary.cms_certid_t):

int CRSRVLib.CryptoServerLibrary.FLAG_CMS_CERTID_TYPE_ISSN = (1) - по издателю и серийному номеру.

int CRSRVLib.CryptoServerLibrary.FLAG_CMS_CERTID_TYPE_SKID = (2) - по идентификатору ключа владельца.

Тип CMS-сообщения (член type класса CRSRVLib.CryptoServerLibrary.cms_msginf_t):

int CRSRVLib.CryptoServerLibrary.FLAG_CMS_MSGINF_TYPE_SIGN = (1) - CMS-сообщение является подписанным

int CRSRVLib.CryptoServerLibrary.FLAG_CMS_MSGINF_TYPE_ENVL = (2) - CMS-сообщение является зашифрованным

Тип структуры информации о получателе CMS-сообщения (член type класса CRSRVLib.CryptoServerLibrary.cms_recinf_t): int

CRSRVLib.CryptoServerLibrary.FLAG_CMS_RECINF_TYPE_KTRI = (1) - структура информации о получателе типа KeyTransport.

int CRSRVLib.CryptoServerLibrary.FLAG_CMS_RECINF_TYPE_KARI = (2) - структура информации о получателе типа KeyAgreement.

1.4.6 Общие классы

Класс **CRSRVLib.CryptoServerLibrary.mem_blk_t**

Блок данных.

Состав:

– int **len**

Размер буфера (не может быть более 2 Гбайт).

– byte[] **buf**

Буфер (массив с данными).

1.4.7 Классы контекстов потоковых функций

Класс **CRSRVLib.CryptoServerLibrary.strhash_t**

Контекст выполнения потоковых функций вычисления хэш-значения данных (для создания объекта следует использовать конструктор по умолчанию).

Примечание - Данный класс не поддерживает доступ к своим членам.

Класс **CRSRVLib.CryptoServerLibrary.strcms_handle_t**

Контекст выполнения потоковых функций CMS-сообщений (для создания объекта следует использовать конструктор по умолчанию).

Примечание - Данный класс не поддерживает доступ к своим членам.

1.4.8 Классы, уникально идентифицирующие сертификат

Класс **CRSRVLib.CryptoServerLibrary.issuer_and_serial_t**

Издатель и серийный номер сертификата.

Состав:

– String **issuer**

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

– String **serialNumber**

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

Класс **CRSRVLib.CryptoServerLibrary.certid_t**

Уникальный идентификатор сертификата.

Состав:

– int **type**

Выбор идентификации (ID_ISSUER_AND_SERIAL или ID_KEYID или ID_CERTHASH или ID_ALIAS). Рекомендуется использовать идентификацию ID_ALIAS.

– issuer_and_serial_t **ias**

Издатель и серийный номер сертификата (в случае ID_ISSUER_AND_SERIAL).

– String **keyId**

Идентификатор ключа (в случае ID_KEYID).

– mem_blk_t **certHash**

Хэш сертификата (в случае ID_CERTHASH).

– String **alias**

Символическое имя сессии КС (в случае ID_ALIAS).

1.4.9 Классы, описывающие сертификат

Класс **CRSRVLib.CryptoServerLibrary.altname_t**

Альтернативное имя (издателя или владельца сертификата).

Состав:

– String **emailAddress**

RFC 822 Email адрес.

– String **DNS**

DNS адрес.

– String **URI**

URI адрес.

– String **IP**

IP адрес.

– String **organizationName**

Наименование организации.

– String **registeredAddress**

Зарегистрированный адрес.

– String **surname**

Ф.И.О владельца сертификата.

– String **businessCategory**

Должность.

– String **telephoneNumber**

Номер телефона.

– String **description**

Описание.

– String **account_number**

Номер расчетного счета.

– String **bank_id**

Банковский идентификационный код.

– String **physicalDelivery**

Почтовый адрес.

– String **exchange_address**

Адрес Microsoft Exchange.

– String **notes_address**

Адрес Lotus Notes.

Класс **CRSRVLib.CryptoServerLibrary.othername_t**

Другое имя.

Состав:

– String **oid**

OID, идентифицирующий другое имя.

– String **name**

Текстовые данные другого имени.

Класс **CRSRVLib.CryptoServerLibrary.altname_ex_t**

Массив других имен альтернативного имени (издателя или владельца сертификата).

Состав:

– int **othername_num**

Число других имен.

– othername_t[] **othernames**

Массив других имен.

Класс CRSRVLib.CryptoServerLibrary.basic_constraints_ex_t

Базовые ограничения.

Состав:

– int **ca**

Признак сертификата ЦС.

– int **pathlen**

Максимальная длина цепочки. Значение -1 означает отсутствие ограничения на длину цепочки.

Класс CRSRVLib.CryptoServerLibrary.extkeyusage_t

Расширенное использование ключа сертификата.

Состав:

– String **oid**

OID, идентифицирующий применение ключа сертификата.

Класс CRSRVLib.CryptoServerLibrary.extension_t

Дополнение сертификата.

Состав:

– String **oid**

OID, идентифицирующий дополнение.

– int **type**

Тип дополнения.

– int **critical**

Флаг критичности.

– int **len**

Размер блока данных расширения.

– byte[] **data**

Блок данных расширения в DER-кодировке.

Класс CRSRVLib.CryptoServerLibrary.policy_t

Политика использования сертификата.

Состав:

– String **oid**

OID, идентифицирующий политику.

– String **org_name**

Название организации.

– String **text**

Текст политики использования сертификата.

Класс CRSRVLib.CryptoServerLibrary.certificate_t

Класс сертификата.

Состав:

– int **fields**

Содержит побитовое ИЛИ тех и только тех констант FIELD_XXX, для которых в соответствующих им полях объекта класса заданы значения.

– String **issuer**

X.500 имя издателя (в виде строки DN LDAP \0). При задании FIELD_ISSUER в поле fields.

– String **serialNumber**

Серийный номер сертификата (в виде hex-строки \0). При задании FIELD_SERIALNUMBER в поле fields.

– String **subject**

X.500 имя владельца (в виде строки DN LDAP \0).

– String **algorithm**

Алгоритм открытого ключа. При задании FIELD_ALGORITHM в поле fields.

– int **notBefore**

Время начала действия сертификата. При задании FIELD_NOTBEFORE в поле fields.

– int **notAfter**

Время окончания действия сертификата. При задании FIELD_NOTAFTER в поле fields.

– int **keyUsage**

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

– int **notBeforePrivate**

Время начала действия закрытого ключа. При задании FIELD_NOTBEFOREPRIVATE в поле fields.

– int **notAfterPrivate**

Время окончания действия закрытого ключа. При задании FIELD_NOTAFTERPRIVATE в поле fields.

– altname_t **issuerAltName**

Альтернативное имя издателя. При задании FIELD_ISSUERALTNAME в поле fields.

– altname_t **subjectAltName**

Альтернативное имя владельца. При задании FIELD_SUBJECTALTNAME в поле fields.

– String **keyId**

Идентификатор ключа, соответствующего сертификату. При задании FIELD_KEYID в поле fields.

– int **policy_num**

Число политик использования.

– policy_t[] **policies**

Массив политик использования. При задании FIELD_POLICY в поле fields.

– int **extkeyusage_num**

Число расширенных использований ключа.

– extkeyusage_t[] **extKeyUsage**

Массив расширенных использований ключа. При задании FIELD_EXTKEYUSAGE в поле fields.

– int **extension_num**

Число расширений.

– extension_t[] **extensions**

Массив расширений. При задании FIELD_EXTENSIONS в поле fields.

– mem_blk_t **certEncoded**

Сертификат в DER-кодировке. При задании FIELD_CERTENCODED в поле fields.

– mem_blk_t **certHash**

Хэш сертификата. При задании FIELD_CERTHASH в поле fields.

1.4.10 Классы, описывающие CAC

Класс CRSRVLib.CryptoServerLibrary.revcert_t

Аннулированный сертификат.

Состав:

– String **serialNumber**

Серийный номер аннулированного сертификата.

– int **revtime**

Время аннулирования сертификата.

– int **reason**

Причина аннулирования сертификата - может принимать одно из следующих значений:

- **-1** - причина отсутствует;
- **0** - причина не указана;
- **1** - компрометация ключа;
- **2** - компрометация ключа ЦС;
- **3** - изменена принадлежность;
- **4** - сертификат заменен;
- **5** - действие сертификата остановлено;
- **6** - действие сертификата приостановлено;
- **8** - удаление из CAC;
- **9** - привилегия отозвана;
- **10** - компрометация ключа ЦР.

Класс CRSRVLib.CryptoServerLibrary.crl_t

Класс CAC.

Состав:

– int **fields**

Содержит побитовое ИЛИ тех и только тех констант FIELD_CRL_XXX, для которых в соответствующих им полях объекта класса заданы значения.

- String **issuer**

X.500 имя издателя (в виде строки DN LDAP \0). При задании FIELD_CRL_ISSUER в поле fields.

- int **lastUpdate**

Время начала действия CAC. При задании FIELD_CRL_LASTUPDATE в поле fields.

- int **nextUpdate**

Время окончания действия CAC. При задании FIELD_CRL_NEXTUPDATE в поле fields.

- int **number**

Порядковый номер CAC. При задании FIELD_CRL_NUMBER в поле fields.

- int **revcert_num**

Число аннулированных сертификатов.

- revcert_t[] **revcerts**

Массив аннулированных сертификатов. При задании FIELD_CRL_REVOKED в поле fields.

- mem_blk_t **crlEncoded**

CAC в DER-кодировке. При задании FIELD_CRL_CRLENCODED в поле fields.

- mem_blk_t **crlHash**

Хэш CAC - в настоящее время не поддерживается. При задании FIELD_CRL_CRLHASH в поле fields.

1.4.11 Параметры выполнения функций

Класс CRSRVLib.CryptoServerLibrary.sign_param_t

Параметры вычисления ЭП.

Состав:

- int **flag**

Флаги (могут использоваться FLAG_CMS_SIGN_XXX).

- certid_t **mycert**

Идентификатор сертификата для определения используемой сессии КС (должен уникально идентифицировать рабочий сертификат сессии при использовании Криптографического сервера).

Класс CRSRVLib.CryptoServerLibrary.verify_param_t

Параметры проверки ЭП.

Состав:

- int **flag**

Флаги (могут использоваться FLAG_CMS_VERIFY_XXX).

- certid_t **mycert**

Идентификатор сертификата для определения используемой сессии КС (должен уникально идентифицировать рабочий сертификат сессии при использовании Криптографического сервера).

- int **keyUsage**

Область применения ключа ЭП.

- int **policy_num**

Количество политик использования.

- policy_t[] **policies**

Массив политик использования.

- int **extkeyusage_num**

Количество расширенных использований ключа.

- extkeyusage_t[] **extKeyUsage**

Массив расширенных использований ключа.

- int **nSignToDelete**

Количество ЭП с конца, которые необходимо удалить. Для удаления всех ЭП следует установить равным DELETE_ALL_SIGNS.

- int **minsigns**

Минимальное количество ЭП.

- int **info**

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

Класс CRSRVLib.CryptoServerLibrary.decrypt_param_t

Параметры расшифрования.

Состав:

- int **flag**

Флаги (могут использоваться FLAG_CMS_DECRYPT_XXX).

- certid_t **mycert**

Идентификатор сертификата для определения используемой сессии КС (должен уникально идентифицировать рабочий сертификат сессии при использовании Криптографического сервера).

- int **info**

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

Класс CRSRVLib.CryptoServerLibrary.encrypt_param_t

Параметры зашифрования.

Состав:

- int **flag**

Флаги (могут использоваться FLAG_CMS_ENCRYPT_XXX).

- certid_t **mycert**

Идентификатор сертификата для определения используемой сессии КС (должен уникально идентифицировать рабочий сертификат сессии при использовании Криптографического сервера).

- int **receiver_num**

Количество шаблонов сертификатов получателей.

- certificate_t[] **receivers**

Массив шаблонов для поиска сертификатов получателей.

Класс CRSRVLib.CryptoServerLibrary.find_param_t

Параметры поиска сертификата(ов) по заданному шаблону.

Состав:

– int **flag**

Флаги (могут использоваться FLAG_FIND_XXX).

– certid_t **mycert**

Идентификатор сертификата для определения используемой сессии КС (должен уникально идентифицировать рабочий сертификат сессии при использовании Криптографического сервера). При подключении к КС и задании флага FLAG_FIND_MY использование значения null вернет информацию о всех рабочих сертификатах сессий КС.

– certificate_t **certTemplate**

Шаблон сертификата для поиска.

– int **info**

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

Класс CRSRVLib.CryptoServerLibrary.import_param_t

Параметры импортирования объекта или объектов.

Состав:

– int **flag**

Флаги (могут использоваться FLAG_IMPORT_XXX).

– certid_t **mycert**

Идентификатор сертификата для определения используемой сессии КС (должен уникально идентифицировать рабочий сертификат сессии при использовании Криптографического сервера).

Класс CRSRVLib.CryptoServerLibrary.verify_policy_param_t

Параметры построения цепочки и проверки политики сертификата или САС.

Состав:

– int **size**

Размер объекта класса в байтах (не используется).

– int **flag**

Флаги (могут использоваться FLAG_POLICY_XXX).

– certid_t **mycert**

Идентификатор сертификата для определения используемой сессии КС (должен уникально идентифицировать рабочий сертификат сессии при использовании Криптографического сервера).

– int **check_time**

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

– int **keyusage**

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

– int **eku_num**

Количество расширенных использований ключа.

– extkeyusage_t[] **extkeyusages**

Массив расширенных использований ключа. Сертификат будет проверяться на наличие заданных расширенных использований ключа.

– int **policy_num**

Количество политик использования.

- policy_t[] **policies**

Массив политик использования. Сертификат будет проверяться на наличие заданных политик использования.

Класс CRSRVLib.CryptoServerLibrary.tsp_request_param_t

Параметры простановки штампа времени CMS-сообщений.

Состав:

- int **flag**

Флаги (могут использоваться FLAG_TSP_REQUEST_XXX).

- certid_t **mycert**

Идентификатор сертификата для определения используемой сессии КС (должен уникально идентифицировать рабочий сертификат сессии при использовании Криптографического сервера).

- int **index**

Поле с порядковым номером (индексом) ЭП CMS-сообщения, для которой следует запросить штамп времени.

Класс CRSRVLib.CryptoServerLibrary.tsp_response_param_t

Параметры вычисления ЭП и формирования штампов времени.

Состав:

- int **flag**

Флаги (могут использоваться FLAG_TSP_RESPONSE_XXX).

- certid_t **mycert**

Идентификатор сертификата для определения используемой сессии КС (должен уникально идентифицировать рабочий сертификат сессии при использовании Криптографического сервера).

Класс CRSRVLib.CryptoServerLibrary.tsp_verify_param_t

Параметры простановки штампа времени CMS-сообщений.

Состав:

- int **flag**

Флаги (могут использоваться FLAG_TSP_VERIFY_XXX).

- certid_t **mycert**

Идентификатор сертификата для определения используемой сессии КС (должен уникально идентифицировать рабочий сертификат сессии при использовании Криптографического сервера).

- int **index**

Поле с порядковым номером (индексом) ЭП CMS-сообщения, для которой следует запросить штамп времени.

- int **info**

Поле с требуемой возвращаемой информацией о сертификате сервера штампов времени.

Класс CRSRVLib.CryptoServerLibrary.ocsp_request_param_t

Параметры получения online-статуса сертификата.

Состав:

– int **flag**

Поле с маской флагов (зарезервировано, должно быть равно 0).

– certid_t **mycert**

Идентификатор сертификата для определения используемой сессии КС (должен уникально идентифицировать рабочий сертификат сессии при использовании Криптографического сервера).

Класс CRSRVLib.CryptoServerLibrary.ocsp_response_param_t

Параметры вычисления ЭП online-статуса сертификата.

Состав:

– int **flag**

Поле с маской флагов (зарезервировано, должно быть равно 0).

– certid_t **mycert**

Идентификатор сертификата для определения используемой сессии КС (должен уникально идентифицировать рабочий сертификат сессии при использовании Криптографического сервера).

Класс CRSRVLib.CryptoServerLibrary.ocsp_verify_param_t

Параметры проверки ЭП online-статуса сертификата.

Состав:

– int **flag**

Поле с маской флагов (зарезервировано, должно быть равно 0).

– certid_t **mycert**

Идентификатор сертификата для определения используемой сессии КС (должен уникально идентифицировать рабочий сертификат сессии при использовании Криптографического сервера).

– int **info**

Поле с требуемой возвращаемой информацией о сертификате сервера OSCP ответчика.

1.4.12 Результаты выполнения функций

Класс CRSRVLib.CryptoServerLibrary.sign_status_t

Результат проверки отдельной ЭП.

Состав:

– int **status**

Результат проверки ЭП (VCERT_OK - в случае успеха проверки, иначе код ошибки).

– int **time**

Время вычисления ЭП в часовом поясе UTC (если присутствует в подписанном сообщении, иначе равно "0").

– certificate_t **cert**

Указатель на сертификат, на котором была выполнена ЭП (если сертификат не найден, то значение равно NULL).

Класс CRSRVLib.CryptoServerLibrary.verify_result_t

Результат проверки ЭП подписанного сообщения.

Состав:

– int **sign_num**

Количество ЭП в подписанном сообщении.

– sign_status_t[] **signs**

Массив результатов проверки отдельных ЭП сообщения.

Класс CRSRVLib.CryptoServerLibrary.decrypt_result_t

Результат расшифрования зашифрованного сообщения.

Состав:

– certificate_t **sender**

Сертификат отправителя зашифрованного сообщения (для систем с неанонимным шифрованием).

Класс CRSRVLib.CryptoServerLibrary.find_result_t

Результат поиска сертификата(ов) по заданному шаблону.

Состав:

– int **num**

Количество найденных сертификатов.

– certificate_t[] **certs**

Массив найденных сертификатов.

Класс CRSRVLib.CryptoServerLibrary.tsp_verify_result_t

Результат проверки штампа времени для заданной ЭП.

Состав:

– int **time**

Поле со временем простановки (вычисления ЭП) штампа времени в часовом поясе UTC.

– certificate_t **mycert**

Сертификат сервера штампов времени.

Класс CRSRVLib.CryptoServerLibrary.ocsp_verify_result_t

Результат проверки ЭП online-статуса сертификата.

Состав:

– int **status**

Поле с online-статусом сертификата (соответствует полю status структуры ocsp_verify_result_t).

– int **reason**

Для аннулированного сертификата, поле с причиной аннулирования сертификата.

– int **revtime**

Для аннулированного сертификата, поле со временем аннулирования сертификата.

– int **thisupd**

Поле со временем начала действия данного online-статуса сертификата.

– int **nextupd**

Поле со временем окончания действия данного online-статуса сертификата.

– certificate_t **cert**

Сертификат сервера OSCP ответчика.

Класс CRSRVLib.CryptoServerLibrary.cms_certid_t

Класс идентификатора сертификата CMS-сообщения.

Состав:

– int **type**

Поле с типом структуры идентификатора сертификатов CMS-сообщения. Принимает значения FLAG_CMS_CERTID_TYPE_XXX

– String **issuer**

Поле с X.500-именем издателя сертификата в виде строки.

– String **serialNum**

Поле с серийным номером сертификата в виде строки.

– String **certHash**

Поле с хэш-значением пары Имя издателя/Серийный номер сертификата в виде строки.

– String **subjKeyId**

Поле с данными дополнения "Идентификатор ключа владельца" в виде строки.

Класс CRSRVLib.CryptoServerLibrary.cms_siginf_t

Класс информации о подписанте CMS-сообщения.

Состав:

– uint32_t **version**

Поле с номером версии структуры информации о подписанте CMS-сообщения.

– string_t **digestAlg**

Поле с строкой объектного идентификатора (OID) алгоритма хэширования.

– string_t **signatureAlg**

Поле с строкой объектного идентификатора (OID) алгоритма ЭП.

– cms_certid_t **signerId**

Поле со структурой идентификатора сертификата подписанта CMS-сообщения.

– date_t **signingTime**

Поле со временем вычисления ЭП.

– uint32_t **authAttr_num**

Количество элементов массива строк объектных идентификаторов аутентифицированных атрибутов, находящихся в ЭП.

– policy_t * **authAttrs**

Поле с массивом строк объектных идентификаторов аутентифицированных атрибутов, находящихся в ЭП.

– uint32_t **unauthAttr_num**

Количество элементов массива строк объектных идентификаторов неаутентифицированных атрибутов, находящихся в ЭП.

– policy_t * **unauthAttrs**

Поле с массивом строк объектных идентификаторов неаутентифицирован-

ных атрибутов, находящихся в ЭП.

Класс CRSRVLib.CryptoServerLibrary.cms_recinf_t

Класс информации о получателе CMS-сообщения.

Состав:

– int **type**

Поле с типом структуры информации о получателе CMS-сообщения.

– int **index**

Поле с номером (индексом) структуры информации о получателе CMS-сообщения.

– int **version**

Поле с номером версии структуры информации о получателе CMS-сообщения.

– cms_certid_t **originatorId**

Поле со структурой идентификатора сертификата отправителя CMS-сообщения.

– String **publicKeyAlg**

Поле с строкой объектного идентификатора (OID) алгоритма открытого ключа сертификата получателя.

– String **cipherAlg**

Поле с строкой объектного идентификатора (OID) алгоритма согласования или зашифрования сеансового ключа.

– cms_certid_t **recipientId**

Поле со структурой идентификатора сертификата получателя CMS-сообщения.

Класс CRSRVLib.CryptoServerLibrary.cms_msginf_t

Класс информации о CMS-сообщении.

– int **type**

Поле с типом CMS-сообщения:

– int **flags**

Поле с маской (побитовым ИЛИ) флагов CMS-сообщения:

– int **signer_num**

Количество элементов массива структур с информацией о подписантах CMS-сообщения.

– cms_siginf_t **signers**

Поле с массивом структур с информацией о подписантах CMS-сообщения.

– int **recipient_num**

Количество элементов массива структур с информацией о получателях CMS-сообщения.

– cms_recinf_t **recipients**

Поле с массивом структур с информацией о получателях CMS-сообщения.

– String **cipherAlg**

Поле с строкой объектного идентификатора (OID) алгоритма шифрования данных.

– int **unprotAttr_num**

Количество элементов массива строк объектных идентификаторов незащищенных атрибутов.

– String **unprotAttrs**

Поле с массивом строк объектных идентификаторов незащищенных атрибутов.

1.5 Описание функций

1.5.1 Функции инициализации и деинициализации

void **CRSRVLib.CryptoServerLibrary.Detach** ()

Функция отсоединения от КС и уничтожения контекста

Примечание - Данную функцию всегда необходимо вызывать по окончании использования объекта основного класса библиотеки (контекста) для очистки выделенных ресурсов. После вызова данной функции повторное использование уничтоженного контекста не разрешается.

int **CRSRVLib.CryptoServerLibrary.VCERTR_Authorize** (certid_t mycert, int func, mem_blk_t auth)

Функция авторизации выполнения функций КС с использованием пароля

Соответствует функции VCERT_Authorize.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

– **mycert** (in) идентификатор сертификата для определения используемой сессии КС;

– **func** (in) тип функции, выполнение которого необходимо аутентифицировать;

– **auth** (in) аутентификационные данные (пароль).

1.5.2 Функции получения описаний ошибок

String **CRSRVLib.CryptoServerLibrary.VCERTR_GetErrorText** (int error, String message, int len)

Функция получения текстового сообщения по коду ошибки

Возвращаемые значения:

строку с текстовым описанием ошибки.

Аргументы:

– **error** (in) код ошибки;

– **str** (out) строка для текстового описания ошибки (не используется);

– **len** (in) размер строки для текстового описания ошибки (не используется).

1.5.3 Функции экспорта и импорта объектов СУС

int **CRSRVLib.CryptoServerLibrary.VCERTR_Import** (import_param_t pImportPara, mem_blk_t in, mem_blk_t auth)

Функция добавления объекта/объектов из блока данных

Соответствует функции VCERT_ImportMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pImportPara** (in) параметры выполнения импорта;
- **in** (in) импортируемый блок данных;
- **auth** (in) аутентификационные данные.

1.5.4 Функции разбора и получения информации об объектах СУС

int **CRSRVLib.CryptoServerLibrary.VCERTR_ParseCert** (mem_blk_t enccert, int info, certificate_t cert, mem_blk_t auth)

Функция разбора и получения информации о сертификате

Соответствует функции VCERT_ParseCert.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **enccert** (in) сертификат в DER-кодировке;
- **info** (in) требуемая информация о сертификате (константы FIELD_XXX);
- **cert** (out) объект класса разобранного сертификата;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_ParseCrl** (mem_blk_t enccrl, int info, crl_t crl, mem_blk_t auth)

Функция разбора и получения информации о САС

Соответствует функции VCERT_ParseCrl.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **enccrl** (in) САС в DER-кодировке;
- **info** (in) требуемая информация о САС (константы FIELD_CRL_XXX);
- **crl** (out) объект класса разобранного САС;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_ParseAltnameEx** (mem_blk_t encaltname, altname_ex_t altname, mem_blk_t auth)

Функция разбора и получения массива других имен альтернативного имени

Соответствует функции VCERT_ParseAltnameEx.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **encaltname** (in) альтернативное имя в DER-кодировке;
- **altname** (out) объект класс разобранного альтернативного имени;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_ParseBasicConstraintsEx**

(mem_blk_t encbc, basic_constraints_ex_t bc, mem_blk_t auth)

Функция разбора и получения информации о базовых ограничениях сертификата

Соответствует функции VCERT_ParseBasicConstraintsEx.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **encbc** (in) базовые ограничения в DER-кодировке;
- **bc** (out) объект класс разобранных базовых ограничений;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_ParseKeyIdentifierEx** (mem_blk_t ikid, mem_blk_t okid, mem_blk_t auth)

Функция разбора и получения информации об идентификаторе ключа издателя

Соответствует функции VCERT_ParseKeyIdentifierEx.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **ikid** (in) блок памяти с данными дополнения идентификатора ключа издателя в DER-кодировке;
- **okid** (out) блок памяти с разобранным идентификатором ключа издателя;
- **auth** (in) аутентификационные данные.

1.5.5 Функции построения и проверки цепочек объектов СУС

int **CRSRVLib.CryptoServerLibrary.VCERTR_VerifyCert** (verify_param_t pVerifyPara, mem_blk_t enccert, certificate_t cert, mem_blk_t auth)

Функция построения цепочки и проверки действительности сертификата

Соответствует функции VCERT_VerifyCert.

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки сертификата;
- **enccert** (in) проверяемый сертификат в DER-кодировке;
- **cert** (out) раскодированный и разобранный сертификат;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_VerifyPolicy** (verify_policy_param_t pPolicyPara, mem_blk_t der, mem_blk_t auth)

Функция построения цепочки и проверки политики сертификата или САС

Соответствует функциям VCERT_VerifyCertificatePolicy и VCERT_VerifyCrlPolicy.

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pPolicyPara** (in) параметры проверки сертификата или САС;
- **der** (in) проверяемый сертификат или список аннулированных сертификатов (САС) в DER-кодировке;

- **auth** (in) аутентификационные данные.

1.5.6 Функции вычисления хэш-значений

int **CRSRVLib.CryptoServerLibrary.VCERTR_BlkJHash** (String algorithm, mem_blk_t data, mem_blk_t hash, mem_blk_t auth)

Функция хэширования блока данных

Соответствует функции VCERT_BlkJHashMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **algorithm** (in) строка объектного идентификатора (OID) алгоритма хэширования;
- **data** (in) хэшируемый блок данных;
- **hash** (out) буфер для хэш - функции блока данных;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_StrHashInit** (String algorithm, strhash_handle_t hStr, mem_blk_t auth)

Функция инициализации потокового хэширования блока данных

Соответствует функции VCERT_StrHashInitMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки (в этом случае контекст выполнения потоковой операции не создается).

Аргументы:

- **algorithm** (in) объектный идентификатор (OID) алгоритма хэширования;
- **hStr** (out) контекст выполнения потоковой операции;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_StrHashUpdate** (strhash_handle_t hStr, mem_blk_t data)

Функция продолжения потокового хэширования блока данных

Соответствует функции VCERT_StrHashUpdateMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **hStr** (in) контекст выполнения потоковой операции;
- **data** (in) указатель на блок данных.

int **CRSRVLib.CryptoServerLibrary.VCERTR_StrHashFinal** (strhash_handle_t hStr, mem_blk_t hash)

Функция финализации потокового хэширования блока данных

Соответствует функции VCERT_StrHashFinalMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **hStr** (in) контекст выполнения потоковой операции;
- **hash** (out) буфер для хэш - функции блока данных.

1.5.7 Функции вычисления и проверки ЭП хэш-значений

int **CRSRVLib.CryptoServerLibrary.VCERTR_SignHash** (certid_t mycert, mem_blk_t hash, mem_blk_t sign, mem_blk_t auth)

Функция вычисления ЭП хэш-функции данных

Соответствует функции VCERT_SignHashMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

– **mycert** (in) идентификатор сертификата для определения используемой сессии КС (должен уникально идентифицировать рабочий сертификат сессии при использовании Криптографического сервера);

– **hash** (in) блок с хэш-функцией данных;

– **sign_out** (out) вычисленная ЭП хэш-функции данных;

– **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_VerifyHash** (certid_t mycert, mem_blk_t sender, mem_blk_t hash, mem_blk_t sign, mem_blk_t auth)

Функция проверки ЭП хэш-функции данных

Соответствует функции VCERT_VerifyHashMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

– **mycert** (in) идентификатор сертификата для определения используемой сессии КС (должен уникально идентифицировать рабочий сертификат сессии при использовании Криптографического сервера);

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

– **hash** (in) блок с хэш-функцией данных;

– **sign** (in) ЭП хэш-функции данных;

– **auth** (in) аутентификационные данные.

1.5.8 Функции поиска сертификатов

int **CRSRVLib.CryptoServerLibrary.VCERTR_FindCert** (find_param_t pFindPara, find_result_t pFindResult, mem_blk_t auth)

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

Соответствует функции VCERT_FindCert.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

– **pFindPara** (in) параметры поиска и шаблон сертификата;

– **pFindResult** (out) результат поиска - массив найденных сертификатов;

– **auth** (in) аутентификационные данные.

1.5.9 Функции блочного вычисления совмещенной ЭП CMS-сообщений

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsBlkAttSign** (sign_param_t pSignPara, mem_blk_t data, mem_blk_t ocms, mem_blk_t auth)

Функция блочного вычисления совмещенной ЭП блока памяти

Соответствует функции VCERT_CmsBlkAttSignMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pSignPara** (in) параметры вычисления ЭП CMS-сообщений;
- **data** (in) блок памяти с данными для вычисления ЭП;
- **ocms** (out) блок памяти с подписанным CMS-сообщением;
- **auth** (in) аутентификационные данные.

1.5.10 Функции потокового вычисления совмещенной ЭП CMS-сообщений

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrAttSignInit** (sign_param_t pSignPara, mem_blk_t data, strcms_handle_t hStr, mem_blk_t auth)

Функция инициализации потокового вычисления совмещенной ЭП блока памяти

Соответствует функции VCERT_CmsStrAttSignInitMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pSignPara** (in) параметры вычисления ЭП CMS-сообщений;
- **data** (in) блок памяти с началом данных для вычисления ЭП;
- **hStr** (out) контекст потоковой операции;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrAttSignUpdate** (sign_param_t pSignPara, strcms_handle_t hStr, mem_blk_t data, mem_blk_t ocms)

Функция продолжения потокового вычисления совмещенной ЭП блока памяти

Соответствует функции VCERT_CmsStrAttSignUpdateMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pSignPara** (in) параметры вычисления ЭП CMS-сообщений;
- **hStr** (in) контекст потоковой операции;
- **data** (in) блок памяти с продолжением данных для вычисления ЭП;
- **ocms** (out) блок памяти с продолжением подписанного CMS-сообщения.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrAttSignFinal** (sign_param_t pSignPara, strcms_handle_t hStr, mem_blk_t ocms)

Функция финализации потокового вычисления совмещенной ЭП блока памяти

Соответствует функции VCERT_CmsStrAttSignFinalMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pSignPara** (in) параметры вычисления ЭП CMS-сообщений;
- **hStr** (in) контекст потоковой операции;
- **ocms** (out) блок памяти с окончанием подписанного CMS-сообщения.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrAttSignFile** (sign_param_t pSignPara, String in, String out, mem_blk_t auth)

Функция потокового вычисления совмещенной ЭП файла

Соответствует функции VCERT_CmsStrAttSignFile.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pSignPara** (in) параметры вычисления ЭП CMS-сообщений;
- **in** (in) файл (не нулевой длины) с данными для вычисления ЭП;
- **out** (out) файл с подписанным CMS-сообщением;
- **auth** (in) аутентификационные данные.

1.5.11 Функции блочного вычисления отдельной ЭП CMS-сообщений

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsBlkDetSign** (sign_param_t pSignPara, mem_blk_t data, mem_blk_t icms, mem_blk_t ocms, mem_blk_t auth)

Функция блочного вычисления отдельной ЭП блока памяти

Соответствует функции VCERT_CmsBlkDetSignMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pSignPara** (in) параметры вычисления ЭП CMS-сообщений;
- **data** (in) блок памяти с данными для вычисления ЭП;
- **icms** (in) блок памяти (опциональный) с подписанным CMS-сообщением для добавления ЭП;
- **ocms** (out) блок памяти с подписанным CMS-сообщением;
- **auth** (in) аутентификационные данные.

1.5.12 Функции потокового вычисления отдельной ЭП CMS-сообщений

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrDetSignInit** (sign_param_t pSignPara, mem_blk_t icms, strcms_handle_t hStr, mem_blk_t auth)

Функция инициализации потокового вычисления отдельной ЭП блока памяти

Соответствует функции VCERT_CmsStrDetSignInitMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pSignPara** (in) параметры вычисления ЭП CMS-сообщений;

- **icms** (in) блок памяти (опциональный) с подписанным CMS-сообщением для добавления ЭП;
- **hStr** (out) контекст потоковой операции;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrDetSignUpdate** (sign_param_t pSignPara, strcms_handle_t hStr, mem_blk_t data)

Функция продолжения потокового вычисления отдельной ЭП блока памяти
Соответствует функции VCERT_CmsStrDetSignUpdateMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pSignPara** (in) параметры вычисления ЭП CMS-сообщений;
- **hStr** (in) контекст потоковой операции;
- **data** (in) блок памяти с продолжением данных для вычисления ЭП.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrDetSignFinal** (sign_param_t pSignPara, strcms_handle_t hStr, mem_blk_t ocms)

Функция финализации потокового вычисления отдельной ЭП блока памяти
Соответствует функции VCERT_CmsStrDetSignFinalMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pSignPara** (in) параметры вычисления ЭП CMS-сообщений;
- **hStr** (in) контекст потоковой операции;
- **ocms** (out) блок памяти с подписанным CMS-сообщением.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrDetSignFile** (sign_param_t signParam, String data, String sign_in, String sign_out, mem_blk_t auth)

Функция потокового вычисления отдельной ЭП файла
Соответствует функции VCERT_CmsStrDetSignFile.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **signParam** (in/out) параметры вычисления ЭП CMS-сообщений;
- **data** (in) файл (не нулевой длины) с данными для вычисления ЭП;
- **sign_in** (in) файл (опциональный) с подписанным CMS-сообщением для добавления ЭП;
- **sign_out** (out) файл с подписанным CMS-сообщением;
- **auth** (in) аутентификационные данные.

1.5.13 Функции блочной проверки совмещенных ЭП CMS-сообщений

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsBlkAttVerify** (verify_param_t pVerifyPara, mem_blk_t icms, mem_blk_t data, verify_result_t pVerifyResult, mem_blk_t auth)

Функция блочной проверки совмещенных ЭП блока памяти

Соответствует функции VCERT_CmsBlkAttVerifyMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки ЭП CMS-сообщений;
- **icms** (in) блок памяти с подписанным CMS-сообщением;
- **data** (out) блок памяти с выходными данными;
- **pVerifyResult** (out) результат проверки ЭП CMS-сообщения;
- **auth** (in) аутентификационные данные.

1.5.14 Функции потоковой проверки совмещенных ЭП CMS-сообщений

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrAttVerifyInit** (verify_param_t pVerifyPara, mem_blk_t icms, strcms_handle_t hStr, mem_blk_t auth)

Функция инициализации потоковой проверки совмещенных ЭП блока памяти

Соответствует функции VCERT_CmsStrAttVerifyInitMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки ЭП CMS-сообщений;
- **icms** (in) блок памяти с началом подписанного CMS-сообщения;
- **hStr** (out) контекст потоковой операции;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrAttVerifyUpdate** (verify_param_t pVerifyPara, strcms_handle_t hStr, mem_blk_t icms, mem_blk_t data)

Функция продолжения потоковой проверки совмещенных ЭП блока памяти

Соответствует функции VCERT_CmsStrAttVerifyUpdateMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки ЭП CMS-сообщений;
- **hStr** (in) контекст потоковой операции;
- **icms** (in) блок памяти с продолжением подписанного CMS-сообщения;
- **data** (out) блок памяти с продолжением выходных данных.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrAttVerifyFinal** (verify_param_t pVerifyPara, strcms_handle_t hStr, mem_blk_t data, verify_result_t pVerifyResult)

Функция финализации потоковой проверки совмещенных ЭП блока памяти

Соответствует функции VCERT_CmsStrAttVerifyFinalMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки ЭП CMS-сообщений;

- **hStr** (in) контекст потоковой операции;
- **data** (out) блок памяти с окончанием выходных данных;
- **pVerifyResult** (out) результат проверки ЭП CMS-сообщения.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrAttVerifyFile** (verify_param_t verifyParam, String data, String sign_out, verify_result_t verifyResult, mem_blk_t auth)

Функция потоковой проверки совмещенных ЭП файла

Соответствует функции VCERT_CmsStrAttVerifyFile.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **verifyParam** (in) параметры проверки ЭП CMS-сообщений;
- **data** (in) файл (не нулевой длины) с подписанным CMS-сообщением;
- **sign_out** (out) файл с выходными данными;
- **verifyResult** (out) результат проверки ЭП CMS-сообщения;
- **auth** (in) аутентификационные данные.

1.5.15 Функции блочной проверки отделенных ЭП CMS-сообщений

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsBlkDetVerify** (verify_param_t pVerifyPara, mem_blk_t data, mem_blk_t icms, mem_blk_t ocms, verify_result_t pVerifyResult, mem_blk_t auth)

Функция блочной проверки отделенных ЭП блока памяти

Соответствует функции VCERT_CmsBlkDetVerify.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки ЭП CMS-сообщений;
- **data** (in) блок памяти с данными для проверки ЭП;
- **icms** (in) блок памяти с подписанным CMS-сообщением;
- **ocms** (out) блок памяти с выходным CMS-сообщением;
- **pVerifyResult** (out) результат проверки ЭП CMS-сообщения;
- **auth** (in) аутентификационные данные.

1.5.16 Функции потоковой проверки отделенных ЭП CMS-сообщений

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrDetVerifyInit** (verify_param_t pVerifyPara, mem_blk_t icms, strcms_handle_t hStr, mem_blk_t auth)

Функция инициализации потоковой проверки отделенных ЭП блока памяти

Соответствует функции VCERT_CmsStrDetVerifyInitMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки ЭП CMS-сообщений;
- **icms** (in) блок памяти с подписанным CMS-сообщением;

- **hStr** (out) контекст потоковой операции;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrDetVerifyUpdate** (verify_param_t pVerifyPara, strcms_handle_t hStr, mem_blk_t data)

Функция продолжения потоковой проверки отделенных ЭП блока памяти

Соответствует функции VCERT_CmsStrDetVerifyUpdateMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки ЭП CMS-сообщений;
- **hStr** (in) контекст потоковой операции;
- **data** (in) блок памяти с продолжением данных для проверки ЭП;

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrDetVerifyFinal** (verify_param_t pVerifyPara, strcms_handle_t hStr, mem_blk_t ocms, verify_result_t pVerifyResult)

Функция финализации потоковой проверки отделенных ЭП

Соответствует функции VCERT_CmsStrDetVerifyFinalMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки ЭП CMS-сообщений;
- **hStr** (in) контекст потоковой операции;
- **ocms** (out) блок памяти с выходным CMS-сообщением;
- **pVerifyResult** (out) результат проверки ЭП CMS-сообщения;

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrDetVerifyFile** (verify_param_t verifyParam, String data, String sign_in, String sign_out, verify_result_t verifyResult, mem_blk_t auth)

Функция потоковой проверки отделенных ЭП файла

Соответствует функции VCERT_CmsStrDetVerifyFile.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **verifyParam** (in) параметры проверки ЭП CMS-сообщений;
- **data** (in) файл (не нулевой длины) с данными для проверки ЭП;
- **sign_in** (in) файл (не нулевой длины) с подписанным CMS-сообщением;
- **sign_out** (out) файл с выходным CMS-сообщением;
- **verifyResult** (out) результат проверки ЭП CMS-сообщения;
- **auth** (in) аутентификационные данные.

1.5.17 Функции блочного зашифрования CMS-сообщений

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsBlkEncrypt** (encrypt_param_t pEncryptPara, mem_blk_t data, mem_blk_t ocms, mem_blk_t auth)

Функция блочного зашифрования блока памяти

Соответствует функции VCERT_CmsBlkEncryptMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pEncryptPara** (in) параметры зашифрования CMS-сообщений;
- **data** (in) блок памяти с данными для зашифрования;
- **ocms** (out) блок памяти с зашифрованным CMS-сообщением;
- **auth** (in) аутентификационные данные.

1.5.18 Функции потокового зашифрования CMS-сообщений

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrEncryptInit** (encrypt_param_t pEncryptPara, mem_blk_t data, strcms_handle_t hStr, mem_blk_t auth)

Функция инициализации потокового зашифрования блока памяти

Соответствует функции VCERT_CmsStrEncryptInit.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pEncryptPara** (in) параметры зашифрования CMS-сообщений;
- **data** (in) блок памяти с началом данных для зашифрования;
- **hStr** (out) контекст потоковой операции;;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrEncryptUpdate** (encrypt_param_t pEncryptPara, strcms_handle_t hStr, mem_blk_t data, mem_blk_t ocms)

Функция продолжения потокового зашифрования блока памяти

Соответствует функции VCERT_CmsStrEncryptUpdate.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pEncryptPara** (in) параметры зашифрования CMS-сообщений;
- **hStr** (in) контекст потоковой операции;;
- **data** (in) блок памяти с продолжением данных для зашифрования;
- **ocms** (out) блок памяти с продолжением зашифрованного CMS-сообщения;

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrEncryptFinal** (encrypt_param_t pEncryptPara, strcms_handle_t hStr, mem_blk_t ocms)

Функция финализации потокового зашифрования блока памяти

Соответствует функции VCERT_CmsStrEncryptFinal.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pEncryptPara** (in) параметры зашифрования CMS-сообщений ;
- **hStr** (in) контекст потоковой операции;;
- **ocms** (out) блок памяти с окончанием зашифрованного CMS-сообщения;

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrEncryptFile** (encrypt_param_t encryptParam, String in, String out, mem_blk_t auth)

Функция потокового зашифрования файла

Соответствует функции VCERT_CmsStrEncryptFile.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **encryptParam** (in) параметры зашифрования CMS-сообщений ;
- **in** (in) файл (не нулевой длины) с данными для зашифрования;
- **out** (out) файл с зашифрованным CMS-сообщением;
- **auth** (in) аутентификационные данные.

1.5.19 Функции блочного расшифрования CMS-сообщений

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsBlkDecrypt** (decrypt_param_t pDecryptPara, mem_blk_t icms, mem_blk_t data, decrypt_result_t pDecryptResult, mem_blk_t auth)

Функция блочного расшифрования блока памяти

Соответствует функции VCERT_CmsBlkDecryptMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pDecryptPara** (in) параметры расшифрования CMS-сообщений;
- **icms** (in) блок памяти с зашифрованным CMS-сообщением;
- **data** (out) блок памяти с расшифрованными данными;
- **pDecryptResult** (out) результат расшифрования CMS-сообщения;
- **auth** (in) аутентификационные данные.

1.5.20 Функции потокового расшифрования CMS-сообщений

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrDecryptInit** (decrypt_param_t pDecryptPara, mem_blk_t icms, strcms_handle_t hStr, mem_blk_t auth)

Функция инициализации потокового расшифрования блока памяти

Соответствует функции VCERT_CmsStrDecryptInitMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pDecryptPara** (in) параметры расшифрования CMS-сообщений;
- **icms** (in) блок памяти с началом зашифрованного CMS-сообщения;
- **hStr** (out) контекст потоковой операции;;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrDecryptUpdate** (decrypt_param_t pDecryptPara, strcms_handle_t hStr, mem_blk_t icms, mem_blk_t data)

Функция продолжения потокового расшифрования блока памяти

Соответствует функции VCERT_CmsStrDecryptUpdateMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pDecryptPara** (in) параметры расшифрования CMS-сообщений;
- **hStr** (in) контекст потоковой операции;;
- **icms** (in) блок памяти с продолжением зашифрованного CMS-сообщения;
- **data** (out) блок памяти с продолжением расшифрованных данных.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrDecryptFinal** (decrypt_param_t pDecryptPara, strcms_handle_t hStr, mem_blk_t data, decrypt_result_t pDecryptResult)

Функция финализации потокового расшифрования блока памяти

Соответствует функции VCERT_CmsStrDecryptFinalMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pDecryptPara** (in) параметры расшифрования CMS-сообщений;
- **hStr** (in) контекст потоковой операции;;
- **data** (out) блок памяти с окончанием расшифрованных данных;
- **pDecryptResult** (out) результат расшифрования CMS-сообщения.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrDecryptFile** (decrypt_param_t decryptParam, String in, String out, decrypt_result_t decryptResult, mem_blk_t auth)

Функция потокового расшифрования файла

Соответствует функции VCERT_CmsStrDecryptFile.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **decryptParam** (in) параметры расшифрования CMS-сообщений;
- **in** (in) файл (не нулевой длины) с зашифрованным CMS-сообщением;
- **out** (out) файл с расшифрованными данными;
- **decryptResult** (out) результат расшифрования CMS-сообщения;
- **auth** (in) аутентификационные данные.

1.5.21 Функции блочного преобразования отделенных ЭП в совмещенные

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsBlkSignAttach** (mem_blk_t data, mem_blk_t icms, mem_blk_t ocms, int flag, mem_blk_t auth)

Функция блочного преобразования отделенных ЭП блока памяти в совмещенные

Соответствует функции VCERT_CmsBlkSignAttachMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **data** (in) блок памяти с подписанными данными;

- **icms** (in) блок памяти с подписанным CMS-сообщением с отделенными ЭП;
- **ocms** (out) блок памяти с подписанным CMS-сообщением с совмещенными ЭП;
- **flag** (in) маска флагов (зарезервировано, должно быть равно 0);
- **auth** (in) аутентификационные данные.

1.5.22 Функции блочного преобразования совмещенных ЭП в отделенные

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsBlkSignDetach** (mem_ - blk_t icms, mem_blk_t data, mem_blk_t ocms, int flag, mem_blk_t auth)

Функция блочного преобразования совмещенных ЭП блока памяти в отделенные

Соответствует функции VCERT_CmsBlkSignDetachMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **icms** (int) блок памяти с подписанным CMS-сообщением с совмещенными ЭП ;
- **data** (out) блок памяти с подписанными данными;
- **ocms** (out) блок памяти с подписанным CMS-сообщением с отделенными ЭП;
- **flag** (in) маска флагов (зарезервировано, должно быть равно 0);
- **auth** (in) аутентификационные данные.

1.5.23 Функции потокового преобразования совмещенных ЭП в отделенные

nt **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrSignDetachInit** (mem_ - blk_t icms, strcms_handle_t hStr, int flag, mem_blk_t auth)

Функция инициализации потокового преобраз. совмещенных ЭП блока памяти

Соответствует функции VCERT_CmsStrSignDetachInitMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **icms** (in) блок памяти с началом подписанного CMS-сообщения с совмещенными ЭП;
- **hStr** (out) контекст потоковой операции;
- **flag** (in) маска флагов (зарезервировано, должно быть равно 0);
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrSignDetachUpdate** (strcms_handle_t hStr, mem_blk_t icms, mem_blk_t data)

Функция продолжения потокового преобраз. совмещенных ЭП блока памяти

Соответствует функции VCERT_CmsStrSignDetachUpdateMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **hStr** (in) контекст потоковой операции;
- **icms** (in) блок памяти с продолжением подписанного CMS-сообщения с совмещенными ЭП;
- **data** (out) блок памяти с продолжением подписанных данных;

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrSignDetachFinal** (strcms_handle_t hStr, mem_blk_t data, mem_blk_t ocms)

Функция финализации потокового преобраз. совмещенных ЭП блока памяти

Соответствует функции VCERT_CmsStrSignDetachFinalMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **hStr** (in) контекст потоковой операции;
- **data** (int) блок памяти с окончанием подписанных данных;
- **ocms** (out) блок памяти с подписанным CMS-сообщением с отделенными ЭП;

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrSignDetachFile** (String in, String data_out, String out, int flag, mem_blk_t auth)

Функция потокового преобразования совмещенных ЭП файла в отделенные

Соответствует функции VCERT_CmsStrSignDetachFile.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **in** (in) файл (не нулевой длины) с подписанным CMS-сообщением с совмещенными ЭП;
- **data_out** (out) файл с подписанными данными;
- **out** (out) файл с подписанным CMS-сообщением с отделенными ЭП;
- **flag** (in) маска флагов (зарезервировано, должно быть равно 0);
- **auth** (in) аутентификационные данные.

1.5.24 Функции блочного получения информации о CMS-сообщениях

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsBlkMsgInf** (mem_blk_t cms, cms_msginf_t pMsgInf, int flag, mem_blk_t auth)

Функция блочного получения информации о CMS-сообщениях в виде блока памяти

Соответствует функции VCERT_CmsBlkMsgInfMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **cms** (in) блок памяти с CMS-сообщением;
- **pMsgInf** (out) информация о CMS-сообщении;
- **flag** (in) маска флагов (зарезервировано, должно быть равно 0);
- **auth** (in) аутентификационные данные.

1.5.25 Функции потокового получения информации о CMS-сообщениях

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrMsgInfInit** (mem_blk_t cms, strcms_handle_t hStr, int flag, mem_blk_t auth)

Функция инициализации потокового получения информации о блоке памяти CMS

Соответствует функции VCERT_CmsStrMsgInfInitMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **cms** (in) блок памяти с началом CMS-сообщения;
- **hStr** (out) контекст потоковой операции;
- **flag** (in) маска флагов (зарезервировано, должно быть равно **0**);
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrMsgInfUpdate** (strcms_handle_t hStr, mem_blk_t cms)

Функция продолжения потокового получения информации о блоке памяти CMS

Соответствует функции VCERT_CmsStrMsgInfUpdateMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **hStr** (in) контекст потоковой операции;
- **cms** (in) блок памяти с продолжением CMS-сообщения;

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrMsgInfFinal** (strcms_handle_t hStr, cms_msginf_t pMsgInf)

Функция финализации потокового получения информации о блоке памяти CMS

Соответствует функции VCERT_CmsStrMsgInfFinalMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **hStr** (in) контекст потоковой операции;
- **pMsgInf** (out) информация о CMS-сообщении;

int **CRSRVLib.CryptoServerLibrary.VCERTR_CmsStrMsgInfFile** (String data, cms_msginf_t info, int flag, mem_blk_t auth)

Функция потокового получения информации о CMS-сообщениях в виде файла

Соответствует функции VCERT_CmsStrMsgInfFile.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **data** (in) файл (не нулевой длины) с CMS-сообщением;
- **info** (out) структура с информацией о CMS-сообщении (освобождается функцией);
- **flag** (in) маска флагов (зарезервировано, должно быть равно **0**);

- **auth** (in) аутентификационные данные.

1.5.26 Функции блочной простановки и проверки штампов времени

int **CRSRVLib.CryptoServerLibrary.VCERTR_TspBlkRequestFromCms** (tsp_request_param_t pRequestPara, mem_blk_t icms, mem_blk_t oreq, mem_blk_t auth)

Функция блочного создания запроса на получение штампа времени блока памяти

Соответствует функции VCERT_TspBlkRequestFromCmsMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pRequestPara** (in) параметры простановки штампа времени CMS-сообщений;

- **icms** (in) блок памяти с подписанным CMS-сообщением;

- **oreq** (out) блок памяти с запросом на получение штампа времени;

- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_TspSignResponse** (tsp_response_param_t pResponsePara, mem_blk_t ireq, mem_blk_t ores, mem_blk_t auth)

Функция блочного вычисления ЭП и формирования штампов времени

Соответствует функции VCERT_TspSignResponse.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pResponsePara** (in) параметры вычисления ЭП и формирования штампов времени;

- **ireq** (in) блок памяти с запросом на получение штампа времени;

- **ores** (out) блок памяти с сформированным штампом времени;

- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_TspVerifyResponse** (tsp_verify_param_t pVerifyPara, mem_blk_t ires, tsp_verify_result_t pVerifyResult, mem_blk_t auth)

Функция блочной проверки ЭП сформированного штампа времени

Соответствует функции VCERT_TspVerifyResponse.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки штампа времени CMS-сообщений;

- **ires** (in) блок памяти с сформированным штампом времени;

- **pVerifyResult** (out) результат проверки штампа времени для заданной ЭП;

- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_TspBlkUrlStampCms** (tsp_

request_param_t pRequestPara, String url, mem_blk_t icms, mem_blk_t ocms, mem_blk_t auth)

Функция блочной простановки штампа времени блока памяти CMS на сервере
Соответствует функции VCERT_TspBlkUrlStampCmsMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pRequestPara** (in) параметры простановки штампа времени CMS-сообщений;
- **url** (in) строка (URI) с адресом сервера штампов времени;
- **icms** (in) блок памяти с подписанным CMS-сообщением;
- **ocms** (out) блок памяти с подписанным CMS-сообщением, включающим штамп времени;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERT_TspBlkVerifyCms** (tsp_verify_param_t pVerifyPara, mem_blk_t icms, tsp_verify_result_t pVerifyResult, mem_blk_t auth)

Функция блочной проверки штампа времени блока памяти CMS
Соответствует функции VCERT_TspBlkVerifyCmsMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки штампа времени CMS-сообщений;
- **icms** (in) блок памяти с подписанным CMS-сообщением, включающим штамп времени;
- **pVerifyResult** (out) результат проверки штампа времени для заданной ЭП;
- **auth** (in) аутентификационные данные.

1.5.27 Функции потоковой простановки и проверки штампов времени

int **CRSRVLib.CryptoServerLibrary.VCERT_TspStrUrlStampCmsInit** (tsp_request_param_t pRequestPara, String url, mem_blk_t icms, strmcs_handle_t hStr, mem_blk_t auth)

Функция инициализации потоковой простановки штампа времени CMS на сервере

Соответствует функции VCERT_TspStrUrlStampCmsInitMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pRequestPara** (in) параметры простановки штампа времени CMS-сообщений;
- **url** (in) строка (URI) с адресом сервера штампов времени;
- **icms** (in) блок памяти с началом подписанного CMS-сообщения;
- **hStr** (out) контекст потоковой операции;

- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_TspStrUrlStampCmsUpdate**
(tsp_request_param_t pRequestPara, String url, strcms_handle_t hStr, mem_blk_t
icms, mem_blk_t ocms)

Функция продолжения потоковой простановки штампа времени CMS на сервере

Соответствует функции VCERT_TspStrUrlStampCmsUpdateMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pRequestPara** (in) параметры простановки штампа времени CMS-сообщений;

- **url** (in) строка (URI) с адресом сервера штампов времени;

- **hStr** (in) контекст потоковой операции;

- **icms** (in) блок памяти с продолжением подписанного CMS-сообщения;

- **ocms** (out) блок памяти с продолжением подписанного CMS-сообщения, включающего штамп времени.

int **CRSRVLib.CryptoServerLibrary.VCERTR_TspStrUrlStampCmsFinalMem**
(tsp_request_param_t pRequestPara, String url, strcms_handle_t hStr, mem_blk_t
ocms)

Функция финализации потоковой простановки штампа времени CMS на сервере

Соответствует функции VCERT_TspStrUrlStampCmsFinalMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pRequestPara** (in) параметры простановки штампа времени CMS-сообщений;

- **url** (in) строка (URI) с адресом сервера штампов времени;

- **hStr** (in) контекст потоковой операции;

- **ocms** (out) блок памяти с окончанием подписанного CMS-сообщения, включающего штамп времени.

int **CRSRVLib.CryptoServerLibrary.VCERTR_TspStrUrlStampCmsFile** ((tsp_request_param_t pRequestPara, String url, String in, String out, mem_blk_t
auth)

Функция потоковой простановки штампа времени файла CMS на сервере

Соответствует функции VCERT_TspStrUrlStampCmsFile.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pRequestPara** (in/out) параметры простановки штампа времени CMS-сообщений;

- **url** (in) строка (URI) с адресом сервера штампов времени;

- **in** (in) файл (не нулевой длины) с подписанным CMS-сообщением;

- **out** (out) файл с подписанным CMS-сообщением, включающим штамп вре-

мени;

- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERT_TspStrVerifyCmsInit** (tsp_verify_param_t pVerifyPara, mem_blk_t icms, strcms_handle_t hStr, mem_blk_t auth)

Функция инициализации потоковой проверки штампа времени блока памяти CMS

Соответствует функции VCERT_TspStrVerifyCmsInitMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки штампа времени CMS-сообщений;
- **icms** (in) блок памяти с началом подписанного CMS-сообщения, включающего штамп времени;
- **hStr** (out) контекст потоковой операции;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERT_TspStrVerifyCmsUpdate** (tsp_verify_param_t pVerifyPara, strcms_handle_t hStr, mem_blk_t icms)

Функция продолжения потоковой проверки штампа времени блока памяти CMS

Соответствует функции VCERT_TspStrVerifyCmsUpdateMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки штампа времени CMS-сообщений;
- **hStr** (in) контекст потоковой операции;
- **icms** (in) блок памяти с продолжением подписанного CMS-сообщения, включающего штамп времени.

int **CRSRVLib.CryptoServerLibrary.VCERT_TspStrVerifyCmsFinal** (tsp_verify_param_t pVerifyPara, strcms_handle_t hStr, tsp_verify_result_t pVerifyResult)

Функция финализации потоковой проверки штампа времени блока памяти CMS

Соответствует функции VCERT_TspStrVerifyCmsFinalMem.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки штампа времени CMS-сообщений;
- **hStr** (in) контекст потоковой операции;
- **pVerifyResult** (out) результат проверки штампа времени для заданной ЭП.

int **CRSRVLib.CryptoServerLibrary.VCERT_TspStrVerifyCmsFile** (tsp_verify_param_t pVerifyPara, String data, tsp_verify_result_t pVerifyResult, mem_blk_t auth)

Функция потоковой проверки штампа времени файла CMS

Соответствует функции VCERT_TspStrVerifyCmsFile.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки штампа времени CMS-сообщений;
- **data** (in) файл (не нулевой длины) с подписанным CMS-сообщением, включающим штамп времени;
- **pVerifyResult** (out) результат проверки штампа времени для заданной ЭП;
- **auth** (in) аутентификационные данные.

1.5.28 Функции получения online-статуса сертификата

int **CRSRVLib.CryptoServerLibrary.VCERTR_OcspCreateRequest** (oosp_request_param_t pRequestPara, mem_blk_t cert, mem_blk_t rqst, mem_blk_t auth)

Функция создания запроса на получение online-статуса сертификата

Соответствует функции VCERT_OcspCreateRequest.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pRequestPara** (in) параметры получения online-статуса сертификата;
- **cert** (in) блок памяти с сертификатом для получения online-статуса в DER-кодировке или PEM-формате;
- **rqst** (out) блок памяти с созданным запросом на получение online-статуса сертификата;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_OcspSignResponse** (oosp_response_param_t pResponsePara, mem_blk_t rqst, mem_blk_t resp, mem_blk_t auth)

Функция обработки запроса на получение online-статуса сертификата

Соответствует функции VCERT_OcspSignResponse.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pResponsePara** (in) параметры вычисления ЭП online-статуса сертификата;
- **rqst** (in) блок памяти с запросом на получение online-статуса сертификата;
- **resp** (out) блок памяти с подписанным online-статусом сертификата;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_OcspUrlObtainResponse** (oosp_request_param_t pRequestPara, String url, mem_blk_t cert, mem_blk_t resp, mem_blk_t auth)

Функция получения online-статуса сертификата на заданном OSCP сервере

Соответствует функции VCERT_OcspUrlObtainResponse.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pRequestPara** (in) параметры получения online-статуса сертификата;
- **url** (in) строка (URI) с адресом сервера OSCP ответчика;

- **cert** (in) блок памяти с сертификатом для получения online-статуса в DER-кодировке или PEM-формате;
- **resp** (out) блок памяти с подписанным online-статусом сертификата;
- **auth** (in) аутентификационные данные.

int **CRSRVLib.CryptoServerLibrary.VCERTR_OcspVerifyResponse** (ocsp_verify_param_t pVerifyPara, mem_blk_t resp, ocsp_verify_result_t pVerifyResult, mem_blk_t auth)

Функция проверки ЭП online-статуса сертификата

Соответствует функции VCERT_OcspVerifyResponse.

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **pVerifyPara** (in) параметры проверки ЭП online-статуса сертификата;
- **resp** (in) блок памяти с подписанным online-статусом сертификата;
- **pVerifyResult** (out) результат проверки ЭП online-статуса сертификата;
- **auth** (in) аутентификационные данные.

1.5.29 Функции выработки случайного числа заданной длины

int **CRSRVLib.CryptoServerLibrary.VCERTR_GenRandom** (int len, mem_blk_t random, mem_blk_t auth)

Функция генерации случайного блока данных

Возвращаемые значения:

VCERT_OK (0) в случае успеха или ненулевой код ошибки.

Аргументы:

- **len** (in) размер случайного блока данных;
- **random** (out) случайный блок данных;
- **auth** (in) аутентификационные данные.

2 ОПИСАНИЕ ОШИБОЧНЫХ СИТУАЦИЙ

Ниже (Таблица 1) приведено описание возможных ошибочных ситуаций. В левой колонке указано символьное имя ошибки и шестнадцатеричное значение ее кода, в правой колонке приведено детальное описание и причина возникновения ошибки.

Таблица 1 – Описание ошибочных ситуаций

Имя и код ошибки	Описание и причина возникновения ошибки
VCERT_OK (0x00000000)	Успешное завершение функции
VCERT_E_GENERIC (0xE0700001)	Общая (внутренняя) ошибка библиотеки. Указывает на возможную ошибку в самой библиотеке или на искажения в ее настройках
VCERT_E_INVALID_PARAMETER (0xE0700002)	В функцию был передан неверный параметр. Возникает в случае передачи нулевого указателя, неверно заполненной структуры объекта системы управления сертификатами (СУС) или параметров или при неверном размере блока памяти
VCERT_E_INVALID_CONTEXT (0xE0700003)	Неверный контекст библиотеки, потоковой или другой операции. Вероятно, искажены настройки профиля пользователя или обнаружена ошибка в синтаксисе конфигурационного файла pkil.conf
VCERT_E_OPERATION_NOT_SUPPORTED (0xE0700004)	Операция (или функция) не поддерживается. Выполнен вызов функции или операции, не поддерживаемой библиотекой или не разрешенной для контекста библиотеки
VCERT_E_INVALID_FLAG (0xE0700005)	В функцию был передан неверный флаг. В параметре или в структуре параметров функции указана неверная маска (битовое ИЛИ) флагов
VCERT_E_NO_MEMORY (0xE0700006)	Недостаточно оперативной памяти. Вероятно, произведен вызов блочной функции над слишком большим блоком памяти или файлом
VCERT_E_DIGEST (0xE0700007)	Ошибка вычисления хэш-значения. Вероятно, неверен объектный идентификатор (OID) алгоритма хэширования
VCERT_E_CERT_USAGE (0xE0700008)	Неверное использование сертификата. В рабочем сертификате отсутствует требуемое разрешенное использование ключа проверки ЭП/открытого ключа шифрования, регламент или расширенное использование ключа проверки ЭП/открытого ключа шифрования
VCERT_E_CERT_FIND_PRIVATE_KEY (0xE0700009)	Не найден ключ ЭП, соответствующий данному сертификату. Отсутствует ключевой носитель с требуемым ключом ЭП, неверен ПИН-код устройства типа смарт-карта или неверен пароль ключа ЭП
VCERT_E_CMS_ADD_SIGNATURE (0xE070000C)	Ошибка добавления ЭП к сообщению в формате CMS/PKCS#7. Вероятно, что недостаточно ресурсов для выполнения операции, произошел сбой аппаратного датчика случайных чисел (ДСЧ) или нет доступа к ФКН vdToken с неизвлекаемым ключом ЭП
VCERT_E_CMS_ASN1_DECODE (0xE070000F)	Ошибка выполнения ASN.1-распаковки сообщения в формате CMS/PKCS#7. Вероятно, CMS-сообщение повреждено или искажено
VCERT_E_CMS_ASN1_ENCODE (0xE0700010)	Ошибка выполнения ASN.1-упаковки сообщения в формате CMS/PKCS#7. Вероятно, возникла нехватка ресурсов для выполнения операции
VCERT_E_SIGN_HASH (0xE0700012)	Ошибка вычисления ЭП хэш-значения. Вероятно, неверна длина хэш-значения, произошел сбой аппаратного ДСЧ или нет доступа к ФКН vdToken с неизвлекаемым ключом ЭП
VCERT_E_VERIFY_POLICY (0xE0700013)	Ошибка добавления регламента в контекст проверки сертификата. Вероятно, объектный идентификатор (OID) регламента неверен
VCERT_E_VERIFY_EXTKEYUSAGE (0xE0700014)	Ошибка добавления расширенного использования ключа в контекст проверки сертификата. Вероятно, объектный идентификатор (OID) расширенного использования ключа проверки ЭП/открытого ключа шифрования неверен
VCERT_E_OVERFLOW (0xE0700016)	Ошибка переполнения - либо данные слишком велики, либо буфер слишком мал. Вероятно, произведен вызов блочной функции над слишком большим блоком памяти или файлом
VCERT_E_PKCS10_DAMAGED (0xE0700017)	PKCS#10 запрос на сертификат поврежден или искажен
VCERT_E_REVREQ_DAMAGED (0xE0700018)	Запрос на аннулирование сертификата поврежден или искажен
VCERT_E_VERIFY (0xE0700019)	Общая ошибка проверки ЭП CMS/PKCS#7 сообщения. Возникла ошибка при проверке хотя бы одной ЭП CMS-сообщения
VCERT_E_CMS_INVALID_TYPE (0xE0700022)	Неверный тип содержимого сообщения в формате CMS/PKCS#7. Вероятно, в функцию проверки ЭП передано зашифрованное CMS-сообщение или наоборот

Имя и код ошибки	Описание и причина возникновения ошибки
VCERT_E_CMS_NO_RECIPIENTS (0xE0700024)	Отсутствуют или неверны данные сертификатов получателей зашифрованного сообщения в формате CMS/PKCS#7. CMS-сообщение повреждено или искажено
VCERT_E_CMS_NOT_RECIPIENT (0xE0700026)	Владелец сертификата не является получателем зашифрованного сообщения в формате CMS/PKCS#7. Идентификатор рабочего сертификата отсутствует в списке получателей зашифрованного CMS-сообщения
VCERT_E_CMS_KEY_DECRYPT (0xE0700027)	Ошибка расшифрования сеансового ключа зашифрованного CMS/PKCS#7 сообщения. Вероятно, CMS-сообщение повреждено или искажено, или нет доступа к ФКН vdToken с неизвлекаемым закрытым ключом шифрования
VCERT_E_DATA_DECRYPT (0xE0700028)	Ошибка расшифрования блока данных. Вероятно, CMS-сообщение повреждено или искажено
VCERT_E_RANDOM (0xE0700029)	Ошибка генерации случайного числа. Вероятно, произошел сбой аппаратного ДСЧ
VCERT_E_OPEN_CONFIG (0xE071002A)	Ошибка доступа к конфигурационному файлу pkil.conf . В текущем рабочем каталоге процесса не найден конфигурационный файл pkil.conf
VCERT_E_READ_CONFIG (0xE071002B)	Ошибка разбора конфигурационного файла pkil.conf . Обнаружена ошибка в формате конфигурационного файла pkil.conf
VCERT_E_NO_DEFAULT_CONFIG (0xE071002C)	Профиль по умолчанию не указан в конфигурационном файле pkil.conf . Вероятно, была произведена попытка инициализации контекста библиотеки с профилем по умолчанию
VCERT_E_OPEN_PSTORE (0xE070002D)	Ошибка доступа к ПСП или к подписанному справочнику. Вероятно, путь (URI) к ПСП или подписанному справочнику неверен
VCERT_E_OPEN_LOCALSTORE (0xE070002E)	Ошибка доступа к ЛСП. Вероятно, путь (URI) к ЛСП неверен
VCERT_E_VERIFY_STORE_USAGE (0xE070002F)	Подписанный справочник имеет неверный идентификатор использования. Вероятно, произведена попытка использовать подписанное обновление от Центра сертификации (ЦС) или Центра регистрации (ЦР) вместо ПСП или наоборот
VCERT_E_VERIFY_STORE (0xE0700030)	Ошибка проверки целостности ПСП или подписанного справочника. Вероятно, произошла ошибка построения или проверки цепочки сертификата подписанта или подписанный справочник поврежден или искажен
VCERT_E_OPEN_LDAPSTORE (0xE0700031)	Ошибка доступа к ССС. Вероятно, путь (URI) к ССС неверен, отсутствует сетевое подключение к ССС или доступ к ССС запрещен из-за отсутствия билета Kerberos
VCERT_E_VERIFY_CERT (0xE0700034)	Ошибка построения и проверки цепочки сертификата. Вероятно, срок действия рабочего сертификата или ключа ЭП истек, не найдены сертификат ЦС или САС, необходимые для построения цепочки, или срок действия САС истек
VCERT_E_CERT_MISSING (0xE0700035)	Сертификат издателя не был найден в доступных справочниках. В доступных справочниках отсутствует сертификат ЦС, необходимый для построения цепочки, при этом не разрешен или отсутствует доступ к точкам AIA
VCERT_E_CERT_EXPIRED (0xE0700036)	Срок действия сертификата уже истёк
VCERT_E_CERT_DAMAGED (0xE0700037)	Сертификат повреждён или искажен
VCERT_E_CERT_BROKEN - CONSTRAINT (0xE0700038)	Нарушены базовые ограничения цепочки сертификата
VCERT_E_CERT_REVOKED (0xE0700039)	Сертификат был аннулирован издателем
VCERT_E_CERT_UNTRUSTED (0xE070003A)	Цепочка сертификации не оканчивается доверенным сертификатом. В ПСП отсутствует необходимый сертификат корневого ЦС
VCERT_E_CRL_MISSING (0xE070003B)	САС издателя не был найден в доступных справочниках. В доступных справочниках отсутствует САС, необходимый для построения цепочки, при этом не разрешен или отсутствует доступ к точкам CDP
VCERT_E_CRL_EXPIRED (0xE070003C)	Срок действия САС уже истек
VCERT_E_CRL_DAMAGED (0xE070003D)	САС повреждён или искажен
VCERT_E_CERT_BROKEN - HIERARCHY (0xE070003E)	Нарушено ограничение иерархии цепочки сертификата
VCERT_E_CHAIN_ERROR (0xE070003F)	Общая ошибка построения и проверки цепочки сертификата. Вероятно, цепочка слишком длинная
VCERT_E_INVALID_USAGE (0xE0700041)	Ошибка использования сертификата не по назначению. В проверяемом сертификате отсутствует требуемое разрешенное использование ключа проверки ЭП/открытого ключа шифрования, регламент или расширенное использование ключа проверки ЭП/открытого ключа шифрования

BAMБ.00096-06 33 02

Имя и код ошибки	Описание и причина возникновения ошибки
VCERT_E_INVALID_SIGNATURE (0xE0700042)	ЭП недостоверна. Проверяемые данные повреждены или искажены, или неверен ключ проверки ЭП, который был использован для проверки ЭП
VCERT_E_PUBKEY_NOT_FOUND (0xE0700043)	У сертификата неизвестный ключ проверки ЭП/открытый ключ шифрования. Вероятно, неверен алгоритм ключа проверки ЭП/открытого ключа шифрования сертификата
VCERT_E_UPDATECRL (0xE0700045)	Общая ошибка обновления САС. Вероятно, при критичном обновлении одного из САС, находящихся в ЛСП, произошла ошибка
VCERT_E_CERT_NOT_FOUND (0xE0700046)	Сертификат не был найден в доступных справочниках. При поиске в доступных справочниках не был найден ни один сертификат, удовлетворяющий заданному шаблону
VCERT_E_CERT_NOT_YET_VALID (0xE0700047)	Срок действия сертификата еще не наступил
VCERT_E_NO_ATTACHED_SIGNER (0xE070004A)	Сертификат подписанта отсутствует в сообщении в формате CMS/PKCS#7. Вероятно, был установлен флаг проверки ЭП FLAG_CMS_VERIFY_REQUIREATTACHEDSIGNER
VCERT_E_KERBEROS_FAILURE (0xE070004B)	Ошибка получения или обновления билета Kerberos. Вероятно, нет доступа к Центру распределения ключей (Key Distribution Center, KDC) или имя пользователя и пароль неверны
VCERT_E_KEY_EXPIRED (0xE070004C)	Ключ ЭП/закрытый ключ шифрования уже истек
VCERT_E_KEY_NOT_YET_VALID (0xE070004D)	Ключ ЭП/закрытый ключ шифрования еще недействителен
VCERT_E_CRL_NOT_YET_VALID (0xE070004E)	Срок действия САС еще не наступил
VCERT_E_INIT_CSP (0xE070004F)	Ошибка выполнения инициализации Средства КЗИ. Вероятно, Средство КЗИ не установлено, или его конфигурация искажена
VCERT_E_ENUM_OBJECTS (0xE0700050)	Ошибка доступа к справочнику при переборе объектов. Вероятно, путь (URI) к справочнику неверен, или отсутствует подключение к справочнику по сети
VCERT_E_ENUM_NO_MORE (0xE0700051)	В справочнике больше нет объектов для перебора. Перебор справочника завершен, все объекты были успешно считаны
VCERT_E_INVALID_X500_NAME (0xE0700052)	Текстовая строка, содержащая X.500-имя, имеет неверное представление. Вероятно, строка с X.500-именем искажена или содержит неверный RDN
VCERT_E_INVALID_HEX_STRING (0xE0700053)	Текстовая строка, содержащая шестнадцатеричное число, имеет неверное представление. Текстовая строка с шестнадцатеричным числом должна иметь вид 00:01:0E:0F
VCERT_E_CMS_STREAM_MISMATCH (0xE0700054)	Обнаружено несоответствие между потоковым признаком обрабатываемых данных и вызванной функцией. Вероятно, произошла попытка вызова блочной функции для обработки CMS-сообщения, имеющего ASN.1 кодировку неопределенной длины, или наоборот
VCERT_E_CMS_DETACH_MISMATCH (0xE0700055)	Обнаружено несоответствие между признаком отсоединенной ЭП обрабатываемых данных и вызванной функцией. Вероятно, произошла попытка вызова функции, предназначенной для обработки CMS-сообщений с присоединенными ЭП, для обработки CMS-сообщения с отсоединенными ЭП, или наоборот
VCERT_E_CMS_INVALID_DIGESTS (0xE0700056)	Отсутствуют или неверны данные алгоритмов хэширования подписанного сообщения в формате CMS/PKCS#7. Вероятно, CMS-сообщение повреждено или искажено
VCERT_E_CMS_INVALID_SIGNERS (0xE0700057)	Отсутствуют или неверны данные сертификатов подписантов подписанного сообщения в формате CMS/PKCS#7. Вероятно, CMS-сообщение повреждено или искажено
VCERT_E_CMS_INVALID_CIPHER (0xE0700058)	Зашифрованное сообщение в формате CMS/PKCS#7 содержит неизвестный или неверный алгоритм шифрования. Вероятно, CMS-сообщение повреждено или искажено
VCERT_E_CMS_DATA_SIGNING (0xE0700059)	Ошибка вычисления ЭП подписанного сообщения в формате CMS/PKCS#7. Вероятно, что недостаточно ресурсов для выполнения операции, произошел сбой аппаратного ДСЧ или нет доступа к ФКН vdToken с неизвлекаемым ключом ЭП
VCERT_E_CMS_OMAC_MISMATCH (0xE070005A)	Имитовставка зашифрованного сообщения в формате CMS/PKCS#7 не совпадает с вычисленной. Вероятно, CMS-сообщение повреждено или искажено
VCERT_E_FIND_SESSION (0xE0700081)	Требуемая сессия криптосервера не была найдена. В функцию библиотеки был передан идентификатор несуществующей сессии КС
VCERT_E_CMS_NOT_ENCRYPTED (0xE0700083)	CMS/PKCS#7-сообщение не зашифровано или формат сообщения поврежден или искажен. Вероятно, CMS-сообщение повреждено или искажено
VCERT_E_ADD_OBJECT (0xE0700087)	Ошибка добавления объекта в справочник сертификатов. Вероятно, такой объект уже существует или добавление объекта в ССС запрещено

ВАНБ.00096-06 33 02

Имя и код ошибки	Описание и причина возникновения ошибки
VCERT_E_TOO_MANY_CERTS_FOUND (0xE0720089)	Слишком много сертификатов найдено по уникальному критерию поиска. Вероятно, несколько сертификатов содержат один и тот же идентификатор ключа ЭП
VCERT_E_USER_CANCEL (0xE072008A)	Операция была отменена пользователем
VCERT_E_OPEN_INFILE (0xE070008B)	Ошибка открытия входного файла. Вероятно, путь или имя файла неверны или доступ к файлу запрещен
VCERT_E_OPEN_OUTFILE (0xE070008C)	Ошибка открытия выходного файла. Вероятно, путь или имя файла неверны или доступ к файлу запрещен
VCERT_E_READ_FILE (0xE070008D)	Ошибка чтения из входного файла. Вероятно, произошло искажение файловой системы
VCERT_E_WRITE_FILE (0xE070008E)	Ошибка записи в выходной файл. Вероятно, на файловой системе закончилось свободное пространство
VCERT_E_FILE_LENGTH (0xE070008F)	Неверный размер файла (нулевой или более 2Гб)
VCERT_E_DELETE_OBJECT (0xE0700091)	Ошибка удаления объекта из справочника сертификатов. Указанный объект не был удален из кэша контекста библиотеки или сессии КС или из ЛСП сессии КС по команде с АРМ УКС
VCERT_E_TOO_FEW_SIGNATURES (0xE0700092)	Подписанный документ содержит недостаточное количество ЭП. Вероятно, были установлены флаги проверки ЭП FLAG_CMS_VERIFY_DELETESIGNATURES или FLAG_CMS_VERIFY_MINIMUMSIGNATURES , или индекс ЭП для операции со штампом времени слишком велик
VCERT_E_GET_PUBKEY (0xE0700094)	Ошибка получения ключа проверки ЭП/открытого ключа шифрования сертификата. Вероятно, возникла нехватка ресурсов или неверен алгоритм ключа проверки ЭП/открытого ключа шифрования сертификата
VCERT_E_PKCS10_CREATE (0xE0700098)	Ошибка создания нового PKCS#10 запроса. Вероятно, произошла ошибка при генерации или записи ключа ЭП на ключевой носитель или XML шаблон имеет неверный формат
VCERT_E_PKCS10_SIGN (0xE070009A)	Ошибка вычисления ЭП PKCS#10 запроса. Вероятно, произошел сбой аппаратного ДСЧ или нет доступа к ФКН vdToken с неизвлекаемым ключом ЭП
VCERT_E_REVREQ_CREATE (0xE070009B)	Ошибка создания нового запроса на аннулирование. Вероятно, возникла нехватка ресурсов
VCERT_E_REVREQ_SIGN (0xE070009C)	Ошибка вычисления ЭП запроса на аннулирование. Вероятно, произошел сбой аппаратного ДСЧ или нет доступа к ФКН vdToken с неизвлекаемым ключом ЭП
VCERT_E_LOAD_PRIVATE_KEY (0xE070009D)	Ошибка загрузки ключа ЭП/закрытого ключа шифрования. Отсутствует ключевой носитель с требуемым ключом ЭП/закрытым ключом шифрования, неверен ПИН-код устройства типа смарт-карта или неверен пароль ключа ЭП/закрытого ключа шифрования
VCERT_E_ADD_SIGNER (0xE070009E)	Ошибка добавления ЭП к ЛСП или к подписанному справочнику. Вероятно, произошел сбой аппаратного ДСЧ или нет доступа к ФКН vdToken с неизвлекаемым ключом ЭП
VCERT_E_OPEN_IDP (0xE07000A1)	Ошибка доступа к точке распространения САС. Вероятно, к точке CDP запрещен доступ или в точке CDP отсутствует требуемый САС
VCERT_E_READ_IDP (0xE07000A2)	Ошибка чтения из точки распространения САС. Вероятно, возникла проблема с сетевым подключением или в точке CDP отсутствует требуемый САС
VCERT_E_INVALID_CREDENTIALS (0xE02000A8)	Ошибочные данные аутентификации при доступе к сессии криптосервера. Вероятно, длина данных аутентификации равна 0
VCERT_E_ACCESS_DENIED (0xE02000A9)	Доступ к сессии криптосервера запрещен. Вероятно, данные аутентификации неверны
VCERT_E_SESSION_BLOCKED (0xA02000AA)	Сессия криптосервера заблокирована
VCERT_E_CLIENT_INFO (0xE02000AB)	Ошибка получения информации о клиенте из протокола DCE-RPC. Вероятно, произошла системная ошибка библиотеки DCE-RPC при получении сетевого адреса клиента
VCERT_E_UNSECURE_CREDENTIALS (0xE02000AC)	Небезопасные (слишком короткие) данные аутентификации сессии криптосервера. Длина данных аутентификации должна быть не менее 8 символов
VCERT_E_SESSION_TIMEOUT (0xA07000AE)	Истек интервал ожидания доступа к сессии КС из-за того, что данная сессия КС в настоящий момент заблокирована и не готова обрабатывать поступающие запросы. Данная ошибка может возникнуть, только если для данной сессии КС настроен ненулевой интервал ожидания доступа

Имя и код ошибки	Описание и причина возникновения ошибки
VCERT_E_TSP_HASH_LENGTH (0xE0700100)	Неверная длина хэш-значения при создании запроса на штамп времени. Длина хэш-значения не соответствует указанному алгоритму хэширования
VCERT_E_TSP_HASH_ALGORITHM (0xE0700101)	Неверный алгоритм хэширования при создании запроса на штамп времени. Объектный идентификатор (OID) алгоритма хэширования неверен
VCERT_E_TSP_CERT_PURPOSE (0xE0700102)	Сертификат не может быть использован для подписи штампов времени. Сертификат не удовлетворяет условиям использования на сервере штампов времени
VCERT_E_TSP_SIGN_FAILED (0xE0700103)	Ошибка вычисления ЭП штампа времени. Вероятно, произошел сбой аппаратного ДСЧ или нет доступа к ФКН vdToken с неизвлекаемым ключом ЭП
VCERT_E_TSP_NO_DIGEST (0xE0700104)	В списке атрибутов отсутствует хэш-значение ЭП и/или данных. Вероятно, штамп времени поврежден или искажен
VCERT_E_TSP_INVALID_SIGNER_NUM (0xE0700105)	Штамп времени содержит неверное количество ЭП. Вероятно, штамп времени поврежден или искажен
VCERT_E_TSP_NO_TST_INFO (0xE0700106)	Ошибка при получении информационного блока штампа времени. Вероятно, штамп времени поврежден или искажен
VCERT_E_TSP_RESP_ASN1_DECODE (0xE0700107)	Ошибка выполнения ASN.1-распаковки подписанного штампа времени. Вероятно, штамп времени поврежден или искажен
VCERT_E_TSP_RESP_NOT_ISSUED (0xE0700108)	Штамп времени не был выдан авторитетным источником. Вероятно, произошла внутренняя ошибка сервера штампов времени
VCERT_E_TSP_DIGEST_MISMATCH (0xE0700109)	Штамп времени содержит хэш-значение, отличное от хэш-значения ЭП CMS/PKCS#7 сообщения. Вероятно, штамп времени поврежден или искажен
VCERT_E_OCSP_CERT_PURPOSE (0xE0700140)	Сертификат не может быть использован для вычисления ЭП ответов сетевого ответчика. Сертификат не удовлетворяет условиям использования на сервере OCSP ответчика
VCERT_E_OCSP_SIGN_FAILED (0xE0700141)	Ошибка вычисления ЭП ответа сетевого ответчика. Вероятно, произошел сбой аппаратного ДСЧ или нет доступа к ФКН vdToken с неизвлекаемым ключом ЭП
VCERT_E_OCSP_RESP_ASN1_DECODE (0xE0700142)	Ошибка выполнения ASN.1-распаковки подписанного ответа сетевого ответчика. Вероятно, ответ сервера OCSP ответчика поврежден или искажен
VCERT_E_OCSP_RESP_NOT_ISSUED (0xE0700143)	Подписанный ответ не был выдан сетевым ответчиком. Вероятно, произошла внутренняя ошибка сервера OCSP ответчика
VCERT_E_OCSP_NOT_BASICRESP (0xE0700144)	Неверный (небазовый) тип подписанного ответа сетевого ответчика. Вероятно, ответ сервера OCSP ответчика содержит статус более чем для одного сертификата
VCERT_E_OCSP_CERTID_MISMATCH (0xE0700145)	Идентификатор сертификата из подписанного ответа сетевого ответчика не соответствует запрашиваемому. Вероятно, ответ сервера OCSP ответчика поврежден или искажен
VCERT_E_OCSP_ISSUER_MISMATCH (0xE0700146)	Издатель сертификата сетевого ответчика не соответствует издателю проверяемого сертификата. Вероятно, ответ сервера OCSP ответчика поврежден или искажен
VCERT_E_TLS_UNSUPPORTED (0xE0700180)	Функции протокола TLS не могут быть использованы с данным контекстом библиотеки. Вероятно, произведена попытка использования функций протокола TLS с минимальным контекстом библиотеки
VCERT_E_TLS_NEW_CONTEXT (0xE0700181)	Ошибка создания контекста нового сеанса связи протокола TLS. Вероятно, возникла нехватка ресурсов, произошел сбой аппаратного ДСЧ или использован контекст проверки библиотеки
VCERT_E_TLS_INVALID_STATE (0xE0700182)	Контекст сеанса связи протокола TLS находится в неверном состоянии. Вероятно, произведена попытка обмена данными между клиентом и сервером, когда защищенный канал еще не сформирован
VCERT_E_TLS_HANDSHAKE (0xE0700183)	Ошибка выполнения переговоров при формировании нового сеанса связи протокола TLS. Клиент и сервер не смогли сформировать защищенный канал, вероятно из-за отсутствия общих наборов криптографических алгоритмов
VCERT_E_TLS_NOT_COMPLETE (0xE0700184)	Переговоры при формировании нового сеанса связи протокола TLS еще не завершены
VCERT_E_TLS_NO_QUERY_DATA (0xE0700185)	Запрашиваемые данные в контексте сеанса связи протокола TLS отсутствуют. Вероятно, на сервере был выполнен запрос на получение сертификата клиента в DER-кодировке при выполнении односторонней аутентификации
VCERT_E_TLS_WRONG_CERT (0xE0700186)	Сертификат противоположной стороны сеанса связи TLS протокола неверен или искажен
VCERT_E_TLS_WRONG_NAME (0xE0700187)	Сертификат противоположной стороны сеанса связи TLS протокола имеет неверное имя. Вероятно, сертификат сервера имеет в дополнении "Альтернативное имя владельца" DNS-имя отличное от того, которое указал клиент

Имя и код ошибки	Описание и причина возникновения ошибки
VCERT_E_TLS_WRITE_ERROR (0xE0700188)	Ошибка при записи данных сеанса связи протокола TLS. Вероятно, возникла нехватка ресурсов или данные TLS протокола искажены
VCERT_E_TLS_READ_ERROR (0xE0700189)	Ошибка при чтении данных сеанса связи протокола TLS. Вероятно, данные TLS протокола искажены
VCERT_E_TLS_READ_MORE (0xE0700190)	Следует продолжить чтение данных сеанса связи протокола TLS. Вероятно, необходимо продолжить переговоры для завершения создания защищенного канала
ERR_PROFILES_BAD_PARAM (0xE0D50001)	При вызове какой-либо функции библиотеки ей передан параметр с недопустимым значением - скорее всего нулевой указатель
ERR_PROFILES_BUFFER_SIZE (0xE0D50002)	При работе со строками (копирование, чтение из реестра и пр.) размер выделенного буфера недостаточен для размещения строки
ERR_PROFILES_NO_MEMORY (0xE0D50003)	Ошибка выделения памяти - либо произошло исчерпание памяти системы, либо при выделении памяти запрошен неадекватный размер
ERR_PROFILES_GET_INSTANCE (0xE0D50004)	Не инициализирована переменная CRYPTO_hinstance, содержащая HINSTANCE исполняемого модуля, содержащего ресурсы
ERR_PROFILES_CREATE_DLG (0xE0D50005)	Ошибка инициализации модального диалога - скорее всего испорчены ресурсы или неправильно инициализирована переменная CRYPTO_hinstance, содержащая HINSTANCE исполняемого модуля, содержащего ресурсы
ERR_PROFILES_GET_DLG_ITEM (0xE0D50006)	Ошибка доступа к элементам управления (кнопка, поле редактирования, список и т.д.) модального диалога - скорее всего испорчены ресурсы
ERR_PROFILES_GET_USER_DIR (0xE0D50007)	Ошибка при вызове функции SHGetFolderPath() библиотеки shell32.dll, возвращающей каталог пользователя по умолчанию - скорее всего проблемы с файловой системой
ERR_PROFILES_GET_WND_RECT (0xE0D50008)	Ошибка функции GetWindowRect() получающей координаты окна. Глобальные проблемы системы
ERR_PROFILES_SET_WND_POS (0xE0D50009)	Ошибка функции SetWindowPos() устанавливающей положение окна. Глобальные проблемы системы
ERR_PROFILES_CLN_TO_SCR (0xE0D5000A)	Ошибка функции ScreenToClient() приводящей экранные координаты окна к клиентским. Глобальные проблемы системы
ERR_PROFILES_USER_CANCEL (0xE0D5000B)	Пользователь нажал кнопку "Отмена" или клавишу ESC
ERR_PROFILES_NO_REG_KEY (0xE0D5000C)	Отсутствует ключ реестра
ERR_PROFILES_DONT_OPEN_- REG_KEY (0xE0D5000D)	Ошибка открытия ключа реестра
ERR_PROFILES_DONT_CREATE_- REG_KEY (0xE0D5000E)	Ошибка создания ключа реестра
ERR_PROFILES_ACCESS_DENY_- REG_KEY (0xE0D5000F)	Недостаточно прав для создания ключа в реестре
ERR_PROFILES_DONT_DEL_- REG_KEY (0xE0D50010)	Ошибка удаления ключа реестра
ERR_PROFILES_AC_DENY_DEL_- REG_KEY (0xE0D50011)	Недостаточно прав для удаления ключа в реестре
ERR_PROFILES_NO_REG_VAL (0xE0D50012)	Отсутствует значение в реестре
ERR_PROFILES_DONT_READ_- REG_VAL (0xE0D50013)	Ошибка чтения значения в реестре
ERR_PROFILES_DONT_WRITE_- REG_VAL (0xE0D50014)	Ошибка записи значения в реестр
ERR_PROFILES_ACCESS_DENY_- REG_VAL (0xE0D50015)	Недостаточно прав для записи значения в реестр
ERR_PROFILES_BAD_TYPE_- REG_VAL (0xE0D50016)	Неправильный тип значения в реестре
ERR_PROFILES_DONT_ENUM_- REG_VAL (0xE0D50017)	Ошибка перечисления значений в ключе реестра

Имя и код ошибки	Описание и причина возникновения ошибки
ERR_PROFILES_NO_PROFILE (0xE0D50018)	При попытке выбора профиля (не в режиме редактирования) в реестре не обнаружено ни одного профиля либо при записи информации о профилях в реестр не было сформировано ни одного профиля
ERR_PROFILES_BAD_CONFIG (0xE0D50019)	Либо в реестре содержится неадекватное (меньше 2) значение параметра "count" обозначающего количество хранилищ для профиля. Либо в конфигурационном файле профиля (cfg.ini) в разделе [ODBC] не задан или задан пустой параметр local.gdbm при параметре local.gdbm_type равном 2
ERR_PROFILES_PROFILE_NOT_FOUND (0xE0D5001A)	Не найден профиль с заданным именем
ERR_PROFILES_PROF_ALREADY_EXISTS (0xE0D5001B)	При попытке добавления нового профиля без флага, разрешающего перезапись, обнаружено, что профиль с таким именем уже есть
ERR_PROFILES_BAD_PROF_INDEX (0xE0D5001C)	Не найден профиль с заданным номером
ERR_PROFILES_FILE_INSTEAD_DIR (0xE0D5001D)	При попытке создания директории (каталога) для хранения профиля обнаружено, что существует файл с таким именем
ERR_PROFILES_AC_DENY_CREATE_DIR (0xE0D5001E)	Недостаточно прав для создания директории (каталога)
ERR_PROFILES_CREATE_DIR_NO_PARENT (0xE0D5001F)	Попытка создать поддиректорию (подкаталог) отсутствующей директории (каталога)
ERR_PROFILES_CREATE_DIR_NO_ROOT (0xE0D50020)	Попытка создать поддиректорию (подкаталог) при отсутствии корня (например, диска)
ERR_PROFILES_DONT_CREATE_DIR (0xE0D50021)	Ошибка создания директории (каталога), не относящаяся к вышеперечисленным
ERR_PROFILES_ODBC (0xE0D50022)	Ошибка вызова функций SQLAllocHandle(), или SQLSetEnvAttr(), или SQLDriverConnect() библиотеки odbc32.dll. Проблемы библиотеки ODBC
ERR_PROFILES_BAD_LDAP_STRING (0xE0D50023)	Ошибка разбора строки LDAP-соединения
ERR_PROFILES_OPEN_MY_STORE (0xE0D50024)	Ошибка функции CertOpenStore() библиотеки Crypt32.dll, открывающей хранилище личных сертификатов
ERR_PROFILES_ENUM_MY_CERTS (0xE0D50025)	Ошибка функции CertEnumCertificatesInStore() библиотеки Crypt32.dll перечисляющей сертификаты из хранилища личных
ERR_PROFILES_GET_CERT_SUBJECT (0xE0D50026)	Ошибка функции CertNameToStr() Crypt32.dll, получающей имя владельца сертификата. Возможно, испорчен сертификат
ERR_PROFILES_NO_MY_CERTS (0xE0D50027)	Не найдено ни одного сертификата в хранилище личных сертификатов
ERR_PROFILES_FILETIME_TO_SYSTIME (0xE0D50028)	Ошибка функции FileTimeToLocalFileTime() или ф-ии FileTimeToSystemTime() библиотеки Kernel32.dll. Возможно, в сертификате указано неадекватное время
ERR_PROFILES_NO_SUBJ_KEY_ID (0xE0D50029)	В сертификате не найдено расширение 'Идентификатор ключа владельца'
ERR_PROFILES_DECODE_OBJECT (0xE0D5002A)	Ошибка функции CryptDecodeObject(), декодирующей объект в ASN1 кодировке. Возможно, испорчен сертификат
ERR_PROFILES_FIND_CERT_BY_KEYID (0xE0D5002B)	Ошибка поиска сертификата ф-ией CertFindCertificateInStore() с параметром CERT_FIND_CERT_ID по идентификатору ключа владельца. Возможно, сертификат отсутствует
ERR_PROFILES_SHOW_CERT (0xE0D5002C)	Ошибка функции CryptUIDlgViewContext(), отображающей сертификат

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

КЗИ	Криптографическая защита информации
КС	Криптографический сервер
ОС	Операционная система (Operating System)
ППИ	Прикладной программный интерфейс
ППО	Прикладное программное обеспечение
САС	Список аннулированных сертификатов
СКЗИ	Средство криптографической защиты информации
ЭП	Электронная подпись (Digital Signature)

ПЕРЕЧЕНЬ ТАБЛИЦ

1	Описание ошибочных ситуаций	52
---	---------------------------------------	----

[illegible][illegible]