127018, Москва, Сущёвский Вал, 18
Телефон: (495) 995 4820
https://CryptoPro.ru
E-mail: info@CryptoPro.ru

УТВЕРЖДЕНЫ
ЖТЯИ.00103-03 95 01-ЛУ

Средство Криптографической Защиты Информации

КриптоПро CSP Версия 5.0 R3 КС3 (исполнение 3-Base) Правила пользования

ЖТЯИ.00103-03 95 01
Листов 112

2024 г

© ООО «КРИПТО-ПРО», 2000-2024. Все права защищены.

Авторские права на средство криптографической защиты информации КриптоПро CSP и эксплуатационную документацию к нему зарегистрированы в Российском агентстве по патентам и товарным знакам (Роспатент).

Документ входит в комплект поставки программного обеспечения СКЗИ КриптоПро CSP версии 5.0 R3 КС3; на него распространяются все условия лицензионного соглашения. Без специального письменного разрешения ООО «КРИПТО-ПРО» документ или его часть в электронном или печатном виде не могут быть скопированы и переданы третьим лицам с коммерческой целью.

Содержание

Перечень определений и сокращений . . . **6**

1 Назначение СКЗИ и его основные характеристики . . . **7**

2 Порядок распространения СКЗИ . . . **8**

3 Ключевая система и ключевые носители . . . **9**
3.1 Шифрование данных . . . 9
3.2 Создание и проверка ЭП . . . 9
3.3 Ключевые носители . . . 9
3.4 Ключевые контейнеры . . . 10
3.5 Сертификаты ключей . . . 11
3.6 Размеры ключей . . . 11
3.7 Сроки действия ключей . . . 12
3.8 Хранение ключей и ключевых носителей . . . 12
3.9 Взаимодействие с пользователем при работе с ключевыми носителями . . . 13
3.9.1 Счётчики операций при использовании ФКН . . . 13
3.9.2 Требование пароля контейнера/носителя для каждой операции . . . 14
3.9.3 Настройка политики паролей контейнеров . . . 14
3.10 Управление ключами . . . 15
3.10.1 Формирование ключей . . . 15
3.10.2 Компрометация ключей пользователя . . . 16
3.10.3 Уничтожение ключей . . . 16

4 Требования к встраиванию СКЗИ в прикладные системы и к проведению исследований СФ СКЗИ . . . **18**
4.1 Применение СКЗИ без исследований СФ . . . 18
4.2 Применение СКЗИ с проведением исследований по оценке влияния . . . 19
4.3 Применение СКЗИ с проведением тематических исследований . . . 19
4.4 Применение СКЗИ в поддерживаемом[4] программном обеспечении . . . 20
4.5 Требования при встраивании СКЗИ в прикладные системы . . . 21

5 Требования по обеспечению безопасности при эксплуатации СКЗИ . . . **24**
5.1 Общие требования по организации работ по защите от НСД . . . 24
5.2 Требования по размещению технических средств с установленным СКЗИ . . . 24
5.3 Требования по установке СКЗИ, общесистемного и специального ПО на ПЭВМ . . . 25
5.4 Требования по обеспечению защиты от НСД . . . 26
5.5 Требования по обеспечению физической безопасности сервера . . . 28
5.6 Требования по организации процедуры резервного копирования и хранения резервных копий ключевых носителей . . . 29
5.7 Требования по подключению СКЗИ для работы по общедоступным каналам передачи данных . . . 29
5.8 Требования по использованию модулей (средств) доверенной загрузки . . . 30

6 Требования по криптографической защите . . . **31**

7 Разбор конфликтных ситуаций, связанных с применением ЭП . . . **33**
7.1 Порядок разбора конфликтной ситуации . . . 33
7.2 Случаи невозможности проверки значения ЭП . . . 34

Литература . . . **35**

Приложение 1. Вспомогательные утилиты . . . **36**
Утилита контроля целостности cpverify . . . 36
Контроль целостности программного обеспечения с помощью алгоритмов хэширования . . . 36
Контроль целостности программного обеспечения с помощью алгоритмов подписи . . . 39
Утилита безопасного удаления файлов wipefile . . . 40
Утилита для формирования и экспорта ключей в формате PKCS#12 HsmGenArm . . . 41

Коды ошибок . . . 41
Параметры журналирования . . . 42
Требования безопасности при использовании утилиты . . . 42

Приложение 2. Перечень вызовов, использование которых при разработке систем на основе СКЗИ «КриптоПро CSP» версия 5.0 R3 КС3 (исполнение 3-Base) с учётом п.1.5 Формуляра ЖТЯИ.00103-03 30 01 возможно без дополнительных тематических исследований . . . 44
Интерфейс CryptoAPI . . . 44
Интерфейс cpcurl . . . 60
Интерфейс КриптоПро TSP SDK . . . 63
Интерфейс КриптоПро ЭЦП SDK . . . 81
Интерфейс КриптоПро ЭЦП Browser plug-in . . . 99

Аннотация

Данный документ содержит правила использования средства криптографической защиты информации (СКЗИ) «КриптоПро CSP» версия 5.0 R3 КС3 (исполнение 3-Base), его состав, назначение, ключевую систему, требования и условия эксплуатации.

Документ предназначен для администраторов (администраторов ИБ), осуществляющих установку, обслуживание и контроль за соблюдением требований к эксплуатации средств СКЗИ, для администраторов серверов, сетевых ресурсов предприятия и других работников службы информационной безопасности, осуществляющих настройку рабочих мест для работы со средствами СКЗИ, а также для пользователей СКЗИ.

Инструкции администраторам и пользователям различных автоматизированных систем, использующих СКЗИ «КриптоПро CSP» версия 5.0 R3 КС3 (исполнение 3-Base), должны разрабатываться с учётом требований настоящих Правил пользования.

Требования настоящих Правил пользования и эксплуатационной документации, входящей в комплект поставки СКЗИ, могут уточняться с учетом модели угроз и нарушителя ИС, в которой применяется СКЗИ, при проведении исследований ИС.

Перечень определений и сокращений

CRL	Список отозванных сертификатов (Certificate Revocation List)
CTL	Список доверенных сертификатов (Certificate Trust List)
HDD	Жесткий магнитный диск (Hard Disk Drive)
SSD	Твердотельный накопитель (Solid-State Drive)
АРМ	Автоматизированное рабочее место
АС	Автоматизированная система
БиоДСЧ	Биологический датчик случайных чисел
ГМД	Гибкий магнитный диск
ДСЧ	Датчик случайных чисел
ИС	Информационная система
МДЗ	Модуль доверенной загрузки
МП	Мобильное приложение
НСД	Несанкционированный доступ
ОС	Операционная система
ПАК	Программно-аппаратный комплекс
ПДСЧ	Программный датчик случайных чисел
ПК	Персональный компьютер
ПО	Программное обеспечение
ПЭВМ	Персональная электронно-вычислительная машина
СВТ	Средства вычислительной техники
СКЗИ	Средство криптографической защиты информации
СФ	Среда функционирования
УЦ	Удостоверяющий Центр
ЦР	Центр Регистрации
ЦС	Центр Сертификации
ЭД	Электронный документ
ЭП	Электронная подпись

1 Назначение СКЗИ и его основные характеристики

СКЗИ «КриптоПро CSP» версия 5.0 R3 КС3 (исполнение 3-Base) представляет собой программный комплекс, предназначенный для реализации широкого набора решений по обеспечению криптографическими методами информационной безопасности на отдельных рабочих местах, в архитектуре «клиент-сервер», а также в информационных и телекоммуникационных системах различного назначения.

СКЗИ КриптоПро CSP может выступать как в качестве готового к применению средства, так и в качестве платформы для построения на его основе программных, программно-аппаратных и аппаратных решений в области обеспечения информационной безопасности, основанных на применении криптографических алгоритмов.

СКЗИ КриптоПро CSP предназначено для защиты информации, не содержащей сведений, составляющих государственную тайну. Допускается использование СКЗИ для криптографической защиты персональных данных. Запрещено использование СКЗИ для защиты речевой информации.

Организация и обеспечение безопасности хранения, обработки и передачи по каналам связи с использованием СКЗИ информации с ограниченным доступом, не содержащей сведений, составляющих государственную тайну (информации конфиденциального характера), осуществляются в соответствии с [1, 2] и другими руководящими документами по обеспечению безопасности информации.

СКЗИ «КриптоПро CSP» версия 5.0 R3 КС3 (исполнение 3-Base) при выполнении требований эксплуатационной документации, входящих в комплектацию СКЗИ в соответствии с разделом 4 ЖТЯИ.00103-03 30 01. Формуляр, удовлетворяет требованиям к СКЗИ [3] для СКЗИ класса КС3, требованиям к средствам ЭП [4] для средств ЭП класса КС3 и требованиям [5, 6] для СКЗИ по уровню $КС_Б$.

Встраивание СКЗИ в аппаратные, программно-аппаратные и программные средства связи/системы и проведение исследований среды функционирования СКЗИ на выполнение предъявленных к СКЗИ требований должны выполняться в соответствии с разд. 4.

Использование СКЗИ «КриптоПро CSP» версия 5.0 R3 КС3 (исполнение 3-Base) с выключенным режимом усиленного контроля использования ключей **не допускается**. Включение данного режима описано в Руководствах администратора безопасности (для ОС Windows, Linux), входящих в состав эксплуатационной документации на СКЗИ.

2 Порядок распространения СКЗИ

Установочные модули СКЗИ «КриптоПро CSP» версия 5.0 R3 КС3 (исполнение 3-Base) и комплект эксплуатационной документации к нему могут поставляться пользователю Уполномоченной организацией[1] следующими способами:

1) на носителе (CD, DVD-диски);

2) посредством загрузки через информационно-телекоммуникационные сети (в том числе Интернет).

Для получения возможности загрузки установочных модулей СКЗИ и комплекта эксплуатационной документации пользователь направляет свои учётные данные Уполномоченной организации. Учётные данные могут быть направлены посредством заполнения специализированной регистрационной формы на сайте Уполномоченной организации.

После получения Уполномоченной организацией учётных данных пользователю предоставляется доступ на страницу загрузки установочных модулей СКЗИ и комплекта эксплуатационной документации. При загрузке пользователем установочных модулей СКЗИ и комплекта эксплуатационной документации Уполномоченной организацией присваивается учётный номер, идентифицирующий экземпляр СКЗИ, предоставленный пользователю.

На странице загрузки вместе с дистрибутивом и документацией размещается отделенная электронная подпись, для проверки которой необходимо использовать утилиту `cpverify`, полученную доверенным образом и содержащую ключ проверки данной электронной подписи.

Установка СКЗИ на рабочее место пользователя может быть осуществлена только в случае подтверждения целостности полученных установочных модулей СКЗИ и эксплуатационной документации.

Примечание.
1) Средство контроля целостности (`cpverify`) первоначально должно быть получено пользователем на физическом носителе в офисе Уполномоченной организации. Такая утилита считается полученной доверенным образом. Далее полученной доверенным образом признается очередная версия утилиты, полученная любым образом (например, скачанная с сайта `https://cryptopro.ru/`), при условии, что она была проверена другим экземпляром утилиты, полученным ранее доверенным образом, и проверка была успешной.

2) Ключ проверки ЭП, а также информация о нем (дата создания, алгоритм хэш-функции, идентификатор алгоритма подписи) записываются в исходный код утилиты на этапе сборки.

3) Контроль целостности дистрибутива СКЗИ и компонентов среды функционирования (СФ) СКЗИ обеспечивается при помощи утилиты `cpverify` в соответствии с Приложением 1.

[1] ООО «КРИПТО-ПРО» или его официальный дилер/дистрибьютор

3 Ключевая система и ключевые носители

СКЗИ «КриптоПро CSP» версия 5.0 R3 КС3 (исполнение 3-Base) является системой с открытым распределением ключей на основе асимметричной криптографии с использованием закрытого ключа на одной стороне и соответствующего ему открытого ключа на другой стороне. Такие пары ключей могут быть двух типов: ключи электронной подписи (ЭП) и ключи обмена.

Пользователь включается в систему и исключается из неё установленным Удостоверяющим центром порядком.

Пользователь, обладающий правом подписи и/или шифрования данных, вырабатывает на своем рабочем месте или получает у Администратора безопасности (в зависимости от принятой политики безопасности) личные закрытый ключ (ключ ЭП) и открытый ключ (ключ проверки ЭП). На основе каждого открытого ключа (ключа проверки ЭП) Удостоверяющим Центром формируется сертификат открытого ключа (сертификат ключа проверки ЭП).

3.1 Шифрование данных

При зашифровании сообщения пользователем А для пользователя Б, пользователь А формирует симметричный ключ связи (сеансовый ключ информационного обмена) на основе своего закрытого ключа обмена и открытого ключа обмена пользователя Б. Соответственно, для расшифрования этого сообщения пользователем Б формируется тот же симметричный ключ на основе своего закрытого ключа обмена и открытого ключа обмена пользователя А.

Таким образом, для обмена данными каждому пользователю необходимо иметь:

- личный закрытый ключ обмена;
- открытые ключи обмена других пользователей.

Сеансовый ключ информационного обмена вырабатывается на основе схемы Диффи-Хеллмана. Схема Диффи-Хеллмана обеспечивает формирование сеансовых ключей, но не обеспечивает аутентификацию связывающихся сторон. Поэтому данная схема должна использоваться совместно с протоколами аутентификации.

3.2 Создание и проверка ЭП

Для создания электронной подписи используется ключ ЭП, для проверки — соответствующий ключ проверки ЭП. Проверяющий должен быть полностью уверен в принадлежности ключа проверки ЭП конкретному пользователю. Для этой цели используется сертификат ключа проверки ЭП, подписанный УЦ.

Каждому пользователю, обладающему правом подписи, необходимо иметь:

- ключ ЭП;
- ключи проверки ЭП (сертификаты ключей проверки ЭП) других пользователей.

Примечание. Совмещение использования личного ключа для создания ЭП и построения TLS–соединений не допускается. Для указанных операций должны использоваться различные ключи.

3.3 Ключевые носители

Ключи пользователей могут храниться на различных устройствах, называемых «ключевые носители» («ключевые хранилища»). Ключи на ключевых носителях хранятся в виде специальной структуры, называемой «ключевой контейнер».

Примечание. Виды поддерживаемых ключевых носителей и их использование в зависимости от программно-аппаратной платформы отражены в ЖТЯИ.00103-03 30 01. Формуляр, п. 3.10.

С точки зрения принципов использования ключей ключевые носители подразделяются на следующие типы:

- **Функциональный ключевой носитель (ФКН)**. Такой ключевой носитель содержит в себе функциональные

возможности генерации ключей, обеспечения невозможности экспорта хранимых ключей за пределы носителя и содержит в себе реализацию криптографических алгоритмов, выполняемых с использованием хранимых на носителе ключей (т.н. активный вычислитель). Конкретный набор функций и их реализация зависят от ФКН конкретного разработчика и от модуля поддержки ФКН в СКЗИ. Важной особенностью при работе с ФКН, поддерживаемыми СКЗИ КриптоПро CSP, является защита канала связи между ФКН и СКЗИ по протоколу SESPAKE. Таким образом обеспечивается конфиденциальность и неизменность передаваемых между СКЗИ и носителем данных. Для некоторых носителей такая поддержка отсутствует (см. соответствующие примечания к таблице 3.1 документа ЖТЯИ.00103-03 30 01. Формуляр).

• **Пассивный ключевой носитель** (пассивное хранилище ключевой информации). Такой ключевой носитель предназначен только для хранения ключевых контейнеров пользователей. При необходимости использования такого ключевого носителя требуемый ключевой контейнер обрабатывается средствами СКЗИ.

• **КриптоПро Cloud CSP** (модуль взаимодействия с сервисом хранения ключей). Обеспечивает те же функциональные возможности, что и модуль взаимодействия с ФКН, предоставляя интерфейс для взаимодействия с сервером в целях выполнения операций с ключами, хранящимися в сервисе хранения ключей. При этом защита канала связи между СКЗИ и сервисом обеспечивается по протоколу TLS. Необходимо наличие сертификата соответствия ФСБ России (положительного заключения) на средство электронной подписи, реализующее на серверной стороне указанный сервис хранения ключей.

Подробнее об использовании КриптоПро Cloud CSP см. п. 2.4 документа ЖТЯИ.00103-03 92 06. КриптоПро CSP. Инструкция по использованию графического приложения. Инструменты КриптоПро.

При использовании **пассивных** ключевых носителей и ФКН без поддержки SESPAKE необходимо обеспечить выполнение следующих условий:

• подключение съемных носителей должно осуществляться непосредственно к считывателю (считывателю смарт-карт, USB-портам и т.п.), с обеспечением отсутствия канала связи между носителем и СКЗИ, в котором может действовать нарушитель;

• при конструктивном исполнении считывателя ключевого носителя с кабелем необходимо обеспечить нахождение считывателя и кабеля на той же контролируемой территории, что и ПЭВМ, и отсутствие доступа нарушителя к ним.

При невозможности выполнения указанных условий необходимо с учётом модели возможных угроз и нарушителя разработать организационно-технические мероприятия по защите взаимодействия носителя с СКЗИ с последующей оценкой таких мероприятий в рамках проведения соответствующих исследований.

Примечание. При удаленном подключении ключевых носителей необходимо использовать ФКН с поддержкой SESPAKE.

При использовании любых типов ключевых носителей помимо требований эксплуатационной документации на СКЗИ КриптоПро CSP необходимо руководствоваться эксплуатационной документацией на сами носители.

Доступ к ключевому носителю любого типа должен быть защищён паролем и/или PIN-кодом (за исключением случаев, отдельно оговоренных в документации на СКЗИ КриптоПро CSP и используемые носители). Носители могут поддерживать код доступа PUK, используемый для разблокировки носителя. Перед началом работы с ключевым носителем необходимо сменить установленные по умолчанию значения PIN и PUK (при поддержке PIN и PUK на носителе).

3.4 Ключевые контейнеры

Закрытые ключи хранятся в ключевом контейнере, который может содержать в себе:

- только ключ ЭП;
- только ключ обмена;
- ключ ЭП и ключ обмена одновременно.

Единственный ключ ключевого контейнера (ключ ЭП или ключ обмена) называется **первичным** ключом. Если в контейнере два ключа, то первый ключ (ключ ЭП) называется первичным, второй ключ (ключ обмена) — **вторичным**.

Кроме ключей, ключевой контейнер содержит служебную информацию, необходимую для обеспечения криптографической защиты ключей, их целостности и т.п.

Каждый ключевой контейнер (независимо от типа носителя), является самодостаточным и содержит всю необходимую информацию для работы как с самим контейнером, так и с закрытыми (и соответствующими им открытыми) ключами.

Ключевой контейнер содержит следующую информацию:

- первичный ключ;
- маски первичного ключа;
- контрольную информацию первичного ключа;
- вторичный ключ (опциональный).

Каждый закрытый ключ хранится в формате, дополнительно содержащем все константы, необходимые для формирования и экспорта открытого ключа.

Формат ключевого контейнера обеспечивает чтение ключей и соответствующих масок отдельными операциями в раздельные (разнесённые по адресам) области памяти, для чего он содержит шесть зон (реализация зон зависит от типа ключевого носителя).

Ключевой контейнер содержит также дополнительную информацию, необходимую для обеспечения его восстановления при возникновении различных программно-аппаратных сбоев (дополнительная информация включается в тех случаях, когда размер ключевого контейнера не ограничен размерами памяти физического носителя).

Ключевой контейнер может быть представлен в виде QR-кода. На носитель, содержащий QR-код ключевого контейнера, распространяются все требования по обращению с ключевыми носителями.

3.5 Сертификаты ключей

Сертификат открытого ключа обмена и сертификат ключа проверки ЭП представляет собой структурированную двоичную запись в формате ASN.1, содержащую:

- имя субъекта или объекта системы, однозначно идентифицирующее его в системе;
- открытый ключ или ключ проверки ЭП субъекта или объекта системы;
- дополнительные атрибуты, определяемые требованиями использования сертификата в системе;
- ЭП Издателя (Удостоверяющего центра), заверяющую совокупность этих данных.

3.6 Размеры ключей

Размеры ключей электронной подписи:

ключ электронной подписи	256 бит или 512 бит;
ключ проверки электронной подписи	512 бит или 1024 бита.

Размеры ключей, используемых при шифровании:

закрытый ключ	256 бит или 512 бит;
открытый ключ	512 бит или 1024 бита;
симметричный ключ	256 бит.

3.7 Сроки действия ключей

При эксплуатации СКЗИ КриптоПро CSP должны соблюдаться следующие сроки использования пользовательских ключей[2]:

- максимальный срок действия ключа ЭП — 1 год 3 месяца;
- максимальный срок действия ключа проверки ЭП — 15 лет после окончания срока действия соответствующего ключа ЭП;
- максимальный срок действия закрытых и открытых ключей обмена — 1 год 3 месяца;
- максимальный срок действия ключей ЭП/закрытых ключей обмена в неэкспортируемом виде при их использовании в ФКН, КриптоПро Cloud CSP или КриптоПро HSM — 3 года.

При формировании закрытого ключа в контейнер записывается дата истечения срока действия этого ключа, по истечении которого в зависимости от значения параметра **ControlKeyTimeValidity** возможны различные варианты использования этого ключа:

- Значение «0» параметра не накладывает никаких ограничений на использование ключа.
- Значение «1» параметра запрещает формирование ЭП и шифрование в контексте этого ключа (возможно расшифрование ранее зашифрованных сообщений) (*значение по умолчанию*);
- Значение «2» параметра запрещает любые действия с закрытым ключом.

Срок действия ключа определяется следующим образом:

1) Определяется значение срока действия из расширения контейнера. Если в расширении указан срок окончания действия ключа (*NotAfter, Действителен по*), то выбирается это значение. Иначе срок определяется от начала действия ключа (*NotBefore, Действителен с*) + 1 год 3 месяца для пассивных ключевых носителей или + 3 года для неэкспортируемых ключей при их использовании в ФКН и КриптоПро Cloud CSP.

2) Аналогичным образом определяется значение из расширения сертификата. Если расширение в сертификате отсутствует, то сертификат не ограничивает срок действия закрытого ключа.

3) В качестве значения срока действия ключа выбирается наименьшее из полученных значений.

Изменение параметра ControlKeyTimeValidity

Для операционных систем Windows необходимо изменить значение ключа реестра `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Crypto Pro\Cryptography\CurrentVersion\Parameters\ControlKeyTimeValidity` (для 64-битных ОС), `HKEY_LOCAL_MACHINE\SOFTWARE\Crypto Pro\Cryptography\CurrentVersion\Parameters\ControlKeyTimeValidity` (для 32-битных ОС).

Настройка СКЗИ для остальных ОС осуществляется с помощью утилиты `cpconfig` с помощью команды:

```
./cpconfig -ini '\config\parameters' -add long ControlKeyTimeValidity <значение>
```

Примечание. При работе в режиме усиленного контроля использования ключей (режим обязателен к использованию, отключение может производиться только в целях тестирования) значение параметра `ControlKeyTimeValidity` принимается равным «2».

3.8 Хранение ключей и ключевых носителей

СКЗИ может функционировать и хранить ключевую информацию в двух режимах:

- в памяти приложения;
- в «Службе хранения ключей», реализованной в виде системного сервиса.

[2]Сроки действия ключей ЭП и закрытых ключей обмена могут уточняться при проведении работ по оценке влияния аппаратных, программно-аппаратных и программных средств сети (системы) конфиденциальной связи, совместно с которыми предполагается штатное функционирование СКЗИ, на выполнение предъявленных к СКЗИ требований по ТЗ, согласованному с 8 Центром ФСБ России.

Ключи находятся в кэше до завершения приложения или до выключения компьютера (остановки службы), что позволяет использовать закрытый ключ даже после закрытия криптографического контекста (только для пассивных хранилищ ключевой информации без использования криптографических механизмов, реализованных на смарт-карте/токене).

Функционирование и хранение ключей СКЗИ КриптоПро CSP в «Службе хранения ключей» обеспечивает дополнительную защиту ключевой информации от других приложений, выполняющихся на ПЭВМ, но может незначительно замедлить производительность системы.

При хранении ключей и ключевых носителей должны выполняться следующие требования:

• Необходимо обеспечить невозможность доступа к ключевым носителям не допущенных к ним лиц (например, с помощью использования опечатанного сейфа). Пользователь несёт персональную ответственность за хранение личных ключевых носителей.

• Запрещается оставлять без контроля вычислительные средства с установленным СКЗИ после ввода ключевой информации.

• В случае централизованного хранения ключевых носителей в организации, эксплуатирующей СКЗИ, Администратор безопасности несёт персональную ответственность за хранение личных ключевых носителей пользователей.

• Хранение ключей на несъемных носителях (в реестре ОС Windows, в разделе HDD/SSD ПЭВМ и т.п.) допускается только при условии распространения на носитель требований по обращению с ключевыми носителями (в том числе и после удаления ключей).

• В случае невозможности отчуждения ключевого носителя с ключевой информацией от ПЭВМ организационно-техническими мероприятиями должен быть исключен доступ нарушителей к ПЭВМ с ключами.

• Необходимо использовать парольную защиту, если не оговорено иное.

• В случае необходимости проведения ремонтных и регламентных работ аппаратной части среды функционирования (СФ) необходимо обеспечить невозможность доступа нарушителя к ключевой информации, содержащейся в СФ. Конкретный перечень мер должен быть определён исходя из условий эксплуатации СКЗИ.

3.9 Взаимодействие с пользователем при работе с ключевыми носителями

При работе с ключевыми носителями СКЗИ может использовать какой-либо пользовательский интерфейс (UI). Это может происходить, например, при необходимости выбрать носитель или ввести PIN. Чтобы отключить пользовательский интерфейс (например, для автоматизации), в некоторых приложениях существует опция **-silent**.

Также возможно запретить СКЗИ отображать пользовательский интерфейс глобально для всех приложений на данном ПК. Для этого в настройках СКЗИ нужно задать параметр **force_silent** значение «1» (см. ниже), force_silent равный «0» вернёт поведение по умолчанию. Если же вызовы функций требуют отображения пользовательского интерфейса, будет возвращена ошибка `NTE_SILENT_CONTEXT`.

Изменение параметра force_silent

Для операционных систем семейства Windows необходимо изменить значение ключа реестра `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Crypto Pro\Cryptography\CurrentVersion\Parameters\force_silent` (для 64-битных ОС), `HKEY_LOCAL_MACHINE\SOFTWARE\Crypto Pro\Cryptography\CurrentVersion\Parameters\force_silent` (для 32-битных ОС).

Настройка СКЗИ для остальных ОС осуществляется с использованием утилиты `cpconfig` с помощью команды:

```
./cpconfig -ini '\config\parameters' -add long force_silent 1
```

3.9.1 Счётчики операций при использовании ФКН

В носителях ФКН используются три декрементируемых счётчика аутентификации:

- C1 — общий счётчик попыток аутентификации;
- C2 — общий счётчик неуспешных попыток аутентификации;
- C3 — счётчик последовательных неуспешных попыток аутентификации.

Начальное значение счётчиков определяется производителем носителя. При успешной аутентификации счётчик C3 восстанавливается в значение по умолчанию. Если любой из счётчиков достигнет нулевого значения, исполнение защищённых операций на носителе блокируется.

Для восстановления счётчиков в начальное значение требуется сменить пароль.

СКЗИ КриптоПро CSP следит за данными счётчиками и заранее предупреждает пользователя о потенциальной блокировке носителя при дальнейших неуспешных попытках аутентификации и предлагает заранее сменить пароль.

3.9.2 Требование пароля контейнера/носителя для каждой операции

Каждый ключевой контейнер можно создать в режиме повышенной защиты. Созданный таким образом контейнер вне зависимости от ключевого носителя будет требовать предъявление пароля на каждую операцию. Изменить это свойство нельзя.

Для этого при формировании ключа в функцию CryptGenKey необходимо передать флаги (CRYPT_FORCE_KEY_PROTECTION_HIGH | CRYPT_USER_PROTECTED).

В случае импорта ключа из файла pfx с использованием утилиты `certmgr` для требования пароля на каждую операцию необходимо указать флаг `-protected high` (подробнее см. ЖТЯИ.00103-03 93 02. Приложение командной строки для работы с сертификатами).

В ОС семейства Windows настройка требования пароля на каждую операцию может быть изменена с помощью групповой политики «Требовать пароль контейнера/носителя для каждой операции» или с помощью ключа реестра `HKEY_LOCAL_MACHINE\SOFTWARE\[WOW6432Node]\Crypto Pro\Cryptography\CurrentVersion\Parameters\DisableCachePasswords` (см. раздел 4.6 ЖТЯИ.00103-03 91 02. Руководство администратора безопасности. Windows).

Настройка для остальных ОС осуществляется с использованием утилиты `cpconfig` с помощью команды:

```
./cpconfig -ini '\config\parameters' -add long DisableCachePasswords <значение>
```

Значения параметра DisableCachePasswords:

- 0 — не задано;
- 1 — для открываемых ключей: при открытии для всех контейнеров будут требоваться пароли на каждую операцию;
- 2 — для создаваемых ключей: для всех генерируемых на компьютере ключей будет требоваться пароль на каждую операцию;
- 3 — для всех: объединяет 2 предыдущих значения.

3.9.3 Настройка политики паролей контейнеров

Политика паролей указывается с помощью регулярного выражения.

В ОС Windows настройка политики осуществляется с помощью ключа реестра `HKEY_LOCAL_MACHINE\SOFTWARE\[WOW6432Node]\Crypto Pro\Cryptography\CurrentVersion\Parameters\PasswordPolicies\AsciiPasswordPolicy`. Значение по умолчанию — `[ -~]*` (все отображаемые ASCII-символы).

Настройка для остальных ОС осуществляется с использованием утилиты `cpconfig` с помощью команды (указано значение по умолчанию, все отображаемые ASCII-символы):

```
./cpconfig -ini '\config\parameters\PasswordPolicies' -add string "AsciiPasswordPolicy"
      "[ [:punct:][:digit:]ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz]*"
```

Для отключения политики нужно удалить соответствующую ей запись в конфигурации. Не рекомендуется удалять политику по умолчанию (AsciiPasswordPolicy), так как она обеспечивает корректный перенос контейнеров между разными операционными системами.

3.10 Управление ключами

Ключевая система СКЗИ КриптоПро CSP базируется на архитектуре PKI рекомендаций X.509. Формирование и управление сертификатами открытых ключей производится УЦ. В качестве УЦ может выступать Удостоверяющий центр «КриптоПро УЦ» или другие сертифицированные ФСБ России УЦ, обеспечивающие выполнение функций доверенного обращения с сертификатами. Также допускается использование Центра Сертификации корпорации Microsoft (Microsoft Certification Authority) в тестовых целях.

СКЗИ КриптоПро CSP может использоваться в качестве криптоядра в составе различных прикладных систем, организационные схемы управления ключевой системой которых могут отличаться от рассматриваемой.

Взаимодействие с компонентами УЦ при управлении ключами должно осуществляться в соответствии с Регламентом УЦ.

3.10.1 Формирование ключей

Формирование ключей может выполняться следующими способами:

1) вызовами функций генерации ключа (ключевой пары) из прикладного ПО с встроенным СКЗИ (подробнее см. разд. 4 и Приложение 2);

2) с использованием утилиты командной строки cryptcp (функции создания запроса на сертификат);

3) с использованием утилиты командной строки csptest.

Закрытые ключи ЭП и обмена формируются с использованием ПДСЧ с инициализацией его от БиоДСЧ, от внешней гаммы или от физического ДСЧ (подробнее см. п. 3.12 Формуляра).

Примечание. Ключевая информация является конфиденциальной. Внешняя гамма, используемая для инициализации состояния ПДСЧ, является конфиденциальной.

В связи с переходом на использование алгоритма ГОСТ Р 34.10-2012 и соответствующем запрете использования алгоритма ГОСТ Р 34.10-2001 [7], при попытке генерации ключа алгоритма ГОСТ Р 34.10-2001 будет выдано следующее предупреждение:

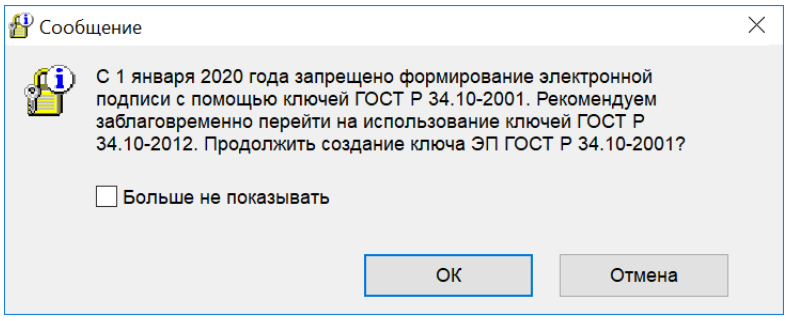


Рисунок 1. Предупреждение о генерации ключа алгоритма ГОСТ Р 34.10-2001 для ОС семейства Windows

При попытке создания подписи с использованием ключа алгоритма ГОСТ Р 34.10-2001 будет выдано следующее предупреждение:

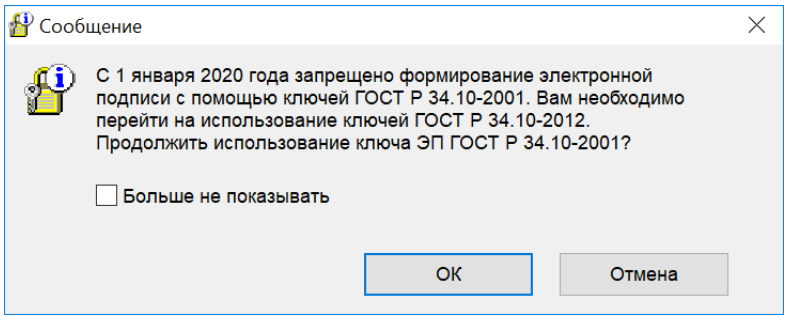


Рисунок 2. Предупреждение о создании подписи с использованием ключа алгоритма ГОСТ Р 34.10-2001 для ОС семейства Windows

Примечание. При использовании операционных систем, отличных от ОС семейства Windows, окно предупреждения может выглядеть иначе, но текстовая составляющая аналогична приведённой на рис. 1 и рис. 2.

При установке флага «Больше не показывать» предупреждения более появляться не будут.

Использование схемы подписи ГОСТ Р 34.10-2001 для создания электронной подписи не допускается.

3.10.2 Компрометация ключей пользователя

К событиям, связанным с компрометацией ключей, относятся, в частности, следующие:

1) потеря ключевых носителей;

2) потеря ключевых носителей с их последующим обнаружением;

3) увольнение сотрудников, имевших доступ к ключевой информации;

4) нарушение правил хранения и уничтожения (после окончания срока действия) закрытого ключа;

5) возникновение подозрений на утечку информации или её искажение в системе конфиденциальной связи;

6) нарушение печати на сейфе с ключевыми носителями;

7) случаи, когда нельзя достоверно установить, что произошло с ключевыми носителями (в том числе случаи, когда ключевой носитель вышел из строя и доказательно не опровергнута возможность того, что данный факт произошел в результате несанкционированных действий злоумышленника).

Различаются два вида компрометации закрытого ключа: явная и неявная. События 1–4 трактуются как явная компрометация ключей, события 5–7 требуют специального рассмотрения в каждом конкретном случае.

При компрометации своего ключа пользователь должен немедленно прекратить связь по сети с другими пользователями. При этом пользователь (или Администратор безопасности) должен немедленно известить ЦР (УЦ) о компрометации ключа пользователя.

По факту компрометации должно быть проведено служебное расследование. Скомпрометированные ключи выводятся из действия. Выведенные из действия скомпрометированные ключевые носители после проведения служебного расследования уничтожаются.

Скомпрометированные ключи подлежат замене. После компрометации ключей пользователь формирует новый закрытый ключ и запрос на сертификат. Так как пользователь не может использовать скомпрометированный ключ для формирования ЭП и передачи запроса в защищённом виде по сети, запрос на сертификат направляется в УЦ в соответствии с Регламентом УЦ.

3.10.3 Уничтожение ключей

Выведенные из действия ключи ЭП/закрытые ключи (в том числе, по истечении срока действия или в случае компрометации ключа) должны быть уничтожены.

Ключи ЭП/закрытые ключи могут быть удалены (уничтожены) следующими способами:

1) функцией «Удалить контейнер» графической утилиты Инструменты КриптоПро (cptools), контрольной панели КриптоПро CSP;

2) с использованием утилиты командной строки wipefile (подробнее см. Приложение 1);

3) с использованием утилиты командной строки csptest;

4) вызовами функций удаления ключа (ключевого контейнера) из прикладного ПО с встроенным СКЗИ (подробнее см. разд. 4 и Приложение 2);

5) путем физического уничтожения ключевого носителя;

6) программными средствами (утилитами) производителей ключевых носителей;

7) иными сертифицированными средствами удаления ключевой информации.

4 Требования к встраиванию СКЗИ в прикладные системы и к проведению исследований СФ СКЗИ

СКЗИ КриптоПро CSP может выступать как в качестве готового к применению средства, так и в качестве платформы для построения на его основе программных, программно-аппаратных и аппаратных решений в области обеспечения информационной безопасности.

Для обеспечения информационной безопасности рабочих мест и информационных систем, использующих СКЗИ, должны быть определены модель возможных угроз и нарушителя и выработана политика безопасности. В зависимости от модели возможных угроз и нарушителя определяется необходимый уровень защиты и, соответственно, необходимый класс СКЗИ.

Возможны следующие варианты применения СКЗИ КриптоПро CSP:

1) применение СКЗИ КриптоПро CSP в составе стандартного программного обеспечения Microsoft, ООО «КРИПТО-ПРО» и других компаний, использующих криптографический интерфейс в соответствии с архитектурой Microsoft (подробнее см. разд. 4.4);

2) встраивание СКЗИ КриптоПро CSP во вновь разрабатываемое или существующее прикладное программное обеспечение, программно-аппаратные и аппаратные решения (подробнее см. ЖТЯИ.00103-03 90 01. КриптоПро CSP. Описание реализации и ЖТЯИ.00103-03 96 01. КриптоПро CSP. Руководство программиста).

При этом указанное программное обеспечение, программно-аппаратные и аппаратные решения по отношению к СКЗИ рассматриваются в качестве среды функционирования (СФ).

Встраивание СКЗИ в СФ проводится в соответствии с порядком, определённым Положением ПКЗ-2005 [2], организациями, имеющими соответствующие лицензии [8] на указанные виды деятельности.

Для указанных вариантов применения СКЗИ в зависимости от условий эксплуатации и от конкретной СФ, в составе которой применяется СКЗИ, могут существовать следующие варианты наличия или отсутствия необходимости проведения исследований СФ:

1) исследования СФ не требуются;

2) требуются исследования по оценке влияния СФ на СКЗИ;

3) требуется проведение тематических исследований СФ со встроенным СКЗИ как самостоятельного шифровального (криптографического) средства[1].

Исследования по оценке влияния СФ на СКЗИ, а также тематические исследования СФ со встроенным СКЗИ как самостоятельного шифровального (криптографического) средства проводятся в соответствии с порядком, определённым Положением ПКЗ-2005 [2], организациями, имеющими соответствующие лицензии [8] на указанные виды деятельности и необходимую область аккредитации.

Перечень компонентов СФ, для которых необходимо проводить исследования, определяется исходя из архитектуры информационной системы и выполняемых функций.

4.1 Применение СКЗИ без исследований СФ

Исследования СФ не требуются при выполнении одного из следующих условий:

1) применение СКЗИ и СФ не подпадает под случаи, приведённые в п. 3 Положения ПКЗ-2005, и иными нормативными документами не определены дополнительные условия применения шифровальных (криптографических) средств;

2) в качестве СФ используются следующие программные компоненты:

- приложения, входящие в состав ОС, под управлением которых может функционировать СКЗИ (см. п. 3.2 Формуляра ЖТЯИ.00103-03 30 01);
- приложение командной строки ОС семейства Windows для формирования запроса на сертификат – certreq;

[1] В соответствии с [8] термины «шифровальное (криптографическое) средство» и «средство криптографической защиты информации» (СКЗИ) являются эквивалентными и включают в себя, в том числе, термин «Средство электронной подписи».

- Microsoft Outlook (из состава Microsoft Office 2007/2010/2013/2016/2019/2021);
- веб-браузер Internet Explorer/Microsoft Edge* (защита TLS соединений);
- сервер IIS (защита TLS соединений);
- сервер nginx версии 1.22.1** (защита TLS соединений);
- сервер Apache версии 2.4.41** (защита TLS соединений);
- сервер Apache версии 2.4.25/2.4.46/2.4.52, используемый в ОС Astra Linux** (защита TLS соединений);
- веб-браузер Chromium GOST, функционирующий в ОС семейств Windows/Linux (защита TLS соединений);
- веб-браузер Яндекс.Браузер, функционирующий в ОС семейств Windows/Linux (защита TLS соединений);
- сервер терминалов и клиент (RDP);
- приложения (модули) из состава СКЗИ (cryptcp, certmgr, stunnel, stunnel_mssp, cpverify), используемые в режиме «ручного вызова»[2];
- иные изделия, в составе которых СКЗИ КриптоПро CSP прошло установленным порядком исследования о оценке влияния или тематические исследования (с учётом правил эксплуатации таких изделий).

Примечание.
* Использование Microsoft Edge без исследований по оценке влияния допускается только в режиме интеграции с Internet Explorer (Internet Explorer mode) (подробнее см. раздел 4.7 документа ЖТЯИ.00103-03 91 02. Руководство администратора безопасности. Windows).
** Установка и эксплуатация должна осуществляться в соответствии с инструкциями документа ЖТЯИ.00103-03 91 03. Руководство администратора безопасности. Linux.

4.2 Применение СКЗИ с проведением исследований по оценке влияния

Проведение исследований по оценке влияния СФ на СКЗИ требуется при одновременном выполнении следующих условий:

- применение СКЗИ и СФ подпадает под случаи, приведённые в п. 3 Положения ПКЗ-2005, или иными нормативными документами определены дополнительные условия применения шифровальных (криптографических) средств;
- в качестве СФ используются программные компоненты из следующего перечня (один пункт или несколько):
  - программные компоненты, использующие приложения (модули) cryptcp, certmgr, stunnel, stunnel_mssp, cpverify, HsmGenArm в режиме «автоматического вызова»[3];
  - программные компоненты, осуществляющие вызовы (напрямую или опосредовано через различные программные интерфейсы) функций СКЗИ, приведённых в Приложении 2 настоящего документа (подробнее об использовании программных интерфейсов СКЗИ см. разд. 4.5).

Оценка влияния СФ СКЗИ на выполнение предъявленных к СКЗИ требований должна выполняться по Техническому заданию, согласованному с 8 Центром ФСБ России.

При реализации (построении) криптографических протоколов с использованием вызовов функций СКЗИ, приведённых в Приложении 2, объём исследований определяется в Техническом задании.

При разработке ИС с использованием функций HsmGenArm необходимо руководствоваться Приложением 1.

4.3 Применение СКЗИ с проведением тематических исследований

Проведение тематических исследований СФ со встроенным СКЗИ как самостоятельного шифровального (криптографического) средства требуется при одновременном выполнении следующих условий:

- применение СКЗИ и СФ подпадает под случаи, приведённые в п. 3 Положения ПКЗ-2005, или иными

[2]Под режимом «ручного вызова» понимается вызов утилиты из командной строки непосредственно пользователем с визуальной обработкой результатов работы утилиты. Ручной вызов приложения HsmGenArm не допускается.
[3]Под режимом «автоматического вызова» понимается вызов утилиты из компонентов прикладной системы, реализованных в исполняемых модулях, скриптах, bat-файлах и т.п.

нормативными документами определены дополнительные условия применения шифровальных (криптографических) средств;

• в качестве СФ используются программные компоненты, осуществляющие вызовы (напрямую или опосредовано через различные программные интерфейсы) функций СКЗИ, не приведённых в Приложении 2 настоящего документа.

4.4 Применение СКЗИ в поддерживаемом[4] программном обеспечении

Программное обеспечение СКЗИ позволяет использовать российские криптографические алгоритмы и сертификаты открытых ключей (ключей проверки ЭП) стандарта X.509 совместно со следующим программным обеспечением (включая, но не ограничиваясь):

• Центр Сертификации — Microsoft Certification Authority, входящий в состав ОС Windows Server 2008 R2/2012/2012 R2/2016/2019/2022

• Электронная почта — Microsoft Outlook (из состава Microsoft Office 2007/2010/2013/2016/2019/2021), Microsoft Outlook Express в составе Internet Explorer/Microsoft Edge, Почта Windows (Windows Mail), Live Mail

• Microsoft Word, Excel из состава Microsoft Office 2007/2010/2013/2016/2019/2021 (с помощью плагина КриптоПро Office Signature)

• Microsoft Exchange Server 2010/2013/2016/2019

• Средства контроля целостности ПО, распространяемого по сети — Microsoft Authenticode

• Службы терминалов для ОС Windows Server 2008 R2/2012/2012 R2/2016/2019/2022 (включая шлюз служб терминалов) и клиент (RDP)

• Internet Explorer/Microsoft Edge и другие браузеры для защиты TCP/IP соединений в сети Интернет (протокол TLS при взаимодействии) — web-сервер IIS, TLS-сервер, TLS-клиент

• Служба доменной аутентификации Winlogon в составе ОС Windows

• Приложение командной строки для формирования запроса на сертификат certreq

• Microsoft SQL Server

• Microsoft ISA Server/Forefront TMG

• Microsoft Forefront UAG

• Служба шифрования данных Microsoft Encrypting File System (EFS) (с помощью КриптоПро EFS)

• Microsoft .NET Framework (с помощью КриптоПро .NET)

• Adobe Reader/Adobe Acrobat (с помощью КриптоПро PDF)

СКЗИ КриптоПро CSP используется в составе и совместно со следующими программными продуктами ООО «КРИПТО-ПРО» (включая, но не ограничиваясь):

• КриптоПро УЦ, Службы УЦ

• КриптоПро Ключ

• КриптоПро HSM

• КриптоПро NGate

• КриптоПро ЭЦП Browser plug-in

• КриптоПро EFS

• КриптоПро Office Signature

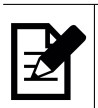
• Secure Pack RUS/КриптоПро SPR

• КриптоПро SSF

• КриптоПро PDF

[4]Под поддержкой понимается техническая совместимость.

- КриптоПро .NET
- КриптоПро Архив
- КриптоПро HLF
- КриптоПро DSS



Примечание. Для указанного ПО в зависимости от условий эксплуатации и от конкретной СФ, в составе которой применяется СКЗИ, могут существовать различные варианты наличия или отсутствия необходимости проведения исследований СФ, изложенные в начале раздела (см. также разд. 1).

4.5 Требования при встраивании СКЗИ в прикладные системы

При разработке пользовательских приложений с использованием функций интерфейса CryptoAPI 1.0 и 2.0 и методов интерфейса cpcurl рекомендуется использовать функции, указанные в Приложении 2 настоящего документа. При этом в части описания параметров функций необходимо руководствоваться документацией Microsoft (для CryptoAPI 1.0 и 2.0) и/или разработчиков libcurl (для cpcurl). В части использования специальных флагов и констант, поддерживаемых СКЗИ «КриптоПро CSP», необходимо руководствоваться разделами документов ЖТЯИ.00103-03 96 01. Руководство программиста (CSP_5_0.chm) с эквивалентными по названию CP функциями[5].

При встраивании СКЗИ КриптоПро CSP в прикладные системы (прикладное ПО, СФ) должны выполняться следующие требования:

1) Сертификаты открытых ключей (ключей проверки ЭП), используемые СКЗИ, следует выпускать Удостоверяющим центром, сертифицированным ФСБ России по классу защиты не ниже класса защиты используемого СКЗИ, с учетом модели угроз и нарушителя информационной системы, в которой применяется СКЗИ, а также с учетом нормативных документов, определяющих создание такой информационной системы.

2) При использовании функций СКЗИ должна быть обеспечена актуальность сертификатов открытых ключей/сертификатов ключей проверки ЭП (с использованием УЦ, построенного на базе Средств УЦ, сертифицированных ФСБ России и правил обработки объектов PKI в СФ).

Актуальность сертификатов открытых ключей/сертификатов ключей проверки ЭП (далее — сертификатов) должна обеспечиваться:

- проверкой корректности ЭП сертификатов из цепочки сертификатов;
- проверкой сроков действия сертификатов из цепочки сертификатов;
- проверкой области использования сертификата;
- проверкой сертификатов из цепочки сертификатов на отозванность при помощи списков отозванных (аннулированных) сертификатов (CRL) или с помощью OCSP-службы;
- проверкой сроков действия CRL (ответа OCSP-службы);
- проверкой корректности ЭП CRL (ЭП ответа OCSP-службы);
- проверкой наличия CRL;
- проверкой соответствия сертификата, указанного в полученном ответе OCSP-службы, сертификату, который был указан в соответствующем запросе к OCSP-службе;
- проверкой актуальности сертификата, используемого для проверки ЭП под CRL (сертификата OCSP-службы); в рамках данной проверки выполняются все выше перечисленные проверки;
- при использовании метки времени должны быть проведены аналогичные проверки для сертификата TSP-сервера, а также проверка ЭП ответа TSP-сервера и проверка соответствия хэш-значения ЭП, содержащегося в ответе TSP-сервера, хэш-значению ЭП, содержащемуся в соответствующем запросе TSP-серверу.

3) При вызове функций СКЗИ требуется указывать имя и тип провайдера из состава СКЗИ «КриптоПро CSP»; в случае использования провайдера «по умолчанию» (NULL) необходимо предусмотреть невозможность использования провайдеров, не входящих в состав СКЗИ. Для функций CryptoAPI допускается использование только провайдеров из состава СКЗИ «КриптоПро CSP» (типы 75, 80 и 81).

4) При вызове функций СКЗИ в прикладном программном обеспечении должна быть предусмотрена проверка кода завершения вызываемой функции с обработкой ошибочных ситуаций и регистрацией событий.

[5] В зависимости от исполнения УЦ ряд функций может быть недоступен.

5) При необходимости обеспечения аутентификации отправителя зашифрованного сообщения (подписанного и зашифрованного сообщения), а также имитозащиты, должны применяться дополнительные механизмы аутентификации/имитозащиты. Указанная необходимость определяется в ТЗ.

6) В случае реализации расшифрования CMS-сообщения типа Enveloped Data в поточном режиме (например, вызовом CryptMsgUpdate с указанием контекста, полученного с помощью CryptMsgOpenToDecode с параметром pStreamInfo $\neq$ NULL) использование выходных (расшифрованных) данных допускается только после успешного завершения расшифрования в поточном режиме. Если по завершении расшифрования в поточном режиме (например, при вызове CryptMsgUpdate с указанием флага fFinal = TRUE) была зафиксирована ошибка, требуется выполнить гарантированное удаление результата расшифрования. Доступом к результату расшифрования должен обладать только владелец сертификата ключа, указанного в сообщении.

7) При использовании КриптоПро ЭЦП Browser plug-in должны выполняться следующие меры:

- Передача JavaScript кода должна осуществляться по защищенному каналу, в противном случае требуется подтвердить целостность полученного с сервера кода.
- Необходимо обеспечить невозможность модификации JavaScript кода на стороне клиента (в т.ч. блокировку отладочных средств и инструментов разработчика в браузерах, запрет на использование других веб-браузеров без блокировки).
- Обновление браузеров должно сопровождаться проверкой блокировки отладочных средств и инструментов разработчика. Настройка блокировки должна быть доступна только администратору системы. Не допускается проведение обновления браузеров в автоматическом режиме.
- В перечень модулей среды функционирования, подлежащих контролю целостности в соответствии с требованиями к эксплуатации СКЗИ, должен быть добавлен встроенный в веб-браузер интерпретатор.

8) Корневые сертификаты должны быть получены по доверенному каналу[6]. Размещение в хранилище корневых сертификатов должно проводиться Администратором безопасности с контролем необходимости наличия конкретных корневых сертификатов, используемых для построения цепочек сертификатов.

9) В эксплуатационной документации на СФ должна быть предусмотрена необходимость проведения проверки ПО BIOS ПЭВМ, на которых предполагается функционирование СКЗИ и СФ, на соответствие методическим документам ФСБ России в области исследований программного обеспечения BIOS, либо проведение указанной проверки для конкретной конфигурации аппаратной платформы.

10) В эксплуатационной документации в зависимости от конкретных условий эксплуатации информационной системы и прикладного ПО должны быть определены условия применения МДЗ и Secure Pack Rus версии 3.0/КриптоПро SPR 4.0.

11) При встраивании СКЗИ в компоненты СФ, в которых функции создания и/или проверки электронной подписи не являются автоматическими, необходимо проводить оценку соответствия компонентов СФ п.п. 8 и/или 9 «Требований к средствам электронной подписи», утвержденных Приказом ФСБ России от 27 декабря 2011 г. № 796 «Об утверждении Требований к средствам электронной подписи и Требований к средствам удостоверяющего центра».

12) В случае опосредованного использования в программном обеспечении вызовов функций СКЗИ техническое задание на проведение работ по оценке влияния такого программного обеспечения на СКЗИ необходимо согласовывать, в том числе, с ООО «КРИПТО-ПРО».

При проведении исследований по оценке влияния должны быть предусмотрены следующие направления исследований (включая, но не ограничиваясь):

- по исходным кодам и эксплуатационной документации на СФ осуществляется анализ выполнения требований и рекомендаций по встраиванию СКЗИ, изложенных в документации на СКЗИ, а также анализ полноты и корректности эксплуатационной документации на компоненты СФ, включая отражение в указанной документации организационно-технических мер по защите информации с использованием СКЗИ и правил обращения с ключевыми документами;
- проверка корректности реализации в СФ работы с криптографическими ключами;
- проверка корректности реализации механизмов контроля целостности СФ и СКЗИ;
- проверка корректности работы криптографических функций в штатном и не штатном режимах и в условиях искаженных и/или отсутствующих объектов PKI;

[6]Под доверенным каналом понимается способ передачи, исключающий возможность подмены корневого сертификата (личная передача, криптографически защищённый канал связи и т.п.).

• проверка механизмов визуализации информации для пользователя (п.п. 8 и/или 9 Приложения 1 к Приказу ФСБ России от 27 декабря 2011 г. № 796 «Об утверждении Требований к средствам электронной подписи и Требований к средствам удостоверяющего центра»);

• в случае опосредованного использования функций программного интерфейса CryptoAPI 2.0, методов интерфейса cpcurl или утилиты HsmGenArm из других программных интерфейсов, необходимо проведение исследований прохождения вызовов и передаваемых данных между вызываемым интерфейсом и интерфейсами CryptoAPI 2.0/cpcurl/утилитой HsmGenArm (построение трасс вызовов).

5 Требования по обеспечению безопасности при эксплуатации СКЗИ

5.1 Общие требования по организации работ по защите от НСД

Защита аппаратного и программного обеспечения от НСД при установке и использовании СКЗИ КриптоПро CSP является составной частью общей задачи обеспечения безопасности информации в системе, в состав которой входит СКЗИ.

При организации работ по защите информации от НСД необходимо выполнение приведённых ниже организационно-технических и административных мер по обеспечению правильного функционирования средств обработки и передачи информации, а также установление соответствующих правил для обслуживающего персонала, допущенного к работе с СКЗИ и защищаемой информацией.

Защита информации от НСД должна обеспечиваться на всех технологических этапах обработки информации и во всех режимах функционирования, в том числе при проведении ремонтных и регламентных работ.

В организации, эксплуатирующей СКЗИ, должен быть назначен **Администратор безопасности**, на которого возлагаются задачи по установке и настройке программно-аппаратных средств, общесистемного и специального ПО, эксплуатируемого совместно с СКЗИ, по установке и первичной настройке СКЗИ, по организации работ по использованию СКЗИ, выработке соответствующих инструкций для пользователей, а также контроля за соблюдением требований по безопасности.

Допускается назначать группу лиц (например, Администратора безопасности СКЗИ и Системного администратора ИС), распределяя между ними приведённые в настоящей эксплуатационной документации обязанности Администратора безопасности с учётом модели угроз и нарушителя ИС, в которой применяется СКЗИ, а также с учетом нормативных документов, определяющих создание такой ИС. В этом случае порядок распределения обязанностей, функций и полномочий между ответственными лицами должен быть зафиксирован в эксплуатационной документации ИС.

Администратор безопасности не должен иметь возможность доступа к конфиденциальной информации пользователей.

Правом доступа к рабочим местам с установленными СКЗИ должны обладать только определённые для эксплуатации лица, прошедшие соответствующую подготовку. Администратор безопасности должен ознакомить каждого пользователя, применяющего СКЗИ, с документацией на СКЗИ, а также с другими нормативными документами, созданными на её основе.

5.2 Требования по размещению технических средств с установленным СКЗИ

При размещении технических средств с установленным СКЗИ:

• Должны быть приняты меры по исключению несанкционированного доступа к ПЭВМ/устройствам, на которых установлены СКЗИ, посторонних лиц, по роду своей деятельности не являющихся персоналом, допущенным к работе с указанными ПЭВМ/устройствами. В случае такой необходимости должен быть обеспечен контроль за их действиями и обеспечена невозможность негативных действий с их стороны на СКЗИ, ПЭВМ/устройства, на которых эксплуатируется СКЗИ, и защищаемую информацию.

• Должны быть приняты меры, исключающие несанкционированное вскрытие корпусов ПЭВМ, устройств и средств, входящих в состав СКЗИ.

• Внутренняя планировка, расположение и укомплектованность рабочих мест в помещениях, в которых расположены ПЭВМ с установленным СКЗИ, должны обеспечивать исполнителям работ, сохранность доверенных им конфиденциальных документов и сведений, включая ключевую информацию.

При этом, с учетом требований, предъявляемых к информационной системе, в которой эксплуатируется СКЗИ, а также с учетом специфики и условий работы конкретных пользователей СКЗИ, требования к эксплуатации СКЗИ в спецпомещениях могут не предъявляться.

• Размещение СКЗИ в помещениях, в которых осуществляется обработка информации, содержащей сведения, составляющие государственную тайну, осуществляется установленным порядком.

При обработке с помощью СКЗИ конфиденциальной информации, передаваемой по каналам связи, выходящим за пределы контролируемой территории, необходимо:

1) Для проводных каналов (электрических и оптических) использовать любое из следующих устройств:

• Волоконно-оптические линии связи;

• Оптические развязывающие устройства, устанавливаемые в тракт передачи информации для создания оптоволоконного фрагмента сети;

• Сертифицированные СКЗИ для передачи информации соответствующего уровня конфиденциальности.

2) Для радиоканалов:

Использовать радиоканал GSM, либо GPRS, либо 3G/4G, либо Wi-Fi, либо другой канал мобильной и беспроводной связи, работающий с цифровой модуляцией штатного информационного сигнала.

5.3 Требования по установке СКЗИ, общесистемного и специального ПО на ПЭВМ

1) Инсталляция СКЗИ КриптоПро CSP на рабочих местах должна производиться только с дистрибутива, полученного по доверенному каналу.

2) К установке общесистемного и специального программного обеспечения, а также СКЗИ КриптоПро CSP, допускаются лица, прошедшие соответствующую подготовку и изучившие документацию на соответствующее ПО и на СКЗИ.

3) На технических средствах, предназначенных для работы с СКЗИ, допустимо использовать только лицензионное программное обеспечение фирм-изготовителей.

4) Из состава системы должно быть исключено оборудование, которое может создавать угрозу безопасности ОС. Также необходимо избегать использования нестандартных аппаратных средств, имеющих возможность влиять на функционирование ПЭВМ или ОС.

5) До загрузки ОС должен быть реализован контроль целостности файлов, критичных для загрузки ОС, и утилиты cpverify с использованием модуля доверенной загрузки.

6) После загрузки ОС должен быть реализован контроль целостности программного обеспечения, входящего в состав СКЗИ, самой ОС и всех исполняемых файлов, функционирующих совместно с СКЗИ, с использованием утилиты cpverify.

7) При установке ПО СКЗИ на ПЭВМ должен быть обеспечен контроль целостности и достоверность дистрибутива СКЗИ и совместно поставляемых с СКЗИ компонент СФ (см. Приложение 1).

8) На ПЭВМ/мобильном устройстве не должны устанавливаться средства разработки ПО, отладчики и трассировщики. Если средства отладки приложений и/или трассировки вызовов нужны для технологических потребностей организации, то их использование должно быть санкционировано Администратором безопасности. При этом должны быть реализованы меры, исключающие возможность использования этих средств для редактирования кода и памяти СКЗИ и приложений, использующих СКЗИ, а также для просмотра кода и памяти СКЗИ и приложений, использующих СКЗИ, в процессе обработки СКЗИ защищаемой информации и/или при загруженной ключевой информации.

9) При установке программного обеспечения СКЗИ КриптоПро CSP следует:

• Предусмотреть меры, исключающие возможность несанкционированного не обнаруживаемого изменения аппаратной части технических средств, на которых установлены СКЗИ (например, путём опечатывания системного блока и разъемов ПЭВМ).

• После завершения процесса установки должны быть выполнены действия, необходимые для осуществления периодического контроля целостности установленного ПО СКЗИ, а также его окружения с помощью утилиты cpverify в соответствии с документацией (см. Приложение 1).

• Программное обеспечение, устанавливаемое на ПЭВМ с СКЗИ, не должно содержать возможностей, позволяющих:

– модифицировать содержимое произвольных областей памяти;

– модифицировать собственный код и код других подпрограмм;
– модифицировать память, выделенную для других подпрограмм;
– передавать управление в область собственных данных и данных других подпрограмм;
– несанкционированно модифицировать файлы, содержащие исполняемые коды при их хранении на жестком диске;
– повышать предоставленные привилегии;
– модифицировать настройки ОС;
– использовать недокументированные фирмой-разработчиком функции ОС.

5.4 Требования по обеспечению защиты от НСД

При использовании СКЗИ должны выполняться следующие меры по защите информации от НСД:

• необходимо разработать и применить политику назначения и смены паролей (для входа в ОС, BIOS, при шифровании на пароле и т.д.).

• необходимо использовать фильтры паролей в соответствии со следующими правилами:

– длина пароля должна быть не менее 8 символов;
– в числе символов пароля обязательно должны присутствовать буквы в верхнем и нижнем регистрах, цифры и специальные символы (@, #, $, &, *, % и т.п.);
– пароль не должен включать в себя легко вычисляемые сочетания символов (имена, фамилии и т. д.), а также общепринятые сокращения (USER, ADMIN, ALEX и т. д.);
– при смене пароля новое значение должно отличаться от предыдущего не менее чем в 4-х позициях;
– личный пароль пользователь не имеет права сообщать никому;
– периодичность смены пароля определяется принятой политикой безопасности, но не должна превышать 6 месяцев.

• пароли для аутентификации пользователей на носителях, работающих в режиме активного вычислителя с защитой канала между носителем и СКЗИ по протоколу SESPAKE, должны удовлетворять следующим требованиям:

– для выработки общего ключа с аутентификацией на основе пароля при использовании кривых с 512-битовым порядком группы точек:
  * пароль должен состоять из букв английского алфавита в верхнем и нижнем регистрах, цифр и спецзнаков;
  * минимальная длина пароля — 4 символа;
  * периодичность смены пароля — не реже 1 раза в полгода.

– для выработки общего ключа с аутентификацией на основе пароля при использовании кривых с 256-битовым порядком группы точек:
  * пароль должен состоять из букв английского алфавита в верхнем и нижнем регистрах, цифр и спецзнаков;
  * минимальная длина пароля — 12 символов;
  * периодичность смены пароля — не реже 1 раза в 147 дней.

ИЛИ

  * пароль должен состоять из букв английского алфавита в верхнем и нижнем регистрах, цифр и спецзнаков;
  * минимальная длина пароля — 11 символов;
  * периодичность смены пароля — не реже 1 раза в 15 дней.

• политика назначения и смены паролей должна удовлетворять следующим требованиям: после 10 неверных попыток ввода пароля пользователем при входе в ОС система должна блокироваться на 1 час (с обеспечением возможности разблокировки учетной записи при обращении пользователя к Администратору безопасности).

• указанная политика обязательна для всех учётных записей, зарегистрированных в ОС.

• пароль для входа в BIOS должен быть известен только Администратору безопасности и быть отличным от пароля для входа в ОС.

• в дополнение к паролю должны использоваться механизмы аутентификации, предоставляемые используемым средством защиты от НСД, или аутентификация по сертификату.

• сохранение паролей в системе допускается только при условии однопользовательского применения ПЭВМ, в противном случае необходимо запретить сохранение паролей, используемых в работе СКЗИ, подробнее см. разд. 3.9.2.

• политика паролей ключевых контейнеров может быть настроена с помощью механизмов СКЗИ (подробнее см. разд. 3.9.3).

При эксплуатации СКЗИ **запрещено**:

• оставлять без контроля вычислительные средства, на которых эксплуатируется СКЗИ (в том числе ключевые носители), после ввода ключевой информации либо иной конфиденциальной информации (при оставлении ПЭВМ/мобильного устройства без контроля должна включаться парольная заставка);

• вносить какие-либо изменения в программное обеспечение СКЗИ;

• осуществлять несанкционированное Администратором безопасности копирование ключевых носителей;

• разглашать содержимое носителей ключевой информации или передавать сами носители лицам, к ним не допущенным, выводить ключевую информацию на дисплей, принтер и иные средства отображения информации (за исключением случаев, предусмотренных эксплуатационной документацией на СКЗИ);

• использовать ключевые носители в режимах, не предусмотренных функционированием СКЗИ;

• вставлять ключевой носитель в устройство считывания в режимах, не предусмотренных штатным режимом использования ключевого носителя;

• записывать на ключевые носители постороннюю информацию;

• использовать ключи с истекшим сроком действия;

• подключать к ПЭВМ дополнительные устройства и соединители, не предусмотренные штатной комплектацией;

• работать на ПЭВМ, если во время его начальной загрузки не проходит встроенный тест ОЗУ, предусмотренный в ПЭВМ;

• изменять настройки, установленные программой установки СКЗИ или Администратором безопасности;

• использовать синхропосылки, вырабатываемые не средствами СКЗИ;

• использовать бывшие в работе ключевые носители для записи новой информации без предварительного уничтожения на них ключевой информации, подлежащей уничтожению, средствами СКЗИ (в соответствии с разд. 3.10.3);

• осуществлять несанкционированное вскрытие системных блоков ПЭВМ.

Администратор безопасности должен сконфигурировать операционную систему, в среде которой планируется использовать СКЗИ, и осуществлять периодический контроль сделанных настроек в соответствии со следующими требованиями:

• в системе регистрируется один пользователь, обладающий правами администратора, на которого возлагается обязанность конфигурировать ОС, настраивать безопасность ОС, а также конфигурировать ПЭВМ/мобильное устройство, на которую установлена ОС;

• не допускается использовать нестандартные, измененные или отладочные версии ОС;

• необходимо исключить возможность загрузки и использования ОС, отличной от предусмотренной штатной работой (например, с помощью настроек BIOS, использования МДЗ и т.п.);

• необходимо исключить возможность удаленного управления, администрирования и модификации ОС и её настроек с использованием незащищённого канала связи; для защиты канала связи необходимо использовать средства, сертифицированные ФСБ России по классу не ниже класса используемого СКЗИ;

• на ПЭВМ должна быть установлена только одна операционная система;

• правом установки и настройки ОС и СКЗИ должен обладать только Администратор безопасности;

• все неиспользуемые ресурсы системы необходимо удалить или отключить (протоколы, сервисы, порты, общие ресурсы, пользователи и т.п.);

• режимы безопасности, реализованные в ОС, должны быть настроены на максимальный уровень;

• всем пользователям и группам, зарегистрированным в ОС, Администратор безопасности в соответствии с политикой безопасности, принятой в организации, даёт минимально возможные для нормальной работы права; каждый пользователь ОС, не являющийся администратором, может просматривать и редактировать только свои настройки в рамках прав доступа, назначенных ему администратором;

• необходимо предусмотреть меры, максимально ограничивающие доступ к следующим ресурсам системы (в соответствующих условиях возможно полное удаление ресурса или его неиспользуемой части):

- системный реестр;
- файлы и каталоги;
- временные файлы;
- журналы системы;
- файлы подкачки;
- кэшируемая информация (пароли и т.п.);
- отладочная информация.

• кроме того, необходимо организовать стирание (по окончании сеанса работы СКЗИ) временных файлов и файлов подкачки, формируемых или модифицируемых в процессе работы СКЗИ; если это не выполнимо, то на жёсткий диск должны распространяться требования, предъявляемые к ключевым носителям;

• должна быть исключена возможность создания аварийного дампа оперативной памяти, так как он может содержать криптографически опасную информацию;

• должно быть исключено попадание в систему программ, позволяющих повышать предоставленные привилегии за счёт ошибок в ОС;

• средствами BIOS должна быть исключена возможность работы на ПЭВМ с СКЗИ, если во время её начальной загрузки не проходят встроенные тесты (POST);

• средствами BIOS должна быть исключена возможность отключения пользователями ISA-устройств и PCI-устройств;

• после инсталляции ОС следует установить все рекомендованные программные обновления и программные обновления, связанные с безопасностью, существующие на момент инсталляции ОС;

• необходимо регулярно устанавливать пакеты обновления безопасности ОС (Service Packs, Hot fix и т.п.), обновлять антивирусные базы, а также исследовать информационные ресурсы по вопросам компьютерной безопасности с целью своевременной минимизации опасных последствий от возможного воздействия на ОС;

• в случае подключения ПЭВМ с установленным СКЗИ к общедоступным сетям передачи данных, необходимо исключить возможность открытия и исполнения файлов и скриптовых объектов (например, JavaScript, VBScript, ActiveX), полученных из общедоступных сетей передачи данных без проведения соответствующих проверок на предмет содержания в них программных закладок и вредоносного ПО, загружаемых из сети;

• при использовании СКЗИ на ПЭВМ/устройствах, подключённых к общедоступным сетям связи, с целью исключения возможности несанкционированного доступа к системным ресурсам используемых операционных систем, к программному обеспечению, в окружении которого функционируют СКЗИ, и к компонентам СКЗИ со стороны указанных сетей должны использоваться дополнительные методы и средства защиты (например, установка межсетевых экранов, IPS/IDS, организация VPN сетей и т.п.). При этом предпочтение должно отдаваться средствам защиты, имеющим сертификат уполномоченного органа по сертификации;

• организовать и использовать систему аудита, организовать регулярный анализ результатов аудита;

• в настройках ОС, отвечающих за ведение журналов событий, необходимо установить режим архивирования журнала при его заполнении;

• организовать и использовать комплекс мероприятий антивирусной защиты;

• должны быть отключены радиоканалы, неиспользуемые в работе СКЗИ.

Аппаратуру, на которой устанавливается СКЗИ, рекомендуется проверить на отсутствие аппаратных закладок.

5.5 Требования по обеспечению физической безопасности сервера

Следует исключить возможность доступа неавторизованного персонала к консоли, системе питания и дополнительным устройствам, подключенным к защищаемому серверу путём установки оборудования в специально выделенное и запираемое помещение (аппаратную или серверную комнату).

Доступ персонала в серверную комнату должен быть регламентирован внутренним распорядком эксплуатирующей организации и должностными инструкциями.

Для исключения сбоев компьютера, вызванных отключением электропитания, необходимо обеспечить электропитание сервера от источника бесперебойного питания достаточной мощности. Как минимум, мощности батарей источника бесперебойного питания должно хватать на время достаточное для корректного автоматического завершения работы сервера.

5.6 Требования по организации процедуры резервного копирования и хранения резервных копий ключевых носителей

При определении регламента резервного копирования и хранения резервных копий ключевых носителей следует обеспечить ответственное хранение резервных копий в запираемых сейфах (шкафах) и определить процедуру выдачи резервных копий ответственному персоналу и уничтожения вышедших из употребления носителей.

Стандартными мерами по организации ответственного хранения ключевых носителей являются:

- маркировка носителей;
- составление описи хранимых носителей с указанием серийных (инвентарных) номеров, дат записи носителей, фамилией сотрудника, создавшего копию для каждого сейфа (шкафа);
- периодическая сверка описи и содержимого сейфов (шкафов);
- организация ответственного хранения и выдачи ключей от сейфов (шкафов);
- возможное опечатывание (опломбирование) сейфов (шкафов);
- уничтожение вышедших из употребления ключевых носителей производится комиссией с составлением акта об уничтожении.

Период проведения резервного копирования определяется с учётом условий эксплуатации СКЗИ в конкретной информационной системе.

5.7 Требования по подключению СКЗИ для работы по общедоступным каналам передачи данных

Порядок подключения СКЗИ к каналам связи должен быть определён эксплуатирующей организацией. Лицом, ответственным за безопасность работы СКЗИ по общедоступным каналам, как правило, должен быть Администратор безопасности.

При подключении СКЗИ к общедоступным каналам передачи данных должна быть обеспечена безопасность защищённой связи. При этом должны быть определены:

- порядок подключения СКЗИ к каналам;
- лицо, ответственное за безопасность работы по общедоступным каналам;
- политика безопасности защищённой связи;
- допустимый состав прикладных программных средств, для которого должно быть исследовано и обосновано отсутствие негативного влияния на СКЗИ по каналу передачи данных;
- перечень допустимых сетевых протоколов;
- средства защиты сетевых соединений;
- система и средства антивирусной защиты.

Перечень стандартных средств ОС определяется на уровне эксплуатирующей организации без проведения дополнительных исследований по оценке их влияния на СКЗИ. При этом должны выполняться:

- своевременное обновление используемых программных средств защиты;
- контроль среды функционирования СКЗИ;

- определение и контроль за использованием сетевых протоколов;
- соблюдение правил пользования СКЗИ и средой функционирования СКЗИ.

Должен быть обеспечен организационно-технический контроль запросов на установление соединения абонентов по протоколу TLS с использованием эфемерных ключей, исключающих возможность использования абонентом не своих атрибутов соединения (такие, как Client_Id и т.п.).

При использовании СКЗИ с другими стандартными программными средствами возможность подключения СКЗИ к общедоступным каналам передачи данных должна быть определена только после проведения дополнительных исследований с оценкой невозможности негативного влияния нарушителя на функционирование СКЗИ, использующего средства общедоступных каналов.

При установке параметров, позволяющих создавать соединения, отличные от криптографически защищённых (ГОСТ TLS и IPsec на основе сертификатов ключей ГОСТ Р 34.10-2012 (ГОСТ 34.10-2018)) соединений, должны быть приняты меры, исключающие утечку требующей защиты информации с защищаемого объекта информации. Проверка достаточности принятых мер защиты проводится при аттестации объекта информатизации с СКЗИ по требованиям информационной безопасности.

5.8 Требования по использованию модулей (средств) доверенной загрузки

Примечание. СКЗИ «КриптоПро CSP» версия 5.0 R3 КС3 (исполнение 3-Base) должно использоваться с МДЗ. Необходимо использовать изделие, сертифицированное ФСБ России. Поставка осуществляется по согласованию с пользователем.

Модули доверенной загрузки (МДЗ) предназначены для организации защиты компьютера от входа посторонних пользователей. Под посторонними пользователями понимаются все лица, не зарегистрированные в системе как пользователи данного компьютера.

МДЗ обеспечивают:

- регистрацию пользователей компьютера и назначение им персональных идентификаторов и паролей на вход в систему;
- идентификацию и аутентификацию пользователя при загрузке компьютера;
- ведение системного журнала, в котором производится регистрация событий, имеющих отношение к безопасности системы;
- контроль целостности программного и аппаратного обеспечения защищаемого компьютера;
- защиту от несанкционированной загрузки операционной системы со съемных носителей;
- наличие в составе аппаратного ДСЧ (опционально).

Установка, настройка и эксплуатация МДЗ должна производиться в соответствии с эксплуатационной документацией на МДЗ. Перед началом эксплуатации МДЗ необходимо ознакомиться с комплектом документации на используемое средство и принять рекомендуемые в документации защитные организационные меры.

Настройка МДЗ на требуемую конфигурацию выполняется Администратором безопасности. Настройка должна исключать возможность вмешательства пользователя в процессы загрузки операционной системы и прикладного ПО и проверки целостности программной среды.

В случае использования средств защиты от НСД также необходимо предусматривать контроль эффективности этих средств. Контроль эффективности должен периодически выполняться Администратором безопасности на основе требований документации на используемые средства защиты от НСД.

6 Требования по криптографической защите

Должны выполняться следующие требования по криптографической защите:

1) Настройки операционных систем для работы с СКЗИ должны производиться в соответствии с документами:

- ЖТЯИ.00103-03 91 02 Руководство администратора безопасности. Windows
- ЖТЯИ.00103-03 91 03 Руководство администратора безопасности. Linux

2) При установке ПО СКЗИ должен быть обеспечен контроль целостности и достоверность дистрибутива СКЗИ и совместно поставляемых с СКЗИ компонент СФ. После завершения процесса установки должны быть выполнены действия, необходимые для осуществления периодического контроля целостности установленного ПО СКЗИ, а также его окружения.

3) Контролем целостности должны быть охвачены файлы, указанные в разделе «Требования по криптографической защите» Руководств администратора безопасности для соответствующих используемых ОС.

4) Если проверка целостности компонентов СКЗИ завершается ошибкой, Администратор безопасности должен выявить причину и обстоятельства нарушения целостности СКЗИ и переустановить СКЗИ в соответствии с инструкцией по установке, описанной в Руководстве администратора безопасности для используемой программно-аппаратной платформы.

5) Периодичность тестового контроля криптографических функций — 10 минут.

6) Периодичность перезагрузки ПЭВМ — 7 дней.

7) Периодичность контроля системы охлаждения процессорного блока ПЭВМ — 1 месяц.

8) Запрещается использовать режим простой замены (ECB) ГОСТ 28147-89, ГОСТ Р 34.13-2015 (ГОСТ 34.13-2018) для шифрования информации, кроме ключевой.

9) Запрещается использование беспроводных клавиатур и компьютерных мышей.

10) При использовании функций шифрования и имитозащиты рекомендуется применять алгоритмы ГОСТ Р 34.12-2015 (ГОСТ 34.12-2018) и ГОСТ Р 34.13-2015 (ГОСТ 34.13-2018). При использовании функции хэширования рекомендуется использовать ГОСТ Р 34.11-2012. В процедурах формирования ЭП необходимо использовать алгоритм хэширования ГОСТ Р 34.11-2012.

11) При использовании СКЗИ для защиты TLS-соединений в роли TLS-сервера под управлением ОС Windows после установки необходимо выполнить следующие действия по настройке используемых криптографических алгоритмов:

1. Проверить, что отключено использование устаревших криптонаборов (cipher suite-ов), посмотрев либо в контрольной панели СКЗИ КриптоПро CSP (вкладка «Настройки TLS», секция «Сервер»), либо значение ключа `tls_server_disable_legacy_cipher_suites` в ветке реестра `HKEY_LOCAL_MACHINE\SOFTWARE\[Wow6432Node]\Crypto Pro\Cryptography\CurrentVersion\Parameters\` (должно быть равно 1).
2. Выполнить перезагрузку компьютера.

При использовании СКЗИ для защиты TLS-соединений в роли TLS-сервера под управлением ОС Linux/Unix после установки необходимо выполнить следующие действия по настройке используемых криптографических алгоритмов:

1. Отключить использование устаревших криптонаборов, установив значение параметра `tls_server_disable_legacy_cipher_suites` равным 1 командой

```
./cpconfig -ini '\config\Parameters' -add long tls_server_disable_legacy_cipher_suites 1
```

2. Выполнить перезагрузку сервиса `cprocsp` командой

```
systemctl restart cprocsp
```

12) При использовании СКЗИ для защиты IP-соединений по протоколам IPsec под управлением ОС Windows после установки «КриптоПро IPsec» необходимо выполнить следующие действия по настройке используемых криптографических алгоритмов:

1. В настройках параметров «IKE хэш» задать использование `CPIKE_GOST_R_34_11_2012_512`.
2. В настройках параметров IKE группы и IKE группы (PFS) выбрать использование `CPIKE_VKO_GOST_R_34_10_2012_256A`, `CPIKE_VKO_GOST_R_34_10_2012_256B`, `CPIKE_VKO_GOST_R_34_10_2012_512A` или `CPIKE_VKO_GOST_R_34_10_2012_512B`.
3. В настройках параметров ESP Combined Mode задать отсутствие отправки «Authentication Algorithm (5)».

4. В групповой политике IP-безопасности для IP-фильтра защищаемого трафика выбрать метод безопасности «Шифрование и обеспечение целостности».

7 Разбор конфликтных ситуаций, связанных с применением ЭП

Применение электронной подписи в автоматизированной системе может приводить к конфликтным ситуациям, заключающимся в оспаривании сторонами (участниками системы) авторства и/или содержимого документа, подписанного электронной подписью.

Разбор подобных конфликтных ситуаций требует применения специального программного обеспечения для выполнения проверок и документирования данных, используемых при выполнении процедуры проверки соответствия ЭП содержимому электронного документа.

Разбор конфликтной ситуации заключается в доказательстве авторства подписи конкретного электронного документа конкретным исполнителем.

Данный разбор основывается на математических свойствах алгоритмов ЭП, реализованных в соответствии со стандартами ГОСТ Р 34.10-2001, ГОСТ Р 34.10-2012 (ГОСТ 34.10-2018), ГОСТ Р 34.11-2012 (ГОСТ 34.11-2018) и ГОСТ Р 34.11-94, гарантирующих невозможность подделки значения ЭП любым лицом, не обладающим ключом ЭП.

При проверке значения ЭП используется ключ проверки ЭП, значение которого вычисляется по значению ключа ЭП при их формировании.

В системе должны быть предусмотрены средства ведения архивов электронных документов с ЭП и сертификатов ключей проверки ЭП.

Разбор конфликтной ситуации выполняется комиссией, состоящей из представителей сторон, службы безопасности и экспертов. Состав комиссии, порядок её формирования, регламент работы, рассмотрение результатов определяется в приложении к Регламенту (Договору), заключаемому между участниками информационного обмена.

Оспаривание результатов работы комиссии и возмещение пострадавшей стороне принесённого ущерба выполняется в установленном действующим законодательством Российской Федерации порядке.

7.1 Порядок разбора конфликтной ситуации

Разбор конфликтной ситуации выполняется по инициативе любого участника информационного обмена и состоит из:

1) предъявления претензии одной стороны другой;
2) формирования комиссии;
3) разбора конфликтной ситуации;
4) принятие мер по урегулированию конфликта.

Разбор конфликтной ситуации проводится с использованием программного обеспечения СКЗИ КриптоПро CSP для электронного документа, авторство или содержание которого оспаривается.

Проверка подписанного электронного документа включает в себя выполнение следующих действий:

1) определение сертификата или нескольких сертификатов, необходимых для проверки ЭП;
2) проверка ЭП электронного документа с использованием каждого сертификата;
3) определение даты создания каждой ЭП в электронном документе;
4) проверка ЭП каждого сертификата, путём построения цепочки сертификатов до сертификата Главного ЦС;
5) проверка действительности сертификатов на текущий момент времени;
6) проверка действительности сертификатов на момент создания ЭП;
7) проверка отсутствия сертификатов в CRL.

При проверке ЭП документа, верификации цепочки сертификатов, отсутствии сертификата в CRL, авторство подписи под документом считается установленным.

Примечание. Несовпадение даты формирования документа и сроков действия сертификата и/или сроков действия ключа ЭП не влияют на определение авторства документа. В таком случае можно сделать предположение о несоблюдении пользователем Регламента (Договора) в части сроков действия ключей, сертификатов или некорректного использования сертификата в прикладном ПО.

7.2 Случаи невозможности проверки значения ЭП

При отсутствии в архиве сертификата открытого ключа (ключа проверки ЭП) пользователя, выполнившего ЭП, доказать авторство документа невозможно. В связи с этим, архив с сертификатами открытых ключей необходимо подвергать регулярному резервному копированию и хранить в течение всего установленного срока хранения.

Литература

[1] Приказ ФАПСИ от 13.06.2001 N 152 «Об утверждении Инструкции об организации и обеспечении безопасности хранения, обработки и передачи по каналам связи с использованием средств криптографической защиты информации с ограниченным доступом, не содержащей сведений, составляющих государственную тайну». URL: http://base.garant.ru/183628/.

[2] Приказ ФСБ России от 09.02.2005 № 66 «Об утверждении Положения о разработке, производстве, реализации и эксплуатации шифровальных (криптографических) средств защиты информации (Положение ПКЗ-2005)». URL: http://base.garant.ru/187947/.

[3] Требования к средствам криптографической защиты информации, предназначенным для защиты информации, не содержащей сведений, составляющих государственную тайну.

[4] Требования к средствам электронной подписи (утверждены приказом ФСБ России от 27 декабря 2011 г. № 796). URL: https://base.garant.ru/70139150.

[5] Специальные требования к средствам криптографической защиты, предназначенным для защиты информации, не содержащей сведений, составляющих государственную тайну, и эксплуатируемым на территории Российской Федерации (СТ-Р).

[6] Требования по защите линейной передачи средств криптографической защиты информации, не содержащей сведений, составляющих государственную тайну.

[7] Министерство связи и массовых коммуникаций Российской Федерации. Уведомление об организации перехода на использование схемы электронной подписи по ГОСТ Р 34.10-2012. URL: https://sc.digital.gov.ru/documents/-/document_library/03jzkn2sfJTq/view_file/38476.

[8] Постановление Правительства РФ от 16.04.2012 N 313 «Об утверждении Положения о лицензировании деятельности по разработке, производству, распространению шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств, выполнению работ, оказанию услуг в области шифрования информации, техническому обслуживанию шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств (за исключением случая, если техническое обслуживание шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств, осуществляется для обеспечения собственных нужд юридического лица или индивидуального предпринимателя)». URL: http://www.consultant.ru/document/cons_doc_LAW_128739/.

Приложение 1
Вспомогательные утилиты

Утилита контроля целостности cpverify

Контроль целостности программного обеспечения с помощью алгоритмов хэширования

Модуль cpverify позволяет осуществлять контроль целостности установленного программного обеспечения в ручном и автоматическом режимах.

При помощи перечисленных ниже опций модуль cpverify может быть использован для следующих контрольных целей:

1) Вычисление значения хэш-функции для файла

```
cpverify -mk filename [-alg algid] [-inverted_halfbytes <inv>]
```

- filename — имя файла, для которого производится вычисление значения хэш-функции
- algid — используемый алгоритм хэширования:
  - GR3411 (по умолчанию)
  - GR3411_2012_256
  - GR3411_2012_512
- -inverted_halfbytes <inv> — указывается, если полубайты в хэш-значении hashvalue перевернуты. Значения inv по умолчанию:
  - 1 — для алгоритма GR3411
  - 0 — для алгоритмов GR3411_2012_256 и GR3411_2012_512

2) Проверка целостности файла

```
cpverify filename [-alg algid] [hashvalue] [-inverted_halfbytes <inv>]
```

- filename — имя файла, целостность которого проверяется
- algid — используемый алгоритм хэширования:
  - GR3411 (по умолчанию)
  - GR3411_2012_256
  - GR3411_2012_512
- hashvalue — значение хэш-функции. Если hashvalue не указан, то хэш-значение берется из файла filename.hsh
- -inverted_halfbytes <inv> — указывается, если полубайты в хэш-значении hashvalue перевернуты. Значения inv по умолчанию:
  - 1 — для алгоритма GR3411
  - 0 — для алгоритмов GR3411_2012_256 и GR3411_2012_512

3) Вычисление значения хэш-функции для каждого из файлов, содержащихся в каталоге в разделе реестра

```
cpverify -rm [-alg algid] [catname]
```

- catname — имя каталога в разделе реестра (если catname не указан, то будут пересчитаны все хэш-значения в разделе реестра)

- algid — используемый алгоритм хэширования:
  - GR3411 (по умолчанию)
  - GR3411_2012_256
  - GR3411_2012_512

Примечание. Текущее значение хэш-функций заменяется на вновь посчитанное.

4) Проверка целостности файлов из каталога в разделе реестра

```
cpverify -rv [catname]
```

- catname — имя каталога в разделе реестра (если catname не указан, то будут проверены все файлы в разделе реестра)

5) Вычисление значения хэш-функции для файла и запись полученного значения в реестр

```
cpverify -addreg -file filename
```

- filename — имя файла, для которого производится вычисление значения хэш-функции

6) Удаление значения хэш-функции из реестра

```
cpverify -delreg -file filename
```

- filename — имя файла, для которого значение хэш-функции необходимо удалить из реестра

7) Вычисление значения хэш-функции для файлов, перечисленных в xml-файле, и запись полученных хэш-значений в файл

```
cpverify -xm in_file out_file [-alg algid] [xmlcatname]
```

- in_file — имя xml-файла с перечнем файлов, для которых производится вычисление значений хэш-функции
- xmlcatname — имя каталога в xml-файле с именем in_file с перечислением файлов, для которых производится вычисление значений хэш-функции (если xmlcatname не указан, то хэш-значения будут посчитаны для всех фалов, перечисленных в xml-файле с именем in_file)
- out_file — имя xml-файла, в который записываются вычисленные значения хэш-функции
- algid — используемый алгоритм хэширования:
  - GR3411 (по умолчанию)
  - GR3411_2012_256
  - GR3411_2012_512

Примечание. Текущее значение хэш-функций заменяется на вновь посчитанное.

8) Проверка целостности файлов, перечисленных в xml-файле

```
cpverify -xv in_file [xmlcatname]
```

- in_file — имя xml-файла с перечнем файлов, для которых выполняется проверка целостности
- xmlcatname — имя каталога в xml-файле с именем in_file с перечислением файлов, для которых выполняется проверка целостности (если xmlcatname не указан, то проверка будет выполнена для всех файлов, перечисленных в xml-файле с именем in_file)

9) Формирование xml-файла, содержащего список файлов, находящихся в разделе реестра под контролем целостности, и хэш-значения этих файлов

```
cpverify -r2x out_file
```

- `out_file` — имя xml-файла с перечнем файлов и хэш-значениями

Примечание. Список контролируемых модулей зависит от версии и исполнения СКЗИ.

10) Установка под контроль целостности файлов, перечисленных в xml-файле

```
cpverify -x2r in_file
```

- `in_file` — имя xml-файла с перечнем файлов

Примечание. Список контролируемых модулей может быть получен при помощи команды `cpverify -r2x out_file`.

Алгоритм действий

Для того, чтобы поставить под контроль целостности установленное программное обеспечение, нужно выполнить следующую последовательность действий:

1) Создать xml-файл, содержащий список устанавливаемых под контроль целостности файлов. Данный xml-файл должен иметь следующую структуру:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<CProIntegrity>
    <catalog name="xmlcatname">
        <entry name="calc.exe">
            <Path>C:\WINDOWS\system32\calc.exe</Path>
            <Algid>00008021</Algid>
        </entry>
        <entry name="verifier.exe">
            <Path>C:\WINDOWS\system32\verifier.exe</Path>
            <Algid>00008021</Algid>
        </entry>
    </catalog>
</CProIntegrity>
```

Примечание. Значение поля Algid зависит от используемого алгоритма хэширования:
- GR3411 — значение 0000801E
- GR3411_2012_256 — значение 00008021
- GR3411_2012_512 — значение 00008022

2) Запустить модуль `cpverify -xm in_file out_file TestControl`, указав в качестве параметра in_file имя созданного xml-файла. Результатом работы модуля будет являться xml-файл с именем out_file, содержащий вычисленные значения хэш-функции для перечисленных в in_file файлов и имеющий следующую структуру:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<CProIntegrity>
    <catalog name="xmlcatname">
        <entry name="calc.exe">
            <Path>C:\WINDOWS\system32\calc.exe</Path>
            <Algid>00008021</Algid>
            <Tag>679837307CDC7AA1E4BDBB75194A24D42C782079AF08E2D362D7624A90D604C7</Tag>
```

```
            </entry>
            <entry name="verifier.exe">
                <Path>C:\WINDOWS\system32\verifier.exe</Path>
                <Algid>00008021</Algid>
                <Tag>9DF987B89A323BEBC3C29BAC0AED42A4F5BD651892AAE79F1EC1D05288D06B9C</Tag>
            </entry>
        </catalog>
    </CProIntegrity>
```

Примечание. Значение поля Algid зависит от используемого алгоритма хэширования:
- GR3411 — значение 0000801E
- GR3411_2012_256 — значение 00008021
- GR3411_2012_512 — значение 00008022

3) Установить под контроль целостности файлы, для которых было вычислено значение хэш-функции, используя модуль cpverify -x2r in_file TestControl, где параметром in_file является xml-файл, полученный в результате вычисления значения хэш-функции в пункте **2)**.

Контроль целостности программного обеспечения с помощью алгоритмов подписи

1) Проверка подписи файла

```
cpverify -file_verify filename [signval] -timestamp date
```

- filename — имя файла, для которого производится проверка подписи
- signval — значение подписи, байтовая строка (если параметр signval не указан, то значение подписи берется из файла filename.sgn)
- date — дата формирования подписи в формате ДД.ММ.ГГГГ

Примечание. Данная команда проверяет подпись с прямой последовательностью полубайт, для проверки подписи с обратной последовательностью байт необходимо использовать команду versign с аналогичным набором параметров. Подпись проверяется на открытом ключе из специального сертификата для подписи кода компании «КРИПТО-ПРО».

2) Создание подписи файла

```
cpverify -file_sign filename -cont cont_name [-pin password]
   [-provname имя_провайдера] [-provtype тип_провайдера]
```

- filename — имя файла, для которого производится формирование подписи
- cont_name — контейнер ключа, который используется для формирования подписи
- password — пароль контейнера ключа
- Provname — имя используемого провайдера
- Provtype — тип используемого провайдера:
  - 75 (по умолчанию)
  - 80
  - 81

Утилита безопасного удаления файлов wipefile

```
wipefile <filename> [<delete>]
```

Модуль `wipefile` предназначен для уничтожения файла с физическим удалением его содержимого. Если в качестве опции указан только целевой файл <filename>, то утилита записывает в него 0 и 1. При указании опции <delete> утилита удаляет файл; удаление файла производится после затирания содержимого файла.

После установки КриптоПро CSP утилита `wipefile` расположена в:

- `C:\Program Files (x86)\Crypto Pro\CSP\wipefile` в ОС семейства Windows;
- `/opt/cprocsp/bin/<архитектура>/wipefile` в ОС семейства Linux/Unix;
- `/opt/cprocsp/bin` в ОС Mac OS.

Утилита для формирования и экспорта ключей в формате PKCS#12 HsmGenArm

```
HsmGenArm --pwd=<password> --pfx=<filename> [--curve=<oid>]
```

Модуль `HsmGenArm` предназначен для формирования ключей в КриптоПро HSM и экспорта ключей в формате PKCS#12 на АРМ с установленным модулем для последующей записи на ключевой носитель пользователя.

Примечание. Модуль HsmGenArm поставляется в комплекте с КриптоПро CSP для ОС семейств Windows и Linux/Unix.

Допустимые значения параметров приведены в табл. 1. Порядок параметров в командной строке не важен.

Таблица 1. Описание параметров утилиты HsmGenArm

Параметр	Описание	Значение по умолчанию
`--pwd=<password>`	Пароль к pfx-файлу	—
`--pfx=<filename>`	Путь к pfx-файлу	—
`[--curve=<oid>]` `(опциональный параметр)`	Параметры генерируемого ключа — идентификатор эллиптической кривой Допустимые значения: 1.2.643.7.1.2.1.1.1 1.2.643.7.1.2.1.1.2 1.2.643.7.1.2.1.1.3 1.2.643.7.1.2.1.1.4	1.2.643.7.1.2.1.1.1

Коды ошибок

В табл. 2 перечислены возможные коды ошибок. Детальная информация выводится в текстовом виде в стандартный поток вывода ошибок утилиты.

Таблица 2. Коды ошибок утилиты HsmGenArm

Код ошибки	Описание
0	Операция выполнена успешно
1	Произошла внутренняя ошибка
2	Недостаточно памяти
3	Ошибка устройства
4	Криптографическая ошибка
5	Командная строка утилиты задана некорректно
6	Ошибка обращения к HSM
7	Ошибка генерации ключа

8	Ошибка формирования транспортного контейнера (pfx)
10	Невозможно создать файл по указанному пути (pfx)

Параметры журналирования

В процессе работы утилиты при создании ключей ведется аудит экспорта ключей. Параметры журналирования указаны в табл. 3.

Таблица 3. Параметры журналирования утилиты HsmGenArm

Параметр	Значение
Дата	Дата в формате dd.mm.yyyy
Значение ключа проверки ЭП	Байтовая строка
Алгоритм ЭП	Алгоритм подписи ГОСТ Р 34.10-2012 с ключом 256 (OID = 1.2.643.7.1.1.1.1) *(значение по умолчанию)*
Параметры алгоритма ЭП	Рабочие параметры A алгоритма подписи ГОСТ Р 34.10-2012 с ключом 256 (OID = 1.2.643.7.1.2.1.1.1) *(значение по умолчанию)* Рабочие параметры B алгоритма подписи ГОСТ Р 34.10-2012 с ключом 256 (OID = 1.2.643.7.1.2.1.1.2) Рабочие параметры C алгоритма подписи ГОСТ Р 34.10-2012 с ключом 256 (OID = 1.2.643.7.1.2.1.1.3) Рабочие параметры D алгоритма подписи ГОСТ Р 34.10-2012 с ключом 256 (OID = 1.2.643.7.1.2.1.1.4)
Результат выполнения (код ошибки)	Возможные значения приведены в табл. 2

Требования безопасности при использовании утилиты

При разработке информационных систем (ИС) с использованием функций утилиты `HsmGenArm` должны выполняться следующие требования:

1) Порядок работы с экспортированным с использованием утилиты `HsmGenArm` ключом определяется исходя из требований ИС, в которой эксплуатируется СКЗИ, с учётом положений документов Р 1323565.1.040-2022 и Р 1323565.1.041-2022. Порядок должен быть зафиксирован в документации на эту систему и проверен в рамках работ по оценке влияния (см. разд. 4.2).

2) Должен быть обеспечен контроль срока действия ключа, экспортированного с использованием утилиты `HsmGenArm`, с учётом требований, предъявляемых к ИС, в которой эксплуатируется СКЗИ.

3) Для ключа, экспортированного с использованием утилиты `HsmGenArm`, установленным порядком должен быть выпущен сертификат.

4) На носитель с pfx-файлом с экспортированным ключом должны распространяться требования безопасности к хранению и использованию ключевых носителей (см. разд. 3).

5) При переносе экспортированных с использованием утилиты `HsmGenArm` ключей на отделяемые ключевые носители необходимо обеспечить гарантированное удаление pfx-файла с экспортированными ключами (например, с использованием утилиты wipefile).

6) Формирование и экспорт ключа в pfx-файл с использованием утилиты `HsmGenArm`, перенос этого ключа на отделяемый ключевой носитель и гарантированное удаление указанного pfx-файла должны выполняться в рамках одного сеанса работы вызывающего ПО.

7) Повторное использование пароля к pfx-файлу допускается только в рамках одного сеанса работы вызывающего ПО.

8) Должна быть исключена возможность одновременного доступа (как локального, так и удалённого) нескольких пользователей вызывающего ПО к СВТ, на которых реализовано данное ПО.

Приложение 2
Перечень вызовов, использование которых при разработке систем на основе СКЗИ «КриптоПро CSP» версия 5.0 R3 КС3 (исполнение 3-Base) с учётом п.1.5 Формуляра ЖТЯИ.00103-03 30 01 возможно без дополнительных тематических исследований

Интерфейс CryptoAPI

Функция	Описание	Ограничения на использование функции
Функции инициализации и настройки провайдера		
`CryptAcquireContext`	Функция создаёт дескриптор криптопровайдера с именем ключевого контейнера, определённым параметром pszContainer.	Допускается использование только провайдеров из состава СКЗИ «КриптоПро CSP» (типы 75, 80 и 81). Полученный дескриптор криптопровайдера должен быть в обязательном порядке освобождён с помощью вызова функции CryptReleaseContext (за исключением вызовов с флагом CRYPT_ DELETEKEYSET).
`CryptReleaseContext`	Функция удаляет дескриптор криптопровайдера, созданного CryptAcquireContext.	Перед вызовом данной функции все дескрипторы объектов ключей и хэширования, работа с которыми производилась совместно с удаляемым дескриптором криптопровайдера, должны быть удалены с помощью вызовов CryptDestroyKey и CryptDestroyHash соответственно.
`CryptContextAddRef`	Функция управляет счётчиком дескрипторов созданного CryptAcquireContext.	
`CryptEnumProviders`	Функция перечисляет установленные криптопровайдеры.	
`CryptEnumProviderTypes`	Функция перечисляет установленные типы криптопровайдеров.	
`CryptGetDefaultProvider`	Функция получает контекст провайдера, установленного в системе по умолчанию.	

CryptGetProvParam	Функция получает параметры криптопровайдера.	
CryptSetProvParam	Функция устанавливает параметры криптопровайдера.	
CryptInstallDefaultContext, CryptSetProvider, CryptSetProviderEx, CryptUninstallDefaultContext	Функции управляют контекстом провайдера по умолчанию.	Допускается использование только провайдеров из состава СКЗИ «КриптоПро CSP» (типы 75, 80 и 81).
Функции генерации и обмена ключами, создания, конфигурирования и удаления ключей		
CryptGenKey	Функция генерирует случайные криптографические ключи или ключевую пару (закрытый/открытый ключи).	Полученный дескриптор ключа должен в обязательном порядке быть удалён с помощью вызова функции CryptDestroyKey до вызова функции CryptReleaseContext для рабочего дескриптора криптопровайдера.
CryptDestroyKey	Функция удаляет ключ, передаваемый через параметр hKey. После удаления ключ (дескриптор ключа) не может использоваться.	
CryptExportKey	Функция экспортирует криптографические ключи из ключевого контейнера криптопровайдера, сохраняя их в защищённом виде.	Разрешено экспортировать только открытые ключи (PUBLICKEYBLOB) с нулевым значением флага или со значением флага CRYPT_ PUBLICCOMPRESS.
CryptGenRandom	Функция заполняет буфер случайными байтами.	
CryptGetKeyParam	Функция возвращает параметры ключа.	
CryptGetUserKey	Функция возвращает дескриптор одной из долговременных ключевых пар в ключевом контейнере.	Полученный дескриптор ключа должен в обязательном порядке быть удалён с помощью вызова функции CryptDestroyKey до вызова функции CryptReleaseContext для рабочего дескриптора криптопровайдера.
CryptImportKey	Функция импортирует криптографический ключ из ключевого блоба в контейнер криптопровайдера.	Разрешено импортировать только открытые ключи (PUBLICKEYBLOB) при нулевом значении флагов. Полученный дескриптор ключа должен в обязательном порядке быть удалён с помощью вызова функции CryptDestroyKey до вызова функции CryptReleaseContext для рабочего дескриптора криптопровайдера.

CryptSetKeyParam	Функция устанавливает параметры ключа.	Разрешено использование только со следующими символьными аргументами: KP_CERTIFICATE, KP_CIPHEROID, KP_DHOID, KP_HASHOID, KP_AUDIT_CONTROL.
Функции обработки криптографических сообщений		
CryptSignMessage	Функция создаёт хэш определённого содержания, подписывает хэш и затем производит кодирование текста исходного сообщения и подписанного хэша.	
CryptVerifyMessageSignature	Функция проверяет электронную подпись сообщения.	
CryptVerifyDetachedMessageSignature	Функция проверяет подписанное сообщение, содержащее отсоединённую (detached) подпись или подписи.	
CryptDecodeMessage	Функция декодирует, расшифровывает и проверяет сообщение.	
CryptDecryptAndVerifyMessageSignature	Функция декодирует и проверяет сообщение.	
CryptEncryptMessage	Функция зашифровывает и производит кодирование сообщения.	
CryptDecryptMessage	Функция декодирует и расшифровывает сообщения. Не допускается автоматический анализ результата работы функции, направленный на проверку корректности сообщения.	
CryptGetMessageCertificates	Функция возвращает дескриптор хранилища сертификатов и CRL из сообщения.	
CryptGetMessageSignerCount	Функция возвращает количество подписантов сообщения.	
CryptHashMessage	Функция создаёт хэшированное сообщение.	
CryptSignAndEncryptMessage	Функция создаёт подписанное и зашифрованное сообщение.	
CryptSignMessageWithKey	Функция создаёт подписанное сообщение.	
CryptVerifyDetachedMessageHash	Функция проверяет отсоединённый хэш.	

CryptVerifyMessageHash	Функция проверяет хэшированное сообщение.	
CryptVerifyMessageSignatureWithKey	Функция проверяет подписанное сообщение.	
CryptMsgCalculateEncodedLength	Функция вычисляет максимальное количество байтов, необходимое для кодированного криптографического сообщения, заданного типом сообщения, параметрами кодирования и общей длиной информации, которая должна быть кодирована.	
CryptMsgOpenToEncode	Функция открывает криптографическое сообщение для кодирования и возвращает дескриптор открытого сообщения.	
CryptMsgOpenToDecode	Функция открывает криптографическое сообщение для декодирования и возвращает дескриптор открытого сообщения.	
CryptMsgUpdate	Функция пополняет текст криптографического сообщения.	
CryptMsgGetParam	Функция получает параметр сообщения после того, как криптографическое сообщение было кодировано или декодировано.	
CryptMsgControl	Функция выполняет контрольное действие.	
CryptMsgClose	Функция закрывает дескриптор криптографического сообщения.	
CryptMsgDuplicate	Функция дублирует дескриптор криптографического сообщения путём увеличения счётчика ссылок.	

Функции работы с алгоритмами хэширования		
`CryptCreateHash`	Функция инициализирует дескриптор нового объекта функции хэширования потока данных.	Разрешено использование только со следующими символьными аргументами: • CALG_GR3411, • CALG_GR3411_2012_256, • CALG_GR3411_2012_512, • CALG_GR3411_HMAC, • CALG_GR3411_2012_256_HMAC, • CALG_GR3411_2012_512_HMAC, • CALG_PBKDF2_94_256, • CALG_PBKDF2_2012_256, • CALG_PBKDF2_2012_512, • CALG_SHAREDKEY_HASH. Дескриптор, полученный с использованием аргумента CALG_GR3411, запрещено подавать в функцию CryptSignHash. Полученный дескриптор объекта хэширования должен в обязательном порядке быть удалён с помощью вызова функции CryptDestroyHash до вызова функции CryptReleaseContext для рабочего дескриптора криптопровайдера.
`CryptDestroyHash`	Функция удаляет объект функции хэширования.	
`CryptDuplicateHash`	Функция создаёт точную копию объекта функции хэширования, включая все его переменные, определяющие внутреннее состояние объекта функции хэширования.	Полученный дескриптор объекта хэширования должен в обязательном порядке быть удалён с помощью вызова функции CryptDestroyHash до вызова функции CryptReleaseContext для рабочего дескриптора криптопровайдера.
`CryptGetHashParam`	Функция возвращает параметры объекта функции хэширования и значение функции хэширования.	Запрещено использование с символьными аргументами HP_OPAQUEBLOB, HP_HASHSTATEBLOB.
`CryptHashCertificate`	Функция вычисляет хэш-функцию от всего содержимого переданного буфера данных сертификата (включая электронную подпись этого сертификата).	
`CryptHashData`	Функция передаёт данные указанному объекту функции хэширования.	

CryptSetHashParam	Функция устанавливает параметры объекта хэширования.	Разрешено использование только со следующими символьными аргументами: • HP_HASHSTARTVECT (при условии, что передаваемый объект функции хэширования был создан функцией CryptCreateHash с символьным аргументом CALG_GR3411), • HP_PBKDF2_SALT, • HP_PBKDF2_PASSWORD, • HP_PBKDF2_COUNT, • HP_OID/KP_HASHOID, • HP_OPEN.
CryptSignHash	Функция возвращает значение электронной подписи от значения функции хэширования.	Запрещено использовать дескриптор, полученный вызовом функции CryptCreateHash с использованием аргумента CALG_GR3411.
CryptVerifySignature	Функция проверяет электронную подпись.	Разрешено использование только с дескрипторами ключей, полученными ранее с помощью вызова CryptImportPublicKeyInfo (CryptImportPublicKeyInfoEx) из сертификата, проверенного с помощью функции CertVerifyCertificateChainPolicy.
Функции работы с сертификатами, списками отозванных (аннулированных) сертификатов (CRL), хранилищем сертификатов		
Списки отозванных сертификатов		
CertAddCRLContextToStore	Функция добавляет контекст CRL в хранилище сертификатов.	
CertAddCRLLinkToStore	Функция создаёт ссылку на CRL в другом хранилище.	
CertAddEncodedCRLToStore	Функция создаёт контекст CRL из кодированного CRL и добавляет его в хранилище сертификатов. Функция создаёт копию контекста CRL перед добавлением его в хранилище.	
CertEnumCRLsInStore	Функция получает первый или следующий CRL в хранилище. Эта функция используется в цикле для того, чтобы последовательно получить все CRL в хранилище.	

CertFreeCRLContext	Функция освобождает контекст CRL, уменьшая счётчик ссылок на единицу. Когда счётчик ссылок обнуляется, функция освобождает память, выделенную под контекст CRL.	
CertCreateCRLContext	Функция создаёт контекст CRL из кодированного CRL. Созданный контекст не помещается в хранилище сертификатов. В созданном контексте функция размещает копию кодированного CRL.	
CertDeleteCRLFromStore	Функция удаляет определённый контекст CRL из хранилища сертификатов.	
CertDuplicateCRLContext	Функция дублирует контекст CRL, увеличивая счётчик ссылок на СОС на единицу.	
CertFindCRLInStore	Функция находит первый или следующий контекст CRL в хранилище сертификатов, который соответствует критерию поиска, определяемому параметром dwFindType и связанным с ним pvFindPara. Эта функция может быть использована в цикле для того, чтобы найти все CRL в хранилище сертификатов, удовлетворяющие заданному критерию поиска.	
CertFindCertificateInCRL	Функция осуществляет поиск заданного сертификата в CRL.	
CertGetCRLFromStore	Функция получает первый или следующий контекст CRL для определённого издателя сертификата из хранилища сертификатов. Эта функция также осуществляет возможную проверку CRL.	
CertSerializeCRLStoreElement	Функция осуществляет сериализацию CRL со своими свойствами.	
CertVerifyCRLRevocation	Функция проверяет наличие сертификата в указанном CRL.	
CertVerifyCRLTimeValidity	Функция проверяет актуальность CRL на момент времени, указанный через входной параметр.	Перед вызовом функции требуется выполнять проверку на неравенство NULL параметра pCrlInfo.

Расширенные свойства сертификата, списка отозванных (аннулированных) сертификатов (CRL) и CTL		
`CertGetCRLContextProperty`	Функция получает расширенное свойство определённого контекста CRL.	
`CertSetCRLContextProperty`	Функция устанавливает расширенное свойство определённого контекста CRL.	
`CertGetCertificateContextProperty`	Функция получает информацию, содержащуюся в расширенном свойстве контекста сертификата.	
`CertEnumCertificateContextProperties`	Функция перечисляет информацию, содержащуюся в расширенных свойствах контекста сертификата.	
`CertSetCertificateContextProperty`	Функция устанавливает расширенное свойство для определённого контекста сертификата.	
`CertEnumCRLContextProperties`	Функция перечисляет расширенные свойства CRL.	
`CertEnumCTLContextProperties`	Функция перечисляет расширенные свойства CTL.	
`CertGetCTLContextProperty`	Функция получает расширенное свойство CTL.	
`CertSetCTLContextProperty`	Функция устанавливает расширенное свойство CTL.	
`CertAddCTLContextToStore`	Функция добавляет контекст CTL в хранилище сертификатов.	
`CertCreateCTLContext`	Функция создаёт контекст кодированного CTL. Созданный контекст не помещается в хранилище сертификатов. Функция делает копию CTL в созданном контексте.	
`CertDuplicateCTLContext`	Функция дублирует контекст CTL, увеличивая счётчик ссылок на CTL на единицу.	
`CertFreeCTLContext`	Функция освобождает контекст CTL, уменьшая счётчик ссылок на единицу. Когда счётчик ссылок обнуляется, функция освобождает память, выделенную под контекст CTL.	
Сертификаты		
`CertAddCertificateContextToStore`	Функция добавляет контекст сертификата в хранилище сертификатов.	

CertAddCertificateLinkToStore	Функция добавляет ссылку на сертификат в другом хранилище.	
CertAddEncodedCertificateToStore	Функция создаёт контекст сертификата из кодированного сертификата и добавляет его в хранилище сертификатов. Созданный контекст не содержит никаких расширенных свойств.	
CertEnumCertificatesInStore	Функция получает первый или следующий сертификат в хранилище сертификатов. Эта функция используется в цикле для того, чтобы последовательно получить все сертификаты в хранилище сертификатов.	
CertFreeCertificateContext	Функция освобождает контекст сертификата, уменьшая счётчик ссылок на единицу.	
CertCreateCertificateContext	Функция создаёт контекст сертификата из кодированного сертификата. Созданный контекст не помещается в хранилище сертификатов. В созданном контексте функция размещает копию кодированного сертификата.	
CertDuplicateCertificateContext	Функция дублирует контекст сертификата, увеличивая счётчик ссылок на единицу.	
CertFindCertificateInStore	Функция находит первый или следующий контекст сертификата в хранилище сертификатов, который соответствует критерию поиска, определяемому параметром dwFindType и связанным с ним pvFindPara.	
CertDeleteCertificateFromStore	Функция удаляет определённый контекст сертификата из хранилища сертификатов.	
CertGetSubjectCertificateFromStore	Функция получает контекст сертификата из хранилища сертификатов, однозначно определяемый его издателем и серийным номером.	
CertGetIssuerCertificateFromStore	Функция осуществляет поиск сертификатов издателей заданного сертификата.	

CertGetValidUsages	Функция осуществляет поиск пересечения KeyUsage для массива сертификатов.	
CertSerializeCertificateStoreElement	Функция осуществляет сериализацию кодированных объектов (сертификата, CRL и CTL) и их свойств, расположенных в контексте объекта.	Перед вызовом функции требуется выполнять проверку на неравенство NULL параметров pCertContext, pbElement и pcbElement.
CertComparePublicKeyInfo	Функция сравнивает два открытых ключа и определяет, являются ли они идентичными.	
CertFindExtension	Функция находит расширение в массиве и возвращает указатель на него.	
CertGetPublicKeyLength	Фукнция возвращает длину ключа в битах.	
CertGetIntendedKeyUsage	Функция получает назначение ключа из сертификата.	
CertCompareCertificateName	Функция сравнивает два сертификата и определяет, являются ли они идентичными.	
CryptSignCertificate	Функция формирует значение хэш-функции/электронной подписи для поля «To Be Signed» в кодированном объекте X509 (которым может являться сертификат, CRL, CTL или запрос на сертификат).	Для формирования электронной подписи разрешается передача по указателю pSignatureAlgorithm структуры, содержащей только следующие значения OID-идентификаторов: • szOID_CP_GOST_R3411_12_256_R3410 ("1.2.643.7.1.1.3.2") • szOID_CP_GOST_R3411_12_512_R3410 ("1.2.643.7.1.1.3.3") Для вычисления хэш-функции разрешается передача по указателю pSignatureAlgorithm структуры, содержащей только следующие возможные варианты значения OID-идентификаторов: • szOID_CP_GOST_R3411_12_256 ("1.2.643.7.1.1.2.2") • szOID_CP_GOST_R3411_12_512 ("1.2.643.7.1.1.2.3")

CryptSignAndEncodeCertificate	Функция формирует значение хэш-функции/электронной подписи для поля «To Be Signed» в кодированном объекте X509 (которым может являться сертификат, CRL, CTL или запрос на сертификат), кодирует результат вычисления и размещает в объекте X509.	Разрешается использование функции только для формирования запроса на сертификат с передачей в качестве параметра lpszStructType только символьного значения X509_CERT_REQUEST_TO_BE_SIGNED (0x3). Разрешается передача по указателю pSignatureAlgorithm структуры, содержащей только следующие возможные значения OID-идентификаторов: • szOID_CP_GOST_R3411_12_256_R3410 ("1.2.643.7.1.1.3.2") • szOID_CP_GOST_R3411_12_512_R3410 ("1.2.643.7.1.1.3.3")
CertVerifyTimeValidity	Функция проверяет актуальность срока действия сертификата на момент времени, указанный через входной параметр.	Перед вызовом функции требуется выполнять проверку на неравенство NULL параметра pCertInfo.
CertVerifyValidityNesting	Функция проверяет то, что срок действия указанного сертификата не выходит за рамки срока действия сертификата издателя.	Перед вызовом функции требуется выполнять проверку на неравенство NULL параметров pSubjectInfo и pIssuerInf.
CryptVerifyCertificateSignature	Функция проверяет электронную подпись в сертификате, CRL или запросе на сертификат с использованием открытого ключа.	Допускается использование только совместно с проверкой всей цепочки сертификатов для обрабатываемого объекта PKI.
CryptVerifyCertificateSignatureEx	Функция проверяет электронную подпись в сертификате, CRL, запросе на сертификат или подписанном ответе от OCSP-сервера с использованием как объекта открытого ключа заявителя, так и сертификата/цепочки сертификатов.	Допускается использование только совместно с проверкой всей цепочки сертификатов для обрабатываемого объекта PKI.
CryptGetObjectUrl	Функция получает в отношении сертификата, CTL или CRL путь URL удалённого объекта.	
OCSP		
CertAddRefServerOcspResponse	Функция увеличивает счётчик ссылок на OCSP ответ.	
CertAddRefServerOcspResponseContext	Функция увеличивает счётчик ссылок на контекст OCSP ответа.	
CertOpenServerOcspResponse	Функция открывает дескриптор OCSP ответа для заданной цепочки сертификатов.	

CertGetServerOcspResponseContext	Функция получает контекст OCSP ответа.	
CertCloseServerOcspResponse	Функция закрывает дескриптор OCSP ответа.	
Оконные функции		
CertSelectCertificate	Функция отображает диалог выбора сертификата по заданным критериям.	
CryptUIDlgCertMgr	Функция отображает диалог управления сертификатами.	
CryptUIDlgSelectCertificate	Функция отображает диалог выбора сертификата.	
CryptUIDlgSelectCertificateFrom-Store	Функция отображает диалог выбора сертификата из хранилища.	
CryptUIDlgViewCertificate	Функция отображает диалог со свойствами сертификата.	
CryptUIDlgViewContext	Функция отображает сертификат, CRL или CTL.	
CryptUIDlgViewSignerInfo	Функция отображает диалог с информацией о подписанте.	
CertSelectionGetSerializedBlob	Функция осуществляет сериализацию сертификата из структуры, используемой для отображения.	
GetFriendlyNameOfCert	Функция преобразует имя сертификата к «читаемому» виду.	
Проверка цепочек сертификатов		
CertVerifyCertificateChainPolicy	Функция проверяет цепочку сертификатов на достоверность, включая соответствие критерию истинности.	
CertGetCertificateChain	Функция строит цепочку сертификатов, начиная с последнего сертификата, в обратном направлении до доверенного корневого сертификата, если это возможно.	
CertFreeCertificateChain	Функция освобождает цепочку сертификатов путём уменьшения счётчика ссылок. Если счётчик ссылок равен нулю, то память, выделенная под цепочку, освобождается.	

CertCreateCertificateChainEngine	Функция создаёт контекст HCERTCHAINENGINE, который позволяет изменять параметры механизма построения цепочки сертификатов. Позволяет ограничивать множество доверенных сертификатов.	
CertFreeCertificateChainEngine	Функция освобождает контекст HCERTCHAINENGINE.	
CertCreateCTLEntryFromCertificateContextProperties	Функция создаёт CTL на основе свойств атрибутов контекста сертификата.	
CertDuplicateCertificateChain	Функция дублирует контекст цепочки.	
CertFindChainInStore	Функция строит цепочку по заданным критериям из хранилища.	
CertFreeCertificateChainList	Функция освобождает массив цепочек.	
CertIsValidCRLForCertificate	Функция проверяет наличие сертификата в CRL.	
CertSetCertificateContextPropertiesFromCTLEntry	Функция устанавливает свойства в контекст сертификата на основе CTL.	
Расширенные свойства сертификата (EKU)		
CertGetEnhancedKeyUsage	Функция получает информацию о расширенном использовании ключа из соответствующего расширения или из расширенных свойств сертификата. Расширенное использование ключа служит признаком правомерного использования сертификата.	
CryptAcquireCertificatePrivateKey	Функция получает дескриптор HCRYPTPROV и параметр dwKeySpec для определённого контекста сертификата.	
Идентификаторы		
CryptFindOIDInfo	Функция получает первую предопределённую или зарегистрированную структуру CRYPT_OID_INFO, согласованную с определённым типом ключа и с ключом.	
CryptEnumOIDInfo	Функция перечисляет зарегистрированные идентификаторы и получает информацию для них.	

Хранилище сертификатов		
`CertOpenStore`	Функция открывает хранилище сертификатов, используя заданный тип провайдера.	
`CertDuplicateStore`	Функция дублирует дескриптор хранилища, увеличивая счётчик ссылок на хранилище на единицу.	
`CertOpenSystemStore`	Функция открывает наиболее часто используемые хранилища сертификатов.	
`CertCloseStore`	Функция закрывает дескриптор хранилища сертификатов и уменьшает счётчик ссылок на хранилища на единицу.	
`CertAddStoreToCollection`	Функция добавляет хранилище в коллекцию.	
`CertControlStore`	Функция управляет уведомлениями о различиях между содержимыми закэшированного хранилища и физического хранилища.	
Работа с открытыми данными и объектами		
`CryptImportPublicKeyInfoEx`	Функция импортирует информацию об открытом ключе в CSP и возвращает дескриптор открытого ключа.	
`CryptImportPublicKeyInfo`	Функция преобразовывает и импортирует информацию об открытом ключе в провайдер и возвращает дескриптор открытого ключа.	
`CryptExportPublicKeyInfoEx`	Функция экспортирует информацию об открытом ключе, связанную с соответствующим закрытым ключом провайдера.	
`CryptExportPublicKeyInfo`	Функция экспортирует информацию об открытом ключе, ассоциированную с соответствующим закрытым ключом провайдера.	
`CertCompareCertificate`	Функция сравнивает два сертификата для того, чтобы определить, являются ли они идентичными.	

CertCompareIntegerBlob	Функция сравнивает два целочисленных блоба для определения того, представляют ли они собой два равных числа.	
Кодирование/декодирование		
CryptDecodeObject	Функция декодирует сертификаты, CRL и запросы на сертификаты.	
CryptDecodeObjectEx	Функция декодирует сертификаты, CRL и запросы на сертификаты.	
CryptEncodeObject	Функция кодирует сертификаты, CRL и запросы на сертификаты.	
CryptEncodeObjectEx	Функция кодирует сертификаты, CRL и запросы на сертификаты.	
Получение объектов из удалённых источников		
CryptRetrieveObjectByUrlA	Функция получает объект инфраструктуры открытых ключей по заданному URL.	
CryptRetrieveObjectByUrlW	Функция является unicode версией функции CryptRetrieveObjectByUrlA.	
Дополнительные функции		
CryptBinaryToString	Функция переводит двоичную строку в строку Base64/HEX.	
CryptStringToBinary	Функция переводит строку Base64/HEX в двоичную строку.	
CryptQueryObject	Функция получает сведения о содержимом объекта интерфейса CryptoAPI (например, сертификате, CRL или CTL) в формате соответствующего хэндла.	
CertFindAttribute	Функция производит поиск атрибута сертификата по идентификатору.	
CertGetNameString	Функция получает имя владельца или издателя сертификата.	
CertNameToStr	Функция декодирует имя из ASN.1 структуры в DN (RFC1779).	
CertSaveStore	Функция записывает хранилище сертификатов (включая списки отозванных и доверенных сертификатов) в виде структуры PKCS#7 или бинарного дампа в память или файл.	

CryptFindCertificateKeyProvInfo	Функция осуществляет поиск закрытого ключа, соответствующего открытому ключу сертификата.	
CryptHashPublicKeyInfo	Функция осуществляет ASN.1-кодирование и хэширование структуры CERT_PUBLIC_KEY_INFO .	
CryptMsgCountersign	Функция вырабатывает добавочную подпись.	
CryptMsgCountersignEncoded	Функция вырабатывает добавочную подпись (кодирует структуру SignerInfo, как определено в PKCS#7).	
CryptMsgVerifyCountersignature-Encoded	Функция проверяет добавочную подпись (декодирует структуру SignerInfo, как определено в PKCS#7).	
CryptMsgVerifyCountersignature-EncodedEx	Функция проверяет добавочную подпись (декодирует структуру SignerInfo, как определено в PKCS#7).	
CryptMemFree	Функция освобождает память, которая была ранее выделена с использованием CryptMemAlloc или CryptMemRealloc.	
CertStrToName	Функция преобразует X.500-строку с завершающим NULL-символом в кодированное имя сертификата.	
SignError	Функция получает код последней ошибки в формате HRESULT.	
SignerFreeSignerContext	Функция освобождает память, содержащую кодированные подписанные данные файла.	

Интерфейс cpcurl

Функция	Описание	Ограничения на использование функции
Функции установки TLS-соединения и приёма/передачи данных		
curl_global_init	Функция инициализирует работу библиотеки.	Не является потокобезопасной, должна быть вызвана единожды перед началом использования библиотеки. После завершения использования библиотеки необходимо вызвать curl_global_cleanup.
curl_global_cleanup	Функция деинициализирует работу библиотеки.	Не является потокобезопасной, должна быть вызвана единожды после завершения использования библиотеки.
curl_easy_init	Функция инициализирует сессию работы с библиотекой.	Является потокобезопасной. Должна быть вызвана только после curl_global_init. Полученный вызовом функции хэндл после завершения работы с ним должен быть освобождён вызовом curl_easy_cleanup.
curl_easy_cleanup	Функция освобождает ресурсы, занятые в рамках указанной сессии работы с библиотекой.	Является потокобезопасной. Должна быть вызвана до вызова функции curl_global_cleanup.
curl_easy_setopt	Функция меняет параметры работы TLS-сессии.	Запрещено использовать с параметрами: • CURLOPT_VERBOSE; • CURLOPT_FTP_SSL_CCC; • CURLOPT_ OPENSOCKETFUNCTION; • CURLOPT_KEYPASSWD; • CURLOPT_PROXY_KEYPASSWD; • CURLOPT_SSL_VERIFYPEER; • CURLOPT_PROXY_SSL_ VERIFYPEER; • CURLOPT_SSL_VERIFYHOST; • CURLOPT_PROXY_SSL_ VERIFYHOST; • CURLOPT_SSL_OPTIONS; • CURLOPT_PROXY_SSL_ OPTIONS; • CURLOPT_PROXY; • CURLOPT_PROXYTYPE; • CURLOPT_HTTPPROXYTUNNEL; • CURLOPT_PRE_PROXY; • CURLOPT_PROXY_TRANSFER_ MODE.

curl_easy_getinfo	Функция получает сведения о соединении.	
curl_easy_perform	Функция передает данные в рамках установленного TLS-соединения.	Используемый дескриптор CURL должен быть получен вызовом curl_easy_init. Перед вызовом данной функции с помощью curl_easy_setopt и параметров CURLOPT_STRICT_GOST, CURLOPT_NOPROXY, CURLOPT_REDIR_PROTOCOLS, CURLOPT_FTPSSLAUTH необходимо: • задать проведение проверки алгоритма ключа проверки ЭП для всех сертификатов в цепочке сертификата сервера; • запретить использовать прокси-сервер при соединении с указанным TLS-сервером; • ограничить возможность redirect-перехода с TLS-соединения на незащищённое; • задать использование протокола TLS для защиты канала FTPS-соединения. При установке TLS-соединения с двусторонней аутентификацией перед вызовом данной функции с помощью curl_easy_setopt и параметра CURLOPT_PROXY_SSLCERT и/или CURLOPT_SSLCERT необходимо указать адрес сертификата TLS-клиента в системном хранилище сертификатов.
curl_easy_send	Функция отправляет данные типа raw-data по установленному ранее каналу.	Используемый дескриптор CURL должен быть получен вызовом curl_easy_init. Для вызова функции необходимо предварительно вызвать curl_easy_setopt (с указанием флага CURLOPT_CONNECT_ONLY) и curl_easy_perform.
curl_easy_recv	Функция принимает данные типа raw-data по установленному ранее каналу.	Используемый дескриптор CURL должен быть получен вызовом curl_easy_init. Для вызова функции необходимо предварительно вызвать curl_easy_setopt (с указанием флага CURLOPT_CONNECT_ONLY) и curl_easy_perform.

Вспомогательные функции обработки строк и данных		
curl_easy_escape	Функция переводит заданную строку в URL-кодировку.	
curl_easy_unescape	Функция переводит заданную URL-кодированную строку в строку символов char.	
curl_formadd	Функция добавляет новую секцию в состав данных типа multipart/formdata, отправляемых по HTTP POST-запросу.	
curl_formfree	Функция освобождает ресурсы, выделенные для данных типа multipart/formdata, составленных ранее набором вызовов curl_formadd.	
curl_slist_append	Функция добавляет строку в список строк структуры curl_slist.	
curl_slist_free_all	Функция освобождает все ресурсы, занятые структурой curl_slist.	

Интерфейс КриптоПро TSP SDK

Метод/свойство/команда	Описание	Ограничения на использование метода/свойства/команды
Низкоуровневый интерфейс КриптоПро TSP (C++) (пространство имён CryptoPro::PKI::TSP)		
`Конструктор класса CAccuracy`	Конструктор создаёт объект CAccuracy, хранящий значение допустимого отклонения от указываемого в штампе времени UTC (используется для формирования TSTInfo поля в штампе времени).	
`Конструктор класса CMSRequest`	Конструктор создаёт объект класса CMSRequest запроса на штамп времени Microsoft (MS TS), хранящий данные (на которые запрашивается TS), атрибуты и OID алгоритма TS.	
`Метод encode класса CMSRequest`	Метод выполняет ASN.1-кодирование сформированного запроса на MS TS.	
`Метод decode класса CMSRequest`	Метод выполняет декодирование двоичного блоба с ASN.1-структурой сформированного запроса на MS TS.	
`Метод get_timeStampAlgorithm класса CMSRequest`	Метод возвращает строковое значение OID-идентификатора алгоритма TS, записанного в запросе MS TS.	
`Метод get_contentInfo класса CMSRequest`	Метод возвращает ASN.1-представление данных, на которые составлен запрос TS.	
`Метод get_attributes класса CMSRequest`	Метод возвращает ASN.1-представление атрибутов, указанных в запросе TS.	
`Метод put_timeStampAlgorithm класса CMSRequest`	Метод устанавливает строковое значение OID-идентификатора алгоритма TS, размещаемое в запросе MS TS.	
`Метод put_contentInfo класса CMSRequest`	Метод устанавливает ASN.1-представление данных, на которые формируется запрос MS TS.	
`Метод put_attributes класса CMSRequest`	Метод устанавливает ASN.1-представление атрибутов, на которые формируется запрос MS TS.	
`Конструктор класса CMSStamp`	Конструктор создаёт объект класса CMSStamp штампа времени Microsoft – CMS-сообщение с данными (на которые сформирован TS), значение TS (дату-время), сертификаты и поле «SignerInfo» с подписью.	

Метод encode класса CMSStamp	Метод выполняет ASN.1-кодирование сформированной и подписанной метки времени MS TS.	
Метод decode класса CMSStamp	Метод выполняет декодирование двоичного блоба с ASN.1-структурой сформированного штампа времени MS TS.	
Метод sign класса CMSStamp	Метод формирует подпись в формате CMS сообщения поверх метки времени MS TS (вложенного подписываемого атрибута) на данные, ранее размещённые внутри CMSStamp.	Разрешено формировать подпись только при выполнении следующих условий: 1) Контекст hCryptProv должен быть предварительно открыт функцией CryptAcquireCertificatePrivateKey. 2) В идентификаторе алгоритма хэш-функции должен содержаться один из следующих OID-идентификаторов: • szOID_CP_GOST_R3411_12_256 ("1.2.643.7.1.1.2.2"), • szOID_CP_GOST_R3411_12_512 ("1.2.643.7.1.1.2.3"). 3) В двоичном блобе должен содержаться сертификат ключа проверки ЭП алгоритма ГОСТ.
Метод verify класса CMSStamp	Метод проверяет CMS-подпись с использованием указанного сертификата.	Допускается использование только совместно с проверкой сертификата и всей соответствующей цепочки сертификатов.
Метод get_content класса CMSStamp	Метод возвращает двоичный блоб данных с подписанным/подписываемым содержимым CMS-сообщения.	
Метод get_signingTime класса CMSStamp	Метод возвращает значение метки времени MS TS в формате даты-времени (объект класса CryptoPro::CDateTime).	
Метод get_signerInfo класса CMSStamp	Метод возвращает декодированную ASN.1-структуру SignerInfo (объект класса CryptoPro::ASN1::CSignerInfo) содержащую данные о подписанте, атрибуты и значение подписи.	
Метод get_certificates класса CMSStamp	Метод возвращает список декодированных ASN.1-структур с Base64-кодированными сертификатами, вложенными в CMS-подпись под меткой времени MS TS.	

Метод put_content класса CMSStamp	Метод устанавливает подписываемое содержимое CMS-сообщения, под которым сформирована метка времени MS TS.	
Метод put_signingTime класса CMSStamp	Метод устанавливает подписываемую метку времени в CMS-сообщении.	
Метод put_certificates класса CMSStamp	Метод устанавливает сертификаты в CMS-сообщение.	
Конструктор класса CRequest	Конструктор создаёт объект запроса на штамп времени CRequest	Разрешается использование только конструктора по умолчанию и конструктора копирования.
Метод get_version класса CRequest	Метод возвращает версию запроса.	
Метод get_hashAlgorithm класса CRequest	Метод возвращает алгоритм хэша, использованного при вычислении хэш-функции от данных.	
Метод get_hashedMessage класса CRequest	Метод возвращает значение хэша данных, на которые запрашивается штамп времени.	
Метод get_certReq класса CRequest	Метод возвращает значение флага certReq (флага включения в штамп сертификата службы TSA).	
Метод get_reqPolicy класса CRequest	Метод возвращает запрашиваемую политику штампов времени.	
Метод get_nonce класса CRequest	Метод возвращает поле «Nonce» запроса на штамп времени.	
Метод get_extensions класса CRequest	Метод возвращает список расширений, вложенных в запрос.	
Метод put_reqPolicy класса CRequest	Метод устанавливает запрашиваемую политику штампов времени.	В качестве значений политик должны использоваться только определённые как допустимые для ИС, в которой применяется вызов.
Метод put_extensions класса CRequest	Метод вкладывает расширения в запрос на штамп времени.	
Метод encode класса CRequest	Метод кодирует запрос в ASN.1 DER-представление.	
Метод decode класса CRequest	Метод декодирует запрос из его закодированного ASN.1 DER-представления.	
Конструктор класса CResponse	Конструктор создаёт объект ответа на запрос штампа времени CResponse.	

Метод get_status класса CResponse	Метод возвращает статус ошибки, содержащийся в ответе службы TSA.	
Метод get_timeStampToken класса CResponse	Метод возвращает штамп времени, содержащийся в ответе службы TSA.	
Метод decode класса CResponse	Метод декодирует ответ на запрос штампа времени из его закодированного ASN.1 DER-представления.	
Конструктор класса CToken	Конструктор создаёт объект штампа времени CToken.	Разрешается использование только конструктора по умолчанию и конструктора копирования.
Метод get_version класса CToken	Метод возвращает версию штампа.	
Метод get_policy класса CToken	Метод возвращает политику, по которой выпущен штамп.	
Метод get_hashAlgorithm класса CToken	Метод возвращает алгоритм хэша, использованного при вычислении хэш-функции от данных.	
Метод get_hashedMessage класса CToken	Метод возвращает значение хэша данных, на которые выдан штамп времени.	
Метод get_serialNumber класса CToken	Метод возвращает серийный номер штампа времени.	
Метод get_genTime класса CToken	Метод возвращает время выдачи штампа.	
Метод get_genTimeZ класса CToken	Метод возвращает время выдачи штампа в виде строки в формате GeneralizedTime.	
Метод get_ordering класса CToken	Метод возвращает значение поля «ordering» (флага, определяющего, можно ли упорядочивать запросы, не учитывая поле «accuracy»).	
Метод get_signerInfo класса CToken	Метод возвращает ASN.1-структуру SignerInfo (информацию о подписчике).	
Метод get_accuracy класса CToken	Метод возвращает точность, с которой выпущен штамп времени.	
Метод get_nonce класса CToken	Метод возвращает поле «Nonce» запроса на штамп времени.	
Метод get_tsa класса CToken	Метод возвращает закодированное имя службы штампов времени.	
Метод get_extensions класса CToken	Метод возвращает список расширений, вложенных в штамп.	

Метод get_signedAttributes класса CToken	Метод возвращает список подписанных атрибутов, вложенных в штамп времени.	
Метод get_unsignedAttributes класса CToken	Метод возвращает список неподписанных атрибутов, вложенных в штамп времени.	
Метод get_certificates класса CToken	Метод возвращает сертификаты вложенные в штамп времени.	
Метод verify класса CToken	Метод декодирует штамп времени из его закодированного ASN.1 DER-представления.	Допускается использование только совместно с проверкой всей цепочки сертификатов для обрабатываемого объекта PKI.
Интерфейс клиентских приложений языка C++ (пространство имён CryptoPro::PKI::TSP::Client)		
Конструктор класса CRequest	Конструктор создаёт объект запроса на штамп времени CRequest.	При групповой политике UseNonceByDefault, установленной равной False, в полученном объекте запроса должен быть в обязательном порядке выставлен флаг включения в запрос поля «Nonce» вызовом put_useNonce.
Метод AddData класса CRequest	Метод добавляет очередной блок данных для поточного вычисления хэш-функции в рамках создания запроса на штамп времени.	
Метод ResetData класса CRequest	Метод сбрасывает информацию о хэше данных, накопленную после вызовов AddData.	
Метод Export класса CRequest	Метод экспортирует созданный запрос к службе штампов времени.	
Метод Import класса CRequest	Метод импортирует существующий запрос к службе штампов времени.	
Метод Display класса CRequest (только в Windows)	Метод отображает окно со свойствами запроса на штамп времени.	
Метод get_CertReq класса CRequest	Метод возвращает значение параметра запроса, согласно которому служба штампов включает свой сертификат в штамп или нет.	
Метод get_ClientCertificate класса CRequest	Метод возвращает сертификат TLS-клиента, который используется для подключения к службе TSA.	
Метод get_DataHash класса CRequest	Метод возвращает указанное в запросе значение хэша данных, на которые запрашивается штамп времени.	

Метод get_DataHashAlg класса CRequest	Метод возвращает ALG_ID алгоритма хэш-функции, вычисляемой от данных, на которые формируется запрос.	
Метод get_DataHashAlgOID класса CRequest	Метод возвращает строковое наименование алгоритма хэш-функции, вычисляемой от данных, на которые формируется запрос.	
Метод get_Flags класса CRequest	Метод возвращает дополнительные опции, используемые для работы с TSA службой.	
Метод get_HTTPStatus класса CRequest	Метод возвращает код статуса HTTP-запроса.	
Метод get_PolicyID класса CRequest	Метод возвращает идентификатор запрашиваемой политики службы штампов времени.	
Метод get_PropPage класса CRequest	Метод возвращает страницу свойств запроса на штамп времени.	
Метод get_ProxyAddress класса CRequest	Метод возвращает адрес прокси-сервера, используемого для подключения к службе TSA.	
Метод get_ProxyAuthType класса CRequest	Метод возвращает тип аутентификации перед прокси-сервером, используемым для подключения к службе TSA.	
Метод get_ProxyPassword класса CRequest	Метод возвращает пароль для аутентификации перед прокси-сервером, используемым для подключения к службе TSA.	
Метод get_ProxyUserName класса CRequest	Метод возвращает имя пользователя для аутентификации перед прокси-сервером, используемым для подключения к службе TSA.	
Метод get_Response класса CRequest	Метод возвращает данные, полученные от TSP-службы, в виде блоба. Доступен после вызова метода Send.	
Метод get_TSAAddress класса CRequest	Метод возвращает адрес TSA-сервера для подключения и отправки запроса на штамп времени.	
Метод get_TSAAuthType класса CRequest	Метод возвращает тип аутентификации перед службой штампов времени.	

Метод get_TSAPassword класса CRequest	Метод возвращает пароль для аутентификации перед службой штампов времени.	
Метод get_TSAUserName класса CRequest	Метод возвращает имя пользователя для аутентификации перед службой штампов времени.	
Метод get_UseNonce класса CRequest	Метод возвращает значение параметра запроса, определяющего необходимость использования «Nonce» в формируемом запросе штампа времени.	
Метод put_CertReq класса CRequest	Метод устанавливает значение параметра запроса, согласно которому служба штампов включает свой сертификат в штамп или нет.	
Метод put_ClientCertificate класса CRequest	Метод устанавливает сертификат TLS-клиента, который используется для подключения к службе TSA.	
Метод put_DataHashAlg класса CRequest	Метод устанавливает значение идентификатора алгоритма хэш-функции, вычисляемой от данных, на которые формируется запрос.	Разрешено использование только со следующими символьными аргументами: • szOID_CP_GOST_R3411_12_256 (CALG_GR3411_2012_256), • szOID_CP_GOST_R3411_12_512 (CALG_GR3411_2012_512)
Метод put_Flags класса CRequest	Метод устанавливает значение флага, влияющего на порядок отправки запроса.	
Метод put_PolicyID класса CRequest	Метод устанавливает запрашиваемую политику штампов времени.	В качестве значений политик должны использоваться только определённые как допустимые для ИС, в которой применяется вызов.
Метод put_ProxyAddress класса CRequest	Метод устанавливает адрес прокси-сервера, используемого при подключении к службе TSA.	
Метод put_ProxyAuthType класса CRequest	Метод устанавливает тип аутентификации перед прокси-сервером, используемом для подключения к службе TSA.	
Метод put_ProxyPassword класса CRequest	Метод устанавливает пароль для аутентификации перед прокси-сервером, используемым для подключения к службе TSA.	

Метод put_ProxyUserName класса CRequest	Метод устанавливает имя пользователя для аутентификации перед прокси-сервером, используемым для подключения к службе TSA.	
Метод put_TSAAddress класса CRequest	Метод устанавливает адрес TSA-сервера для подключения и отправки запроса на штамп времени.	
Метод put_TSAAuthType класса CRequest	Метод устанавливает тип аутентификации перед службой штампов времени.	
Метод put_TSAPassword класса CRequest	Метод устанавливает пароль для аутентификации перед службой штампов времени.	
Метод put_TSAUserName класса CRequest	Метод устанавливает имя пользователя для аутентификации перед службой штампов времени.	
Метод put_UseNonce класса CRequest	Метод устанавливает значение параметра запроса, определяющего необходимость использования «Nonce» в формируемом запросе штампа времени.	Разрешено использование только с параметром, значение которого равно True.
Метод Send класса CRequest	Метод отправляет запрос службе штампов времени и возвращает ответ в виде объекта CStamp после проверки на соответствие значений «Nonce» и, опционально, проверки подписи и сертификата службы TSA.	При использовании групповой политики UseNonceByDefault, установленной равной False, перед вызовом метода должен быть вызван put_UseNonce. В случае вызова с параметром Verify, равным False, необходимо выполнять проверку подписи в штампе времени с использованием сертификата службы TSA и с проверкой всех соответствующих объектов PKI.
Конструктор класса CStamp	Конструктор создаёт объект ответа на запрос штампа времени CStamp.	
Метод Export класса CStamp	Метод экспортирует созданный ответ на запрос к службе штампов времени.	
Метод Import класса CStamp	Метод импортирует существующий ответ на запрос к службе штампов времени.	
Метод Compare класса CStamp	Метод сравнивает с другим ответом на запрос к службе штампов времени по времени выпуска штампа.	

Метод Verify класса CStamp	Метод проверяет подпись штампа времени по вложенному сертификату и отсутствие неизвестных критических расширений.	Проверка штампа времени должна сопровождаться проверкой всех соответствующих подписи сервера TSA объектов PKI. Если штамп был загружен способом, отличным от вызова Import с указанием исходного запроса, то проверка штампа времени должна сопровождаться проверкой его соответствия запросу по значению «Nonce».
Метод VerifyCertificate класса CStamp	Метод проверяет подпись штампа времени по указанному или вложенному сертификату и отсутствие неизвестных критических расширений.	
Метод Display класса CStamp (только для Windows)	Метод отображает окно со свойствами штампа времени.	
Метод get_Certificates класса CStamp	Метод возвращает сертификаты, содержащиеся в штампе времени.	
Метод get_TSACertificate класса CStamp	Метод возвращает вложенный в штамп сертификат службы штампов времени.	
Метод get_FailInfo класса CStamp	Метод возвращает значение поля сведений об ошибке «FailInfo» ответа службы штампов времени.	
Метод get_Status класса CStamp	Метод возвращает числовое значение поля статуса «Status» ответа службы штампов времени.	
Метод get_StatusString класса CStamp	Метод возвращает строковое значение поля статуса «StatusString» ответа службы штампов времени.	
Метод get_DataHashAlg класса CStamp	Метод возвращает ALG_ID алгоритма хэш-функции, вычисляемой от данных, на которые сформирован ответ.	
Метод get_DataHashAlgOID класса CStamp	Метод возвращает строковое наименование алгоритма хэш-функции, вычисляемой от данных, на которые сформирован ответ.	
Метод get_DataHash класса CStamp	Метод возвращает указанное в штампе значение хэша данных, на которые сформирован штамп времени.	

Метод get_PolicyID класса CStamp	Метод возвращает идентификатор политики службы штампов времени, согласно которой штамп был сформирован.	
Метод get_SerialNumber класса CStamp	Метод возвращает серийный номер штампа времени.	
Метод get_Time класса CStamp	Метод возвращает время, зафиксированное в штампе.	
Метод get_Accuracy класса CStamp	Метод возвращает указанную службой штампов времени точность времени, фиксируемого в штампе.	
Метод get_Ordering класса CStamp	Метод возвращает значение поля «ordering» (флага, определяющего, что каждый выпущенный службой штамп имеет время строго больше ранее выпущенных штампов).	
Метод get_HasNonce класса CStamp	Метод возвращает флаг наличия в штампе поля «Nonce».	
Метод get_TSA класса CStamp	Метод возвращает имя службы штампов времени в строковом представлении.	
Метод get_TSABlob класса CStamp	Метод возвращает имя службы штампов времени в бинарном представлении.	
Метод get_DefaultAccuracy класса CStamp	Метод возвращает используемую в отсутствии флага «Ordering» по умолчанию точность времени, фиксируемого в штампе.	
Метод get_PropPage класса CStamp	Метод возвращает объект страницы свойств штампа времени.	
GetMSStamp	Метод создаёт, отправляет запрос к службе штампов времени Microsoft и возвращает полученный ответ в формате Blob.	Разрешено использование только с буфером Signature, полученным ранее из уже сформированного CMS-сообщения вызовом CryptMsgGetParam с параметром CMSG_ENCRYPTED_DIGEST. Получение данным вызовом CMS-подписи должно сопровождаться проверкой этой подписи и всех соответствующих ей объектов PKI.
FreeMSStamp	Метод освобождает память, занятую полученным с помощью GetMSStamp штампом времени.	

Интерфейс клиентских приложений КриптоПро TSP (COM) (объекты TSPCOM.*)		
Метод Import интерфейса ITSPRequest объекта TSPRequest	Метод импортирует в бинарном виде или в кодировке Base64 существующий запрос.	
Метод Export интерфейса ITSPRequest объекта TSPRequest	Метод экспортирует в бинарном виде созданный запрос к службе штампов времени.	
Метод Send интерфейса ITSPRequest объекта TSPRequest	Метод отправляет запрос службе штампов времени и возвращает ответ в виде объекта с интерфейсом ITSPStamp после проверки на соответствие значений «Nonce» и, опционально, проверки подписи и сертификата службы TSA.	При использовании групповой политики UseNonceByDefault, установленной равной False, перед вызовом метода свойство UseNonce должно быть задано равным True. В случае вызова с параметром Verify, равным VARIANT_FALSE, необходимо выполнять проверку подписи в штампе времени с использованием сертификата службы TSA и с проверкой всех соответствующих объектов PKI.
Метод Display интерфейса ITSPRequest объекта TSPRequest	Метод отображает окно со свойствами запроса на штамп времени.	
Свойство TSAAddress интерфейса ITSPRequest объекта TSPRequest	Свойство содержит в себе адрес TSA-сервера для подключения и отправки запроса на штамп времени.	
Свойство CertReq интерфейса ITSPRequest	Свойство содержит в себе значение параметра запроса, согласно которому служба штампов включает свой сертификат в штамп или нет.	
Свойство UseNonce интерфейса ITSPRequest объекта TSPRequest	Свойство содержит в себе значение параметра запроса, определяющего необходимость использования «Nonce» в формируемом запросе штампа времени.	Использование для изменения значения свойства разрешено только для изменения на VARIANT_TRUE.
Свойство PolicyID интерфейса ITSPRequest объекта TSPRequest	Свойство содержит в себе идентификатор запрашиваемой политики службы штампов времени.	В качестве значений политик должны использоваться только определённые как допустимые для ИС, в которой применяется вызов.
Свойство ProxyAddress интерфейса ITSPRequest объекта TSPRequest	Свойство содержит в себе адрес прокси-сервера, используемого для подключения к службе TSA.	
Свойство ProxyPassword интерфейса ITSPRequest объекта TSPRequest	Свойство содержит в себе пароль для аутентификации перед прокси-сервером, используемым для подключения к службе TSA.	

Свойство ProxyUserName интерфейса ITSPRequest объекта TSPRequest	Свойство содержит в себе имя пользователя для аутентификации перед прокси-сервером, используемым для подключения к службе TSA.	
Свойство ProxyAuthType интерфейса ITSPRequest объекта TSPRequest	Свойство содержит в себе тип аутентификации перед прокси-сервером, используемым для подключения к службе TSA.	
Свойство TSAUserName интерфейса ITSPRequest объекта TSPRequest	Свойство содержит в себе имя пользователя для аутентификации перед службой штампов времени.	
Свойство ClientCertificate интерфейса ITSPRequest объекта TSPRequest	Свойство содержит в себе контекст сертификата TLS-клиента, который используется для подключения к службе TSA.	
Свойство TSAAuthType интерфейса ITSPRequest объекта TSPRequest	Свойство содержит в себе тип аутентификации перед службой штампов времени.	
Свойство TSAPassword интерфейса ITSPRequest объекта TSPRequest	Свойство содержит в себе пароль для аутентификации перед службой штампов времени.	
Свойство HashValue интерфейса ITSPRequest объекта TSPRequest	Свойство содержит в себе указанное в запросе значение хэша данных, на которые запрашивается штамп времени (только чтение).	
Свойство HashAlgorithm интерфейса ITSPRequest объекта TSPRequest	Свойство содержит в себе строковый идентификатор алгоритма хэш-функции, вычисляемой от данных, на которые формируется запрос (только чтение).	
Свойство HTTPStatus интерфейса ITSPRequest объекта TSPRequest	Свойство содержит в себе код статуса HTTP-запроса (только чтение).	
Метод Import2 интерфейса ITSPRequest2 объекта TSPRequest	Метод импортирует в формате массива SAFEARRAY существующий запрос.	
Метод Export2 интерфейса ITSPRequest2 объекта TSPRequest	Метод экспортирует в формате массива SAFEARRAY созданный запрос к службе штампов времени.	
Метод AddData интерфейса ITSPRequest2 объекта TSPRequest	Метод добавляет очередной SAFEARRAY-блок данных для поточного вычисления хэш-функции в рамках создания запроса на штамп времени.	

Метод ResetData интерфейса ITSPRequest2 объекта TSPRequest	Метод сбрасывает информацию о хэше данных, накопленную после вызовов AddData, и позволяет снова их использовать.	
Свойство ClientCertificate2 интерфейса ITSPRequest2 объекта TSPRequest	Свойство содержит в себе сертификат TLS-клиента, который используется для подключения к службе TSA.	
Свойство HashAlgorithm2 интерфейса ITSPRequest2 объекта TSPRequest	Свойство содержит в себе строковый идентификатор алгоритма хэш-функции, вычисляемой от данных, на которые формируется запрос.	Использование для изменения значения свойства разрешено только для изменения на следующие символьные аргументы: • 1.2.643.7.1.1.2.2 (szOID_CP_GOST_R3411_12_256), • 1.2.643.7.1.1.2.3 (szOID_CP_GOST_R3411_12_512)
Метод Import интерфейса ITSPStamp объекта TSPStamp	Метод импортирует в бинарном виде или в кодировке Base64 существующий ответ на запрос к службе штампов времени.	
Метод Export интерфейса ITSPStamp объекта TSPStamp	Метод экспортирует в бинарном виде созданный ответ на запрос к службе штампов времени.	
Метод Compare интерфейса ITSPStamp объекта TSPStamp	Метод сравнивает с другим ответом на запрос к службе штампов времени по времени выпуска штампа.	
Метод VerifyStamp интерфейса ITSPStamp объекта TSPStamp	Метод проверяет подпись штампа времени по указанному или вложенному сертификату и отсутствие неизвестных критических расширений.	Проверка ЭП должна сопровождаться проверкой используемых объектов PKI (например, вызовом метода VerifyCertificate)
Метод VerifyCertificate интерфейса ITSPStamp объекта TSPStamp	Метод проверяет указанный или вложенный в штамп сертификат службы штампов времени (проверка значения подписи, сроков действия сертификата и ключей ЭП, проверка на отзыв и назначения ключа).	
Метод TimePart интерфейса ITSPStamp объекта TSPStamp	Метод возвращает компонент (года/месяцы/дни/минуты/секунды/мс/мкс) времени, зафиксированного в штампе.	
Метод Display интерфейса ITSPStamp объекта TSPStamp	Метод отображает окно со свойствами штампа времени.	
Свойство Certificates интерфейса ITSPStamp объекта TSPStamp	Свойство содержит в себе сертификаты, содержащиеся в штампе времени (только чтение).	

Свойство TSACertificate интерфейса ITSPStamp объекта TSPStamp	Свойство содержит в себе вложенный в штамп сертификат службы штампов времени (только чтение).	
Свойство FailInfo интерфейса ITSPStamp объекта TSPStamp	Свойство содержит в себе значение поля сведений об ошибке «FailInfo» ответа службы штампов времени (только чтение).	
Свойство StatusString интерфейса ITSPStamp объекта TSPStamp	Свойство содержит в себе строковое значение поля статуса «StatusString» ответа службы штампов времени (только чтение).	
Свойство Status интерфейса ITSPStamp объекта TSPStamp	Свойство содержит в себе числовое значение поля статуса «Status» ответа службы штампов времени (только чтение).	
Свойство PolicyID интерфейса ITSPStamp объекта TSPStamp	Свойство содержит в себе идентификатор политики службы штампов времени, согласно которой штамп был сформирован (только чтение).	
Свойство SerialNumber интерфейса ITSPStamp объекта TSPStamp	Свойство содержит в себе серийный номер штампа времени (только чтение).	
Свойство Ordering интерфейса ITSPStamp объекта TSPStamp	Свойство содержит в себе значение поля «ordering» – флага, определяющего, что каждый выпущенный службой штамп имеет время строго больше ранее выпущенных штампов (только чтение).	
Свойство TSAName интерфейса ITSPStamp объекта TSPStamp	Свойство содержит в себе имя службы штампов времени в строковом представлении (только чтение).	
Свойство Accuracy интерфейса ITSPStamp объекта TSPStamp	Свойство содержит в себе указанную службой штампов времени точность времени, фиксируемого в штампе (только чтение).	
Свойство Time интерфейса ITSPStamp объекта TSPStamp	Свойство содержит в себе время, зафиксированное в штампе (только чтение).	
Свойство HashValue интерфейса ITSPStamp объекта TSPStamp	Свойство содержит в себе указанное в штампе значение хэша данных, на которые сформирован штамп времени (только чтение).	

Свойство HashAlgorithm интерфейса ITSPStamp объекта TSPStamp	Свойство содержит в себе строковое наименование алгоритма хэш-функции, вычисляемой от данных, на которые сформирован ответ (только чтение).	
Свойство HasNonce интерфейса ITSPStamp объекта TSPStamp	Свойство содержит в себе флаг наличия в штампе поля «Nonce» (только чтение).	
Свойство DefaultAccuracy интерфейса ITSPStamp объекта TSPStamp	Свойство содержит в себе используемую в отсутствии флага Ordering по умолчанию точность времени, фиксируемого в штампе.	Разрешено использование только для чтения значения.
Метод Import2 интерфейса ITSPStamp2	Метод импортирует в формате массива SAFEARRAY существующий ответ на запрос к службе штампов времени.	
Метод Export2 интерфейса ITSPStamp2	Метод экспортирует в формате массива SAFEARRAY существующий ответ на запрос к службе штампов времени.	
Метод Verify интерфейса ITSPStamp2	Метод проверяет подпись штампа времени по указанному или вложенному сертификату и отсутствие неизвестных критических расширений (с указанием списка разрешённых критических расширений).	Проверка ЭП должна сопровождаться проверкой используемых объектов PKI (например, вызовом метода VerifyCertificate или VerifyCertificate2). Если штамп был загружен способом, отличным от вызова Import или Import2 с указанием исходного запроса, то проверка штампа времени должна сопровождаться проверкой его соответствия запросу по значению «Nonce».
Метод VerifyCertificate2 интерфейса ITSPStamp2	Метод проверяет указанный напрямую/через хранилище или вложенный в штамп сертификат службы штампов времени (проверка значения подписи, сроков действия сертификата и ключей ЭП, проверка на отзыв и назначения ключа).	
Интерфейс программы командной строки tsputil		

Команда stampinfo (si) утилиты tsputil	Команда выводит информацию о штампе времени, содержащемся в файле, и о результате проверки подписи в штампе и соответствующего сертификата ключа подписи службы TSA, а также о результате проверки соответствия исходным данным, на которые выдан штамп.	
Команда stampdisp (sd) утилиты tsputil	Команда выводит окно с информацией о штампе времени, содержащемся в файле.	
Команда reqinfo (ri) утилиты tsputil	Команда выводит информацию о запросе на штамп времени, содержащемся в файле.	
Команда license (li) утилиты tsputil	Команда выводит информацию о лицензии и позволяет ввести новый лицензионный номер.	
Команда reqdisp (rd) утилиты tsputil	Команда выводит окно с информацией о запросе на штамп времени, содержащемся в файле.	
Команда sendreq утилиты tsputil	Команда отправляет запрос на штамп в службу штампов времени и записывает полученный штамп в файл.	Файл запроса должен быть сформирован функциями интерфейса КриптоПро TSP SDK (например, командой makereq утилиты tsputil) с учётом ограничений их использования. Должно использоваться только с последующей проверкой сертификата службы TSA, подписавшей штамп времени, и проверкой самого штампа (например, с помощью команды stampinfo утилиты tsputil).

Команда makereq утилиты tsputil	Команда создаёт запрос на штамп времени и сохраняет его в файл.	Разрешено использование только при условии указания следующих символьных значений аргументов: • в качестве параметра alg: 1.2.643.7.1.1.2.2 (szOID_CP_GOST_R3411_12_256) или 1.2.643.7.1.1.2.3 (szOID_CP_GOST_R3411_12_512); • в качестве параметра input-type: INPUT_TYPE_DATA или INPUT_TYPE_SIGNATURE; при указании INPUT_TYPE_SIGNATURE необходимо обеспечить соответствие хэш-значения исходным данным, на которые запрашивается штамп; • в случае использования параметра nonce: значение «yes»; при использовании групповой политики UseNonceByDefault, установленной равной False, параметр nonce должен использоваться в обязательном порядке; • в случае использования параметра policy: значения политик, определённые как допустимые для ИС, в которой применяется вызов.
Команда makestamp утилиты tsputil	Команда получает штамп времени TSP: создаёт запрос на штамп времени, отправляет его в службу штампов и сохраняет полученный штамп в файл.	Разрешено использование только без указания параметра skipverify и при условии указания следующих символьных значений аргументов: • в качестве параметра alg: 1.2.643.7.1.1.2.2 (szOID_CP_GOST_R3411_12_256) или 1.2.643.7.1.1.2.3 (szOID_CP_GOST_R3411_12_512); • в качестве параметра input-type: INPUT_TYPE_DATA или INPUT_TYPE_SIGNATURE; при указании INPUT_TYPE_SIGNATURE необходимо обеспечить соответствие хэш-значения исходным данным, на которые запрашивается штамп; • в случае использования параметра nonce: значение «yes»; при использовании групповой политики UseNonceByDefault, установленной равной False, параметр nonce должен использоваться в обязательном порядке; • в случае использования параметра policy: значения политик, определённые как допустимые для ИС, в которой применяется вызов.

`Команда makemsstamp утилиты tsputil`	Команда получает штамп времени Microsoft: создаёт запрос на штамп времени, отправляет его в службу штампов и сохраняет полученный штамп в файл.	Разрешено использование только с запросом на базе CMS-сообщения, уже сформированного с помощью приведённых в данном перечне функций интерфейса КриптоПро CSP. Получение данным вызовом CMS-подписи должно сопровождаться проверкой этой подписи и всех соответствующих ей объектов PKI.
`Команда cgimstsa утилиты tsputil`	Приложение начинает работать как сценарий CGI и выдает штамп времени Microsoft. Команда принимает запрос на штамп времени со стандартного потока ввода программы и выдает в стандартный поток вывода созданный штамп времени.	Разрешено использование только со следующими символьными аргументами в качестве параметра alg: • 1.2.643.7.1.1.2.2 (szOID_CP_GOST_R3411_12_256); • 1.2.643.7.1.1.2.3 (szOID_CP_GOST_R3411_12_512).
`Команда extractcerts (ec) утилиты tsputil`	Команда копирует сертификаты из штампа TSP во внешний .p7b файл.	

Интерфейс КриптоПро ЭЦП SDK

Функция/метод/свойство	Описание	Ограничения на использование функции/метода/свойства
Низкоуровневый интерфейс КриптоПро ЭЦП SDK (C++)		
`CadesMsgEnhanceSignature`	Дополнение подписи CAdES-BES до форматов CAdES-T или CAdES-X Long Type 1 (в зависимости от заданного формата ЭП во входных параметрах) для указанной подписи в сообщении, переданном через дескриптор.	
`CadesMsgAddEnhancedSignature`	Добавление к подписанному сообщению ЭП формата CAdES-BES, CAdES-T или CAdES-X Long Type 1 (в зависимости от заданного формата ЭП во входных параметрах).	Разрешено использование только при указании параметра `pCadesCosignPara` с установленным флагом CADES_CHECK_CERT_REQ и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012. Для вычисления хэш-функции разрешается передача по указателю `pCadesCosignPara` структуры, содержащей только следующие возможные варианты значения OID-идентификаторов: • szOID_CP_GOST_R3411_12_256 ("1.2.643.7.1.1.2.2") • szOID_CP_GOST_R3411_12_512 ("1.2.643.7.1.1.2.3")
`CadesMsgIsType`	Проверка соответствия заданному типу CAdES-подписи.	
`CadesMsgIsTypeEncoded`	Проверка соответствия закодированной подписи заданному типу CAdES-подписи.	
`CadesMsgVerifySignature`	Проверка CAdES-подписи сообщения.	
`CadesMsgGetCertificateValues`	Получение сертификатов из доказательств подлинности CAdES-подписи, вложенных в сообщение.	
`CadesMsgGetCertificateValuesEx`	Получение сертификатов из доказательств подлинности CAdES-подписи, вложенных в ASN.1-структуру SignerInfo.	
`CadesMsgGetRevocationValues`	Возвращение доказательств проверки на отзыв (списки отозванных сертификатов и ответы службы OCSP), вложенных в сообщение.	

CadesMsgGetRevocationValuesEx	Возвращение доказательств проверки на отзыв (списки отозванных сертификатов и ответы службы OCSP), вложенных в ASN.1-структуру SignerInfo.	
CadesMsgOpenToEncode	Открытие дескриптора криптографического сообщения для формирования CAdES-подписи.	
CadesMsgUIDisplaySignature	Отображение окна свойств CAdES-подписи.	
CadesMsgGetCadesCTimestamps	Получение штампов времени на доказательствах подлинности подписи, вложенных в сообщение.	
CadesMsgGetCadesCTimestampsEx	Получение штампов времени на доказательствах подлинности подписи, вложенных в ASN.1-структуру SignerInfo.	
CadesMsgGetSignatureTimestamps	Получение штампов времени на CAdES-подписи, вложенных в сообщение.	
CadesMsgGetSignatureTimestampsEx	Получение штампов времени на CAdES-подписи, вложенных в ASN.1-структуру SignerInfo.	
CadesMsgGetSigningCertIdHashAlgEx	Получение алгоритма хэширования размещённого в ASN.1-структуре SignerInfo сертификата.	
CadesMsgGetSigningCertIdHashAlg	Получение алгоритма хэширования сертификата, определенного в signing-certificate, signing-certificate-v2 или other-signing-certificate.	
CadesMsgVerifyCountersignature-Encoded	Проверка CAdES-подписи, сформированной для вложенной в сообщение подписи.	

CadesMsgCountersign	Формирование CAdES-подписи для существующей подписи в сообщении.	Разрешено использование при указании массива rgCountersignersp с установленными флагами CADES_CHECK_CERT_REQ и заданными сертификатами ключа проверки ЭП по ГОСТ Р 34.10-2012. Для вычисления хэш-функции разрешается передача по указателю rgCountersignersp структуры, содержащей только следующие возможные варианты значения OID-идентификаторов: • szOID_CP_GOST_R3411_12_256 ("1.2.643.7.1.1.2.2") • szOID_CP_GOST_R3411_12_512 ("1.2.643.7.1.1.2.3")
CadesMsgConvertCreateContext	Создание контекста для преобразования отсоединённой CAdES-подписи в присоединённую.	
CadesMsgConvertUpdate	Передача сообщения для преобразования в прикреплённую CAdES-подпись.	
CadesMsgConvertFreeContext	Освобождение контекста преобразования в прикреплённую CAdES-подпись.	
CadesMsgGetSigningCertId	Получение идентификатора сертификата (находится в криптографическом сообщении), на соответствующем ключе которого была сформирована CAdES-подпись.	
CadesMsgGetSigningCertIdEx	Получение идентификатора сертификата (находится в ASN.1-структуре SignerInfo), на соответствующем ключе которого была сформирована CAdES-подпись.	
Интерфейс упрощенных функций языка C++		
Encrypt	Зашифрование сообщения в формате CMS с данными типа Enveloped Data (поддерживается только ГОСТ 28147-89).	
Decrypt	Расшифрование CMS-сообщения (поддерживается только ГОСТ 28147-89).	

`StreamEncrypt`	Функция пополняет поток зашифровываемых данных CMS-сообщения (поддерживается только ГОСТ 28147-89).	
`StreamDecrypt`	Функция пополняет поток расшифровываемых данных CMS-сообщения (поддерживается только ГОСТ 28147-89).	
`XadesSign`	Функция формирует подпись XML-документа в формате XAdES.	Разрешено использование при указании параметра `pSignPara` с установленным флагом XADES_CHECK_CERT_REQ и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012. Для вычисления хэш-функции разрешается передача по указателю `pSignPara` структуры, содержащей только следующие возможные варианты значения OID-идентификаторов: • szOID_CP_GOST_R3411_12_256 ("1.2.643.7.1.1.2.2") • szOID_CP_GOST_R3411_12_512 ("1.2.643.7.1.1.2.3")
`XadesVerify`	Проверка подписи XML документа в формате XAdES.	
`CadesSignHash`	Формирование открепленной CAdES-подписи по значению функции хэширования.	Разрешено использование при указании параметра `pSignPara` с установленным флагом CADES_CHECK_CERT_REQ и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012. Для вычисления хэш-функции разрешается передача по указателю `pSignPara` структуры, содержащей только следующие возможные варианты значения OID-идентификаторов: • szOID_CP_GOST_R3411_12_256 ("1.2.643.7.1.1.2.2") • szOID_CP_GOST_R3411_12_512 ("1.2.643.7.1.1.2.3")

CadesSignMessage	Формирование CAdES-подписи по значению исходного сообщения.	Разрешено использование при указании параметра pSignPara с установленным флагом CADES_CHECK_CERT_REQ и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012. Для вычисления хэш-функции разрешается передача по указателю pSignPara структуры, содержащей только следующие возможные варианты значения OID-идентификаторов: • szOID_CP_GOST_R3411_12_256 ("1.2.643.7.1.1.2.2") • szOID_CP_GOST_R3411_12_512 ("1.2.643.7.1.1.2.3")
CadesEnhanceMessage	Преобразование CAdES-BES подписи до формата CADES-X Long Type 1 для указанной подписи в сообщении, переданном в формате двоичных данных.	
CadesVerifyHash	Проверка открепленной CAdES-подписи по значению функции хэширования.	
CadesVerifyMessage	Проверка CAdES-подписи по значению исходного неподписанного сообщения.	
CadesUIDisplaySignatures	Отображение окна со списком присоединенных CAdES-подписей и их свойств.	
CadesViewSignatureDetached	Отображение окна со свойствами открепленной CAdES-подписи.	
Интерфейс вспомогательных функций C++		
CadesFreeBlob	Освобождение структуры CRYPT_DATA_BLOB, возвращаемой функциями при работе с CAdES-подписью.	
CadesFreeBlobArray	Освобождение типа CADES_BLOB_ARRAY, возвращаемого функциями при работе с CAdES-подписью.	
CadesFreeVerificationInfo	Освобождение структуры CADES_VERIFICATION_INFO, возвращаемой функциями при работе с CAdES-подписью.	

CadesFreeVerificationInfo	Освобождение списка указателей на структуру XADES_VERIFICATION_INFO, возвращаемую функциями при работе с XAdES-подписью.	
XadesFreeBlob	Освобождение структуры CRYPT_DATA_BLOB, возвращаемой функциями для работы с XAdES подписью.	
CadesFreeSignaturePropPages	Освобождение массива указателей на структуры PPROPSHEETPAGE, возвращаемого функциями отображения CAdES-подписи.	
Интерфейс клиентских приложений COM-объектов (CAdESCOM)		
Свойство Algorithm объекта HashedData	Алгоритм хэш-функции в контексте.	Разрешено использование только со следующими символьными аргументами: • CADESCOM_HASH_ALGORITHM_CP_GOST_3411; • CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_256; • CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_512. Объект хэша разрешено подавать в методы SignHash и CoSignHash объекта CadesSignedData только при установке алгоритма CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_256 или CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_512.
Свойство Value объекта HashedData	Результат хэш-функции (только чтение).	
Свойство DataEncoding объекта HashedData	Тип кодирования данных, передаваемых в хэш-функцию.	
Метод Hash класса HashedData	Вычисление хэш-функции от сообщения.	
Свойство Certificates объекта Store	Коллекция сертификатов в хранилище сертификатов (только чтение).	
Свойство Location объекта Store	Расположение хранилища сертификатов (только чтение).	
Свойство Name объекта Store	Наименование хранилища сертификатов (только чтение).	

Метод Open объекта Store	Открытие на запись или чтение хранилища сертификатов.	Полученный объект хранилища должен быть в обязательном порядке быть закрыт методом Close.
Метод Add объекта Store	Добавление контекста сертификата в открытое хранилище сертификатов.	
Метод AddCRL объекта Store	Добавление контекста CRL в открытое хранилище сертификатов.	
Метод Remove объекта Store	Удаление контекста сертификата из хранилища сертификатов.	
Метод Close объекта Store	Закрытие хранилища сертификатов.	
Свойство IssuerName объекта Certificate	Информация об издателе сертификата (только чтение).	
Свойство SerialNumber объекта Certificate	Информация о серийном номере сертификата (только чтение).	
Свойство SubjectName объекта Certificate	Информация об имени субъекта сертификата (только чтение).	
Свойство Thumbprint объекта Certificate	Информация об отпечатке сертификата (только чтение).	
Свойство ValidFromDate объекта Certificate	Информация о дате, с которой начинается срок действия сертификата (только чтение).	
Свойство ValidToDate объекта Certificate	Информация о дате, с которой прекращается срок действия сертификата (только чтение).	
Свойство Version объекта Certificate	Информация о версии сертификата (только чтение).	
Метод Export объекта Certificate	Экспорт сертификата в строковом формате.	
Метод Import объекта Certificate	Импорт сертификата в контекст из закодированной строки.	
Метод IsValid объекта Certificate	Проверка указанного сертификата и соответствующей ему цепочки сертификатов.	
Метод KeyUsage объекта Certificate	Возвращает объект с расширением использования ключа KeyUsage для сертификата.	
Метод PublicKey объекта Certificate	Объект открытого ключа PublicKey для сертификата.	
Свойство Content объекта CadesSignedData	Данные для подписи.	

Свойство ContentEncoding объекта CadesSignedData	Тип кодирования данных для подписи.	
Свойство Signers объекта CadesSignedData	Коллекция подписантов объекта подписи (только чтение).	
Свойство Certificates объекта CadesSignedData	Коллекция сертификатов в сообщении CAdES-подписи (только чтение).	
Метод SignCades объекта CadesSignedData	Формирование ЭП формата CAdES (CAdES-BES, CAdES-T или CAdES-XL Type 1) по значению исходного сообщения.	Разрешено использование только при указании параметра Signer с установленным в VARIANT_TRUE свойством CheckCertificate и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012.
Метод SignHash объекта CadesSignedData	Создание открепленной ЭП формата CAdES (CAdES-BES, CAdES-T или CAdES-XL Type 1) по контексту хэш-функции.	Разрешено использование только при указании параметра Signer с установленным в VARIANT_TRUE свойством CheckCertificate и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012. Разрешено использование только объекта хэша, полученного при установке алгоритма CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_256 или CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_512.
Метод CoSignHash объекта CadesSignedData	Добавление ЭП в открепленную подпись формата CAdES (CAdES-BES, CAdES-T или CAdES-XL Type 1) по контексту хэш-функции.	Разрешено использование только при указании параметра Signer с установленным в VARIANT_TRUE свойством CheckCertificate и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012. Разрешено использование только объекта хэша, полученного при установке алгоритма CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_256 или CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_512.
Метод VerifyHash объекта CadesSignedData	Проверка ЭП формата CAdES по контексту хэш-функции.	

Метод Sign объекта CadesSignedData	Создание ЭП формата CAdES-XL Type 1.	Разрешено использование только при указании параметра Signer с установленным в VARIANT_TRUE свойством CheckCertificate и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012. Разрешено использование только объекта хэша, полученного при установке алгоритма CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_256 или CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_512.
Метод CoSign объекта CadesSignedData	Добавление к сообщению дополнительной ЭП формата CAdES-XL Type 1.	Разрешено использование только при указании параметра Signer с установленным в VARIANT_TRUE свойством CheckCertificate и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012. Разрешено использование только объекта хэша, полученного при установке алгоритма CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_256 или CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_512.
Метод CoSignCades объекта CadesSignedData	Добавление ЭП формата CAdES (CAdES-BES, CAdES-T или CAdES-XL Type 1) в сообщение.	Разрешено использование только при указании параметра Signer с установленным в VARIANT_TRUE свойством CheckCertificate и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012. Разрешено использование только объекта хэша, полученного при установке алгоритма CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_256 или CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_512.
Метод EnhanceCades объекта CadesSignedData	Дополнение ЭП до ЭП формата CAdES-T, CAdES-X Long Type 1.	
Метод VerifyCades объекта CadesSignedData	Проверка ЭП формата CAdES (CAdES-BES, CAdES-T или CAdES-XL Type 1).	
Метод Verify объекта CadesSignedData	Проверка ЭП формата CAdES-XL Type 1.	
Свойство SignatureType объекта SignedXML	Тип электронной подписи формата XMLDSig (только запись).	

Свойство SignatureMethod объекта SignedXML	Алгоритм электронной подписи формата XMLDSig (только запись).	Разрешено использование только параметров, соответствующих ГОСТ Р 34.11-2012: • XmlDsigGost3410Url2012256; • XmlDsigGost3410Url2012512.
Свойство SignatureOptions объекта SignedXML	Свойства электронной подписи формата XMLDSig (только запись).	
Свойство DigestMethod объекта SignedXML	Алгоритм хэширования для формирования ЭП формата XMLDSig (только запись).	Разрешено использование только параметров, соответствующих ГОСТ Р 34.11-2012: • XmlDsigGost3411Url2012256; • XmlDsigGost3411Url2012512.
Свойство Content объекта SignedXML	Исходное сообщение (XML-документ) для подписи в формате XMLDSig.	
Свойство Signers объекта SignedXML	Коллекция подписантов объекта подписи формата XMLDSig (только запись).	
Метод Sign класса SignedXML	Формирование ЭП сообщения (XML-документа) в формате XMLDSig.	Разрешено использование только при указании параметра Signer с установленным в VARIANT_TRUE свойством CheckCertificate и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012.
Метод Verify класса SignedXML	Проверка ЭП сообщения (XML-документа) формата XMLDSig.	
Свойство Certificate объекта CPSigner	Сертификат ключа проверки ЭП.	Разрешено использование для изменения свойства только при указании сертификата ключа проверки ЭП по ГОСТ Р 34.10-2012.
Свойство TSAAddress объекта CPSigner	Адрес службы штампов времени.	
Свойство Options объекта CPSigner	Параметры сертификата ключа проверки ЭП.	
Свойство CRLs объекта CPSigner	Коллекция CRL (только чтение).	
Свойство OCSPResponses объекта CPSigner	Коллекция ответов OCSP-служб (только чтение).	
Свойство SignatureTimeStampTime объекта CPSigner	Время подписи из штампа времени на значение подписи (только чтение).	
Свойство SigningTime объекта CPSigner	Время подписи из атрибута signingTime (только чтение).	
Свойство UnauthenticatedAttributes объекта CPSigner	Коллекция неподписанных атрибутов (только чтение).	

Свойство AuthenticatedAttributes2 объекта CPSigner	Коллекция подписанных атрибутов (только чтение).	
Свойство SignatureStatus объекта CPSigner	Результат проверки электронной подписи (только чтение).	
Свойство CheckCertificate объекта CPSigner	Флаг проведения проверки сертификата пользователя и проверки соответствующей цепочки сертификатов.	
Свойство Count объекта CPSigners	Количество объектов CPSigner в коллекции (только чтение).	
Метод Item объекта CPSigners	Считывает элемент коллекции CPSigner по указанному индексу.	
Свойство Content объекта CPEnvelopedData	Исходное сообщение для шифрования в формате CMS.	
Свойство ContentEncoding объекта CPEnvelopedData	Тип кодирования данных для шифрования в формате CMS.	
Свойство Algorithm объекта CPEnvelopedData	Алгоритм шифрования в формате CMS (только чтение).	
Свойство Recipients объекта CPEnvelopedData	Сертификаты получателей зашифрованного в формате CMS сообщения (только чтение).	
Метод Encrypt объекта CPEnvelopedData	Шифрование сообщения в формате CMS (поддерживается только ГОСТ 28147-89).	
Метод Decrypt объекта CPEnvelopedData	Расшифрование CMS-сообщения (поддерживается только ГОСТ 28147-89).	
Метод StreamEncrypt объекта CPEnvelopedData	Метод пополняет поток зашифровываемых в формате CMS данных исходного сообщения (поддерживается только ГОСТ 28147-89).	
Метод StreamDencrypt объекта CPEnvelopedData	Метод пополняет поток расшифровываемых данных CMS-сообщения (поддерживается только ГОСТ 28147-89).	
Свойство OID объекта CPAttribute	Объектный идентификатор (OID) подписываемого атрибута подписи (только чтение).	
Свойство Value объекта CPAttribute	Значение подписываемого атрибута подписи в стороковом представлении.	

Свойство Name объекта CPAttribute	Наименование подписываемого атрибута подписи.	
Свойство ValueEncoding объекта CPAttribute	Способ кодирования значения подписываемого атрибута подписи.	
Свойство Count объекта CPAttributes	Количество объектов атрибутов подписи в коллекции (только чтение).	
Метод Item объекта CPAttributes	Получение объекта атрибута подписи в коллекции по заданному индексу.	
Метод Add объекта CPAttributes	Добавление объекта атрибута подписи в коллекцию.	
Метод Clear объекта CPAttributess	Удаление всех объектов атрибутов подписи из коллекции.	
Метод Remove объекта CPAttributes	Удаление объекта атрибута подписи из коллекции по заданному индексу.	
Свойство Count объекта Certificates	Количество объектов сертификатов в коллекции.	
Метод Item объекта Certificates	Получение объекта сертификата из коллекции по заданному индексу.	
Метод Find объекта Certificates	Поиск объекта сертификата в коллекции по заданному критерию.	
Свойство IssuerName объекта CRL	Издатель списка аннулированных сертификатов (только чтение).	
Свойство ThisUpdate объекта CRL	Дата выпуска списка аннулированных сертификатов (только чтение).	
Свойство NextUpdate объекта CRL	Дата следующего обновления списка аннулированных сертификатов (только чтение).	
Свойство Thumbprint объекта CRL	Отпечаток списка аннулированных сертификатов (только чтение).	
Свойство AuthKeyID объекта CRL	Идентификатор ключа УЦ, издавшего список аннулированных сертификатов (только чтение).	
Метод Import объекта CRL	Импорт списка аннулированных сертификатов.	
Метод Export объекта CRL	Экспорт списка аннулированных сертификатов.	
Свойство EKUs объекта ExtendedKeyUsage	Коллекция объектов EKU сертификата (только чтение).	
Свойство IsCritical объекта ExtendedKeyUsage	Флаг, является ли расширение EKU критическим (только чтение).	

Свойство IsPresent объекта ExtendedKeyUsage	Флаг, присутствует ли расширение EKU (только чтение).	
Свойство IsCritical объекта KeyUsage	Флаг, является ли расширение KeyUsage критическим (только чтение).	
Свойство IsCRLSignEnabled объекта KeyUsage	Флаг, установлен ли бит CRLSign в расширении KeyUsage (только чтение).	
Свойство IsDataEnciphermentEnabled объекта KeyUsage	Флаг, установлен ли бит dataEncipherment в расширении KeyUsage (только чтение).	
Свойство IsDecipherOnlyEnabled объекта KeyUsage	Флаг, установлен ли бит decipherOnly в расширении KeyUsage (только чтение).	
Свойство IsDigitalSignatureEnabled объекта KeyUsage	Флаг, установлен ли бит digitalSignature в расширении KeyUsage (только чтение).	
Свойство IsEncipherOnlyEnabled объекта KeyUsage	Флаг, установлен ли бит encipherOnly в расширении KeyUsage (только чтение).	
Свойство IsKeyAgreementEnabled объекта KeyUsage	Флаг, установлен ли бит keyAgreement в расширении KeyUsage (только чтение).	
Свойство IsKeyCertSignEnabled объекта KeyUsage	Флаг, установлен ли бит keyCertSign в расширении KeyUsage (только чтение).	
Свойство IsKeyEnciphermentEnabled объекта KeyUsage	Флаг, установлен ли бит keyEncipherment в расширении KeyUsage (только чтение).	
Свойство IsNonRepudiationEnabled объекта KeyUsage	Флаг, установлен ли бит nonRepudiationEnabled в расширении KeyUsage (только чтение).	
Свойство IsPresent объекта KeyUsage	Флаг, присутствует ли расширение KeyUsage (только чтение).	
Свойство Name объекта OID	Наименование OID.	
Свойство FriendlyName объекта OID	Наименование для отображения OID.	
Свойство Value объекта OID	Значение OID.	
Свойство Algorithm объекта PublicKey	OID алгоритма открытого ключа (только чтение).	

Свойство Length объекта PublicKey	Длина открытого ключа в битах (только чтение).	
Свойство EncodedKey объекта PublicKey	Значение открытого ключа (только чтение).	
Свойство EncodedParametrs объекта PublicKey	Параметры открытого ключа (только чтение).	
Свойство Count объекта Recipients	Число объектов Recipient в коллекции получателей криптографического сообщения (только чтение)	
Метод Item объекта Recipients	Получение объекта сертификат получателя криптографического сообщения из коллекции, соответствующий заданному индексу.	
Метод Add объекта Recipients	Добавление объекта сертификата в коллекцию сертификатов получателей криптографического сообщения.	
Метод Remove объекта Recipients	Удаление объекта сертификата из коллекции получателей криптографического сообщения.	
Метод Clear объекта Recipients	Удаление всех объектов сертификатов получателей криптографического сообщения из коллекции.	
Свойство IsValid объекта SignatureStatus	Результат проверки подписи (только чтение).	
Свойство MajorVersion объекта About	Старший компонент версии (только чтение).	
Свойство MinorVersion объекта About	Младший компонент версии (только чтение).	
Свойство BuildVersion объекта About	Номер сборки (только чтение).	
Свойство Version объекта About	Строковое представление версии (только чтение).	
Свойство PluginVersion объекта About	Объект, содержащий номер сборки, старший компонент версии, младший компонент версии (только чтение).	
Метод ProviderVersion объекта About	Получение строкового представления старшего компонента версии, младшего компонента версии и номера сборки.	
Метод CSPVersion объекта About	Получение объекта, содержащего номер сборки, старший компонент версии, младший компонент версии.	

Метод CSPName объекта About	Получение строкового представления наименования криптопровайдера.	
Свойство MajorVersion объекта Version	Старший компонент версии (только чтение).	
Свойство MinorVersion объекта Version	Младший компонент версии (только чтение).	
Свойство BuildVersion объекта Version	Номер сборки (только чтение).	
Метод toString объекта Version	Получение строкового представления номера версии.	
Интерфейс клиентских приложений COM-объектов (CertEnroll)		
Свойство ContainerName объекта CX509PrivateKey	Наименование контейнера закрытого ключа.	
Свойство UniqueContainerName объекта CX509PrivateKey	Уникальное имя контейнера (только чтение).	
Свойство Existing объекта CX509PrivateKey	Наличие закрытого ключа в контейнере.	
Свойство ExportPolicy объекта CX509PrivateKey	Ограничение на экспорт закрытого ключа.	
Свойство KeySpec объекта CX509PrivateKey	Назначение ключа: подпись, шифрование ключей.	
Свойство ProviderName объекта CX509PrivateKey	Наименование криптопровайдера.	Разрешено использование только со следующими символьными аргументами: • «Crypto-Pro GOST R 34.10-2012 Cryptographic Service Provider»; • «Crypto-Pro GOST R 34.10-2012 Strong Cryptographic Service Provider».
Свойство ProviderType объекта CX509PrivateKey	Тип криптопровайдера.	Разрешено использование только со следующими символьными аргументами (с обязательным заданием наименования криптопровайдера в свойстве ProviderName объекта CX509PrivateKey): • «80»; • «81».
Свойство HashAlgorithm объекта CX509CertificateRequestPkcs10	Алгоритм хэширования, используемый при подписи запроса на сертификат.	Разрешено устанавливать только следующие символьные аргументы: • «1.2.643.7.1.1.2.2» (ГОСТ Р 34.11-2012, 256 бит); • «1.2.643.7.1.1.2.3» (ГОСТ Р 34.11-2012, 512 бит).

Свойство Subject объекта CX509CertificateRequestPkcs10	Имя субъекта (CX500DistinguishedName).	
Свойство X509Extensions объекта CX509CertificateRequestPkcs10	Коллекция расширений (CX509Extensions) (только чтение).	
Метод InitializeFromPrivateKey объекта CX509CertificateRequestPkcs10	Инициализация объекта запроса на сертификат с использованием описания объекта закрытого ключа (CX509PrivateKey).	
Метод Encode объекта CX500DistinguishedName	Инициализация объекта по строке с именем субъекта.	
Метод InitializeEncode объекта CX509ExtensionKeyUsage	Инициализация объекта, задающего расширение KeyUsage.	
Метод InitializeEncode объекта CX509ExtensionAlternativeNames	Инициализация объекта, задающего расширение сертификата с коллекцией SAN.	
Свойство ItemByIndex объекта CAlternativeNames	Получение элемента по индексу из коллекции значений SAN (только чтение).	
Свойство Count объекта CAlternativeNames	Получение количества элементов в коллекции значений SAN (только чтение).	
Метод Add объекта CAlternativeNames	Добавление элемента в коллекцию значений SAN.	
Метод InitializeFromOtherName объекта CAlternativeName	Инициализация объекта, задающего SAN.	
Свойство CertificateDescription объекта CX509Enrollment	Описание сертификата.	
Свойство CertificateFriendlyName объекта CX509Enrollment	Имя сертификата для отображения.	
Метод CreateRequest объекта CX509Enrollment	Создает запрос на сертификат.	
Метод Initialize объекта CX509Enrollment	Инициализация объекта перед установкой полученного сертификата.	
Метод InitializeFromRequest объекта CX509Enrollment	Инициализация объекта с использованием объекта класса CX509CertificateRequestPkcs10, описывающего запрос на сертификат.	
Метод InstallResponse объекта CX509Enrollment	Устанавливает сертификат в хранилище сертификатов.	
Свойство Critical объекта CX509Extension	Задание данного расширения X509-объекта как критическое (только запись).	

Свойство ObjectId объекта CX509Extension	Объектный идентификатор данного расширения X509-объекта (только чтение).	
Метод Initialize объекта CX509Extension	Инициализация объекта расширения X509-объекта по объектному идентификатору CObjectId и значению расширения в кодировке DER.	
Метод InitializeEncode объекта CX509ExtensionCertificatePolicies	Инициализация объекта, задающего расширение X509-объекта CertificatePolicies.	
Метод InitializeEncode объекта CX509ExtensionEnhancedKeyUsage	Инициализация объекта, задающего расширение X509-объекта EnhancedKeyUsage.	
Свойство Count объекта CX509Extensions	Число элементов в коллекции расширений X509-объекта (только чтение).	
Метод ItemByIndex объекта CX509Extensions	Получение объекта расширения X509-объекта из коллекции по указанному индексу.	
Метод Add объекта CX509Extensions	Добавление объекта расширения X509-объекта в коллекцию.	
Метод InitializeEncode объекта CX509ExtensionSubjectSignTool	Инициализация объекта, задающего расширение X509-объекта SubjectSignTool.	
Метод InitializeEncode объекта CX509ExtensionTemplate	Инициализация объекта, задающего расширение X509-объекта Template.	
Свойство PolicyQualifiers объекта CCertificatePolicy	Коллекция CPolicyQualifiers, которые могут быть применены к политике сертификата (только чтение).	
Свойство ObjectId объекта CCertificatePolicy	Объектный идентификатор политики сертификата (только чтение).	
Метод Initialize объекта CCertificatePolicy	Инициализация объекта, задающего политику сертификата, определяющего цели использования сертификата.	
Свойство Count объекта CCertificatePolicies	Число объектов политики сертификата в коллекции (только чтение).	
Метод ItemByIndex объекта CCertificatePolicies	Получение объекта политики сертификата из коллекции с заданным номером.	

Метод Add объекта CCertificatePolicies	Добавляет объект политики сертификата в коллекцию.	
Свойство FriendlyName объекта CObjectId	Наименование объектного идентификатора (только чтение).	
Свойство Value объекта CObjectId	Строковое представление объектного идентификатора (только чтение).	
Метод InitializeFromAlgorithmName объекта CObjectId	Инициализация объекта по имени алгоритма.	
Метод InitializeFromValue объекта CObjectId	Инициализирует объект по идентификатору (OID).	
Свойство Count объекта CObjectIds	Число объектных идентификаторов в коллекции (только чтение).	
Метод ItemByIndex объекта CObjectIds	Получение объектного идентификатора из коллекции по заданному индексу.	
Метод Add объекта CObjectIds	Добавление объектного идентификатора в коллекцию.	

Интерфейс КриптоПро ЭЦП Browser plug-in

Метод/свойство	Описание	Ограничения на использование метода/свойства
Интерфейс CAdESCOM		
Свойство Algorithm объекта HashedData	Алгоритм хэш-функции в контексте.	Разрешено использование только со следующими символьными аргументами: • CADESCOM_HASH_ALGORITHM_CP_GOST_3411; • CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_256; • CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_512. Объект хэша разрешено подавать в методы SignHash и CoSignHash объекта CadesSignedData только при установке алгоритма CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_256 или CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_512.
Свойство Value объекта HashedData	Результат хэш-функции (только чтение).	
Свойство DataEncoding объекта HashedData	Тип кодирования данных, передаваемых в хэш-функцию.	
Метод Hash класса HashedData	Вычисление хэш-функции от сообщения .	
Свойство Certificates объекта Store	Коллекция сертификатов в хранилище сертификатов (только чтение).	
Свойство Location объекта Store	Расположение хранилища сертификатов (только чтение).	
Свойство Name объекта Store	Наименование хранилища сертификатов (только чтение).	
Метод Open объекта Store	Открытие на запись или чтение хранилища сертификатов.	Полученный объект хранилища должен быть в обязательном порядке быть закрыт методом Close.
Метод Add объекта Store	Добавление контекста сертификата в открытое хранилище сертификатов.	
Метод AddCRL объекта Store	Добавление контекста CRL в открытое хранилище сертификатов.	
Метод Remove объекта Store	Удаление контекста сертификата из хранилища сертификатов.	

Метод Close объекта Store	Закрытие хранилища сертификатов .	
Свойство IssuerName объекта Certificate	Информация об издателе сертификата (только чтение).	
Свойство SerialNumber объекта Certificate	Информация о серийном номере сертификата (только чтение).	
Свойство SubjectName объекта Certificate	Информация об имени субъекта сертификата (только чтение).	
Свойство Thumbprint объекта Certificate	Информация об отпечатке сертификата (только чтение).	
Свойство ValidFromDate объекта Certificate	Информация о дате, с которой начинается срок действия сертификата (только чтение).	
Свойство ValidToDate объекта Certificate	Информация о дате, с которой прекращается срок действия сертификата (только чтение).	
Свойство Version объекта Certificate	Информация о версии сертификата (только чтение).	
Метод Export объекта Certificate	Экспорт сертификата в строковом формате (только чтение).	
Метод Import объекта Certificate	Импорт сертификата в контекст из закодированной строки .	
Метод IsValid объекта Certificate	Проверка указанного сертификата и соответствующей ему цепочки сертификатов.	
Метод KeyUsage объекта Certificate	Возвращает объект с расширением использования ключа KeyUsage для сертификата.	
Метод PublicKey объекта Certificate	Объект открытого ключа PublicKey для сертификата.	
Свойство Content объекта CadesSignedData	Данные для подписи.	
Свойство ContentEncoding объекта CadesSignedData	Тип кодирования данных для подписи.	
Свойство Signers объекта CadesSignedData	Коллекция подписантов объекта подписи (только чтение).	
Свойство Certificates объекта CadesSignedData	Коллекция сертификатов в сообщении CAdES-подписи (только чтение).	

Метод SignCades объекта CadesSignedData	Формирование ЭП формата CAdES (CAdES-BES, CAdES-T или CAdES-XL Type 1) по значению исходного сообщения.	Разрешено использование только при указании параметра Signer с установленным в VARIANT_TRUE свойством CheckCertificate и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012.
Метод SignHash объекта CadesSignedData	Создание открепленной ЭП формата CAdES (CAdES-BES, CAdES-T или CAdES-XL Type 1) по контексту хэш-функции.	Разрешено использование только при указании параметра Signer с установленным в VARIANT_TRUE свойством CheckCertificate и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012. Разрешено использование только объекта хэша, полученного при установке алгоритма CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_256 или CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_512.
Метод CoSignHash объекта CadesSignedData	Добавление ЭП в открепленную подпись формата CAdES (CAdES-BES, CAdES-T или CAdES-XL Type 1) по контексту хэш-функции.	Разрешено использование только при указании параметра Signer с установленным в VARIANT_TRUE свойством CheckCertificate и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012. Разрешено использование только объекта хэша, полученного при установке алгоритма CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_256 или CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_512.
Метод VerifyHash объекта CadesSignedData	Проверка ЭП формата CAdES по контексту хэш-функции.	
Метод Sign объекта CadesSignedData	Создание ЭП формата CAdES-XL Type 1.	Разрешено использование только при указании параметра Signer с установленным в VARIANT_TRUE свойством CheckCertificate и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012. Разрешено использование только объекта хэша, полученного при установке алгоритма CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_256 или CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_512.

Метод CoSign объекта CadesSignedData	Добавление к сообщению дополнительной ЭП формата CAdES-XL Type 1.	Разрешено использование только при указании параметра Signer с установленным в VARIANT_TRUE свойством CheckCertificate и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012. Разрешено использование только объекта хэша, полученного при установке алгоритма CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_256 или CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_512.
Метод CoSignCades объекта CadesSignedData	Добавление ЭП формата CAdES (CAdES-BES, CAdES-T или CAdES-XL Type 1) в сообщение.	Разрешено использование только при указании параметра Signer с установленным в VARIANT_TRUE свойством CheckCertificate и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012. Разрешено использование только объекта хэша, полученного при установке алгоритма CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_256 или CADESCOM_HASH_ALGORITHM_CP_GOST_3411_2012_512.
Метод EnhanceCades объекта CadesSignedData	Дополнение ЭП до ЭП формата CAdES-T, CAdES-X Long Type 1.	
Метод VerifyCades объекта CadesSignedData	Проверка ЭП формата CAdES (CAdES-BES, CAdES-T или CAdES-XL Type 1).	
Метод Verify объекта CadesSignedData	Проверка ЭП формата CAdES-XL Type 1.	
Свойство SignatureType объекта SignedXML	Тип электронной подписи формата XMLDSig (только запись).	
Свойство SignatureMethod объекта SignedXML	Алгоритм электронной подписи формата XMLDSig (только запись).	Разрешено использование только параметров, соответствующих ГОСТ Р 34.11-2012: • XmlDsigGost3410Url2012256; • XmlDsigGost3410Url2012512.
Свойство SignatureOptions объекта SignedXML	Свойства электронной подписи формата XMLDSig (только запись).	
Свойство DigestMethod объекта SignedXML	Алгоритм хэширования для формирования ЭП формата XMLDSig (только запись).	Разрешено использование только параметров, соответствующих ГОСТ Р 34.11-2012: • XmlDsigGost3410Url2012256; • XmlDsigGost3410Url2012512.

Свойство Content объекта SignedXML	Исходное сообщение (XML-документ) для подписи в формате XMLDSig.	
Свойство Signers объекта SignedXML	Коллекция подписантов объекта подписи формата XMLDSig (только запись).	
Метод Sign класса SignedXML	Формирование ЭП сообщения (XML-документа) в формате XMLDSig.	Разрешено использование только при указании параметра Signer с установленным в VARIANT_TRUE свойством CheckCertificate и заданным сертификатом ключа проверки ЭП по ГОСТ Р 34.10-2012.
Метод Verify класса SignedXML	Проверка ЭП сообщения (XML-документа) формата XMLDSig.	
Свойство Certificate объекта CPSigner	Сертификат ключа проверки ЭП.	Разрешено использование для изменения свойства только при указании сертификата ключа проверки ЭП по ГОСТ Р 34.10-2012.
Свойство TSAAddress объекта CPSigner	Адрес службы штампов времени.	
Свойство Options объекта CPSigner	Параметры сертификата ключа проверки ЭП.	
Свойство CRLs объекта CPSigner	Коллекция CRL (только чтение).	
Свойство OCSPResponses объекта CPSigner	Коллекция ответов OCSP-служб (только чтение).	
Свойство SignatureTimeStampTime объекта CPSigner	Время подписи из штампа времени на значение подписи (только чтение).	
Свойство SigningTime объекта CPSigner	Время подписи из атрибута signingTime (только чтение).	
Свойство UnauthenticatedAttributes объекта CPSigner	Коллекция неподписанных атрибутов (только чтение).	
Свойство AuthenticatedAttributes2 объекта CPSigner	Коллекция подписанных атрибутов (только чтение).	
Свойство SignatureStatus объекта CPSigner	Результат проверки электронной подписи (только чтение).	
Свойство CheckCertificate объекта CPSigner	Флаг проведения проверки сертификата пользователя и проверки соответствующей цепочки сертификатов.	
Свойство Count объекта CPSigners	Количество объектов CPSigner в коллекции (только чтение).	

Метод Item объекта CPSigners	Считывает элемент коллекции CPSigner по указанному индексу.	
Свойство Content объекта CPEnvelopedData	Исходное сообщение для шифрования в формате CMS.	
Свойство ContentEncoding объекта CPEnvelopedData	Тип кодирования данных для шифрования в формате CMS.	
Свойство Algorithm объекта CPEnvelopedData	Алгоритм шифрования в формате CMS (только чтение).	
Свойство Recipients объекта CPEnvelopedData	Сертификаты получателей зашифрованного в формате CMS сообщения (только чтение).	
Метод Encrypt объекта CPEnvelopedData	Шифрование сообщения в формате CMS (поддерживается только ГОСТ 28147-89).	
Метод Decrypt объекта CPEnvelopedData	Расшифрование CMS-сообщения (поддерживается только ГОСТ 28147-89).	
Метод StreamEncrypt объекта CPEnvelopedData	Метод пополняет поток зашифровываемых в формате CMS данных исходного сообщения (поддерживается только ГОСТ 28147-89).	
Метод StreamDencrypt объекта CPEnvelopedData	Метод пополняет поток расшифровываемых данных CMS-сообщения (поддерживается только ГОСТ 28147-89).	
Свойство OID объекта CPAttribute	Объектный идентификатор (OID) подписываемого атрибута подписи (только чтение).	
Свойство Value объекта CPAttribute	Значение подписываемого атрибута подписи в стороковом представлении.	
Свойство Name объекта CPAttribute	Наименование подписываемого атрибута подписи.	
Свойство ValueEncoding объекта CPAttribute	Способ кодирования значения подписываемого атрибута подписи.	
Свойство Count объекта CPAttributes	Количество объектов атрибутов подписи в коллекции (только чтение).	
Метод Item объекта CPAttributes	Получение объекта атрибута подписи в коллекции по заданному индексу.	
Метод Add объекта CPAttributes	Добавление объекта атрибута подписи в коллекцию.	

Метод Clear объекта CPAttributess	Удаление всех объектов атрибутов подписи из коллекции.	
Метод Remove объекта CPAttributes	Удаление объекта атрибута подписи из коллекции по заданному индексу.	
Свойство Count объекта Certificates	Количество объектов сертификатов в коллекции.	
Метод Item объекта Certificates	Получение объекта сертификата из коллекции по заданному индексу.	
Метод Find объекта Certificates	Поиск объекта сертификата в коллекции по заданному критерию.	
Свойство IssuerName объекта CRL	Издатель списка аннулированных сертификатов (только чтение).	
Свойство ThisUpdate объекта CRL	Дата выпуска списка аннулированных сертификатов (только чтение).	
Свойство NextUpdate объекта CRL	Дата следующего обновления списка аннулированных сертификатов (только чтение).	
Свойство Thumbprint объекта CRL	Отпечаток списка аннулированных сертификатов (только чтение).	
Свойство AuthKeyID объекта CRL	Идентификатор ключа УЦ, издавшего список аннулированных сертификатов (только чтение).	
Метод Import объекта CRL	Импорт списка аннулированных сертификатов.	
Метод Export объекта CRL	Экспорт списка аннулированных сертификатов.	
Свойство EKUs объекта ExtendedKeyUsage	Коллекция объектов EKU сертификата (только чтение).	
Свойство IsCritical объекта ExtendedKeyUsage	Флаг, является ли расширение EKU критическим (только чтение).	
Свойство IsPresent объекта ExtendedKeyUsage	Флаг, присутствует ли расширение EKU (только чтение).	
Свойство IsCritical объекта KeyUsage	Флаг, является ли расширение KeyUsage критическим (только чтение).	
Свойство IsCRLSignEnabled объекта KeyUsage	Флаг, установлен ли бит CRLSign в расширении KeyUsage (только чтение).	
Свойство IsDataEnciphermentEnabled объекта KeyUsage	Флаг, установлен ли бит dataEncipherment в расширении KeyUsage (только чтение)	

Свойство IsDecipherOnlyEnabled объекта KeyUsage	Флаг, установлен ли бит decipherOnly в расширении KeyUsage (только чтение).	
Свойство IsDigitalSignatureEnabled объекта KeyUsage	Флаг, установлен ли бит digitalSignature в расширении KeyUsage (только чтение).	
Свойство IsEncipherOnlyEnabled объекта KeyUsage	Флаг, установлен ли бит encipherOnly в расширении KeyUsage (только чтение).	
Свойство IsKeyAgreementEnabled объекта KeyUsage	Флаг, установлен ли бит keyAgreement в расширении KeyUsage (только чтение).	
Свойство IsKeyCertSignEnabled объекта KeyUsage	Флаг, установлен ли бит keyCertSign в расширении KeyUsage (только чтение).	
Свойство IsKeyEnciphermentEnabled объекта KeyUsage	Флаг, установлен ли бит keyEncipherment в расширении KeyUsage (только чтение).	
Свойство IsNonRepudiationEnabled объекта KeyUsage	Флаг, установлен ли бит nonRepudiationEnabled в расширении KeyUsage (только чтение).	
Свойство IsPresent объекта KeyUsage	Флаг, присутствует ли расширение KeyUsage (только чтение).	
Свойство Name объекта OID	Наименование OID.	
Свойство FriendlyName объекта OID	Наименование для отображения OID.	
Свойство Value объекта OID	Значение OID.	
Свойство Algorithm объекта PublicKey	OID алгоритма открытого ключа (только чтение).	
Свойство Length объекта PublicKey	Длина открытого ключа в битах (только чтение).	
Свойство EncodedKey объекта PublicKey	Значение открытого ключа (только чтение).	
Свойство EncodedParametrs объекта PublicKey	Параметры открытого ключа (только чтение).	
Свойство Count объекта Recipients	Число объектов Recipient в коллекции получателей криптографического сообщения (только чтение).	

Метод Item объекта Recipients	Получение объекта сертификат получателя криптографического сообщения из коллекции, соответствующий заданному индексу.	
Метод Add объекта Recipients	Добавление объекта сертификата в коллекцию сертификатов получателей криптографического сообщения.	
Метод Remove объекта Recipients	Удаление объекта сертификата из коллекции получателей криптографического сообщения.	
Метод Clear объекта Recipients	Удаление всех объектов сертификатов получателей криптографического сообщения из коллекции.	
Свойство IsValid объекта SignatureStatus	Результат проверки подписи (только чтение).	
Свойство MajorVersion объекта About	Старший компонент версии (только чтение).	
Свойство MinorVersion объекта About	Младший компонент версии (только чтение).	
Свойство BuildVersion объекта About	Номер сборки (только чтение).	
Свойство Version объекта About	Строковое представление версии (только чтение).	
Свойство PluginVersion объекта About	Объект, содержащий номер сборки, старший компонент версии, младший компонент версии (только чтение).	
Метод ProviderVersion объекта About	Получение строкового представления старшего компонента версии, младшего компонента версии и номера сборки.	
Метод CSPVersion объекта About	Получение объекта, содержащего номер сборки, старший компонент версии, младший компонент версии.	
Метод CSPName объекта About	Получение строкового представления наименования криптопровайдера.	
Свойство MajorVersion объекта Version	Старший компонент версии (только чтение).	
Свойство MinorVersion объекта Version	Младший компонент версии (только чтение).	
Свойство BuildVersion объекта Version	Номер сборки (только чтение).	

Метод toString объекта Version	Получение строкового представления номера версии.	
Интерфейс CertEnroll		
Свойство ContainerName объекта CX509PrivateKey	Наименование контейнера закрытого ключа.	
Свойство UniqueContainerName объекта CX509PrivateKey	Уникальное имя контейнера (только чтение).	
Свойство Existing объекта CX509PrivateKey	Наличие закрытого ключа в контейнере.	
Свойство ExportPolicy объекта CX509PrivateKey	Ограничение на экспорт закрытого ключа.	
Свойство KeySpec объекта CX509PrivateKey	Назначение ключа: подпись, шифрование ключей.	
Свойство ProviderName объекта CX509PrivateKey	Наименование криптопровайдера.	Разрешено использование только со следующими символьными аргументами: • «Crypto-Pro GOST R 34.10-2012 Cryptographic Service Provider»; • «Crypto-Pro GOST R 34.10-2012 Strong Cryptographic Service Provider».
Свойство ProviderType объекта CX509PrivateKey	Тип криптопровайдера.	Разрешено использование только со следующими символьными аргументами (с обязательным заданием наименования криптопровайдера в свойстве ProviderName объекта CX509PrivateKey): • «80»; • «81».
Свойство HashAlgorithm объекта CX509CertificateRequestPkcs10	Алгоритм хэширования, используемый при подписи запроса на сертификат.	Разрешено устанавливать только следующие символьные аргументы: • «1.2.643.7.1.1.2.2» (ГОСТ Р 34.11-2012, 256 бит); • «1.2.643.7.1.1.2.3» (ГОСТ Р 34.11-2012, 512 бит).
Свойство Subject объекта CX509CertificateRequestPkcs10	Имя субъекта (CX500DistinguishedName).	
Свойство X509Extensions объекта CX509CertificateRequestPkcs10	Коллекция расширений (CX509Extensions) (только чтение).	
Метод InitializeFromPrivateKey объекта CX509CertificateRequestPkcs10	Инициализация объекта запроса на сертификат с использованием описания объекта закрытого ключа (CX509PrivateKey).	

Метод Encode объекта CX500DistinguishedName	Инициализация объекта по строке с именем субъекта.	
Метод InitializeEncode объекта CX509ExtensionKeyUsage	Инициализация объекта, задающего расширение KeyUsage.	
Метод InitializeEncode объекта CX509ExtensionAlternativeNames	Инициализация объекта, задающего расширение сертификата с коллекцией SAN.	
Свойство ItemByIndex объекта CAlternativeNames	Получение элемента по индексу из коллекции значений SAN (только чтение).	
Свойство Count объекта CAlternativeNames	Получение количества элементов в коллекции значений SAN (только чтение).	
Метод Add объекта CAlternativeNames	Добавление элемента в коллекцию значений SAN.	
Метод InitializeFromOtherName объекта CAlternativeName	Инициализация объекта, задающего SAN.	
Свойство CertificateDescription объекта CX509Enrollment	Описание сертификата.	
Свойство CertificateFriendlyName объекта CX509Enrollment	Имя сертификата для отображения.	
Метод CreateRequest объекта CX509Enrollment	Создает запрос на сертификат.	
Метод Initialize объекта CX509Enrollment	Инициализация объекта перед установкой полученного сертификата.	
Метод InitializeFromRequest объекта CX509Enrollment	Инициализация объекта с использованием объекта класса CX509CertificateRequestPkcs10, описывающего запрос на сертификат.	
Метод InstallResponse объекта CX509Enrollment	Устанавливает сертификат в хранилище сертификатов.	
Свойство Critical объекта CX509Extension	Задание данного расширения X509-объекта как критическое (только запись).	
Свойство ObjectId объекта CX509Extension	Объектный идентификатор данного расширения X509-объекта (только чтение).	
Метод Initialize объекта CX509Extension	Инициализация объекта расширения X509-объекта по объектному идентификатору CObjectId и значению расширения в кодировке DER.	

Метод InitializeEncode объекта CX509ExtensionCertificatePolicies	Инициализация объекта, задающего расширение X509-объекта CertificatePolicies.	
Метод InitializeEncode объекта CX509ExtensionEnhancedKeyUsage	Инициализация объекта, задающего расширение X509-объекта EnhancedKeyUsage.	
Свойство Count объекта CX509Extensions	Число элементов в коллекции расширений X509-объекта (только чтение).	
Метод ItemByIndex объекта CX509Extensions	Получение объекта расширения X509-объекта из коллекции по указанному индексу	
Метод Add объекта CX509Extensions	Добавление объекта расширения X509-объекта в коллекцию.	
Метод InitializeEncode объекта CX509ExtensionSubjectSignTool	Инициализация объекта, задающего расширение X509-объекта SubjectSignTool.	
Метод InitializeEncode объекта CX509ExtensionTemplate	Инициализация объекта, задающего расширение X509-объекта Template.	
Свойство PolicyQualifiers объекта CCertificatePolicy	Коллекция CPolicyQualifiers, которые могут быть применены к политике сертификата (только чтение).	
Свойство ObjectId объекта CCertificatePolicy	Объектный идентификатор политики сертификата (только чтение).	
Метод Initialize объекта CCertificatePolicy	Инициализация объекта, задающего политику сертификата, определяющего цели использования сертификата.	
Свойство Count объекта CCertificatePolicies	Число объектов политики сертификата в коллекции (только чтение).	
Метод ItemByIndex объекта CCertificatePolicies	Получение объекта политики сертификата из коллекции с заданным номером.	
Метод Add объекта CCertificatePolicies	Добавляет объект политики сертификата в коллекцию.	
Свойство FriendlyName объекта CObjectId	Наименование объектного идентификатора (только чтение).	
Свойство Value объекта CObjectId	Строковое представление объектного идентификатора (только чтение).	
Метод InitializeFromAlgorithmName объекта CObjectId	Инициализация объекта по имени алгоритма.	

Метод InitializeFromValue объекта CObjectId	Инициализирует объект по идентификатору (OID).	
Свойство Count объекта CObjectIds	Число объектных идентификаторов в коллекции (только чтение).	
Метод ItemByIndex объекта CObjectIds	Получение объектного идентификатора из коллекции по заданному индексу.	
Метод Add объекта CObjectIds	Добавление объектного идентификатора в коллекцию.	

Лист регистрации изменений

Лист регистрации изменений									
№ п/п	Номера листов (страниц)				Всего листов (страниц) в документе	№ документа	Входящий № сопроводительного документа и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					