

127018, Москва, ул. Сущёвский вал, д. 18
Телефон: +7 (495) 995 4820
Факс: +7 (495) 995 4820
<https://www.CryptoPro.ru>
E-mail: info@CryptoPro.ru



Средство Криптографической Защиты Информации	КриптоПро HSM версия 2.0 Комплектации 1 и 2 Правила пользования
---	--

ЖТЯИ.00096-01 95 01

Листов 76

2020 г.

Содержание

1.	Аннотация	3
2.	Назначение и область применения ПАКМ «КриптоПро HSM» версия 2.0	4
3.	Основные технические данные и характеристики ПАКМ.....	5
3.1.	Операционные системы серверных приложений ПАКМ.....	6
3.2.	Ключевые носители ПАКМ.....	6
4.	Аппаратная компонента ПАКМ	8
4.1.	Комплектация 1 Исполнение 1.....	8
4.2.	Комплектация 1 Исполнение 2.....	8
4.3.	Комплектация 1 Исполнение 3.....	9
4.4.	Комплектация 1 Исполнение 4.....	10
4.5.	Комплектация 1 Исполнение 5.....	11
5.	Состав компонент ПО ПАКМ	12
6.	Инструкция по размещению технических средств.....	13
7.	Требования к использованию аппаратных компонент ПАКМ	16
8.	Защита от НСД.....	17
8.1.	Принципы защиты информации от НСД	17
8.2.	Требования по защите от НСД	18
8.3.	Применяемая модель защиты	18
8.4.	Организационные меры защиты	20
8.5.	Организационно-технические меры защиты.....	22
8.6.	Ежесуточное нагрузочное тестирование ПАКМ	23
8.7.	Контроль целостности.....	25
8.8.	Электронный замок	26
8.9.	Защита от вскрытия корпуса ПАКМ	27
9.	Резервирование и восстановление ПАКМ	28
9.1.	Холодное резервирование ПАКМ	28
9.2.	Горячее резервирование ПАКМ.....	29
10.	Защита от вскрытия корпуса ПАКМ	33
11.	Эксплуатация ПАКМ	34
12.	Рекомендации по использованию ПАКМ.....	35
13.	Встраивание ПАКМ	36
	Приложение 1. Перечень вызовов, использование которых при разработке систем на основе ПАКМ «КриптоПро HSM» всех комплектаций с учетом п.1.5 Формуляра ЖТЯИ.00096-01 30 01 возможно без дополнительных тематических исследований	38

1. Аннотация

Данный документ содержит правила пользования программно-аппаратного модуля (ПАКМ) «КриптоПро HSM», его состав, назначение, инструкции по размещению и защиты от НСД.

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

Инструкции администраторам безопасности и пользователям различных автоматизированных систем, использующих ПАКМ «КриптоПро HSM» должны разрабатываться с учетом требований настоящих Правил.

2. Назначение и область применения

ПАКМ «КриптоПро HSM» версия 2.0

ПАКМ «КриптоПро HSM» версия 2.0 — программно-аппаратный криптографический модуль, предназначенный для использования в сетях/системах хранения и обработки информации, не составляющей государственной тайны.

ПАКМ «КриптоПро HSM» представляет собой сетевое устройство, подключаемое либо непосредственно к серверу (хосту), использующему криптографические сервисы ПАКМ, либо в сегмент локальной сети через стандартные сетевые устройства (коммутаторы, маршрутизаторы, концентраторы) для обслуживания групп серверов и компьютеров пользователей сети.

ПАКМ может быть использован в качестве СКЗИ в различных системах/подсистемах криптографической защиты информации, поддерживающих криптографические интерфейсы «КриптоПро CSP».

3. Основные технические данные и характеристики ПАКМ

ПАКМ «КриптоПро HSM» предназначен для выполнения следующих функций:

- формирования/проверки электронной подписи (ЭП) под блоком данных по запросу пользователей;
- шифрования/расшифрования блоков данных по запросам пользователей.

При этом ПАКМ «КриптоПро HSM» обеспечивает:

- реализацию криптографических функций и интерфейс взаимодействия с серверами и рабочими станциями пользователей;
- интерфейс к прикладным криптографическим функциям в соответствии со спецификацией Microsoft Cryptographic Service Provider;
- возможность использования функций ПАКМ «КриптоПро HSM» через интерфейсы Microsoft CryptoAPI;
- идентификацию и аутентификацию пользователей при локальном и удаленном доступе к ПАКМ «КриптоПро HSM»;
- проверку целостности критичного к безопасному функционированию ПО при инициализации ПАКМ «КриптоПро HSM»;
- генерацию случайных чисел с использованием аппаратного ДСЧ;
- генерацию ключей ЭП и шифрования с использованием исходного материала, предоставленного уполномоченной организацией;
- срок действия ключей ЭП, являющихся неэкспортируемыми, составляет не более 3-х лет. Максимальный срок действия ключей проверки ЭП — 15 лет после окончания срока действия соответствующего ключа ЭП. Максимальный срок действия открытых ключей обмена — не более 3-х лет. Максимальный срок действия неэкспортируемых закрытых ключей обмена составляет не более 3-х лет. Срок действия иных ключей не превышает 1 года 3 месяцев¹;
- сопряжение с сервером/серверной группой по отдельному сегменту Ethernet;
- запись сгенерированных ключей на интеллектуальные карты;
- ввод закрытого ключа/ключа ЭП с ключевых носителей на интеллектуальной карте;
- выполнение функций создания и проверки электронной подписи (ЭП) согласно ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012 (ГОСТ 34.10-2018), вычисление хэш-функции согласно ГОСТ Р 34.11-94 и ГОСТ Р 34.11-2012 (ГОСТ 34.11-2018);
- шифрование и имитозащита согласно ГОСТ 28147-89;
- опционально, поддержку алгоритмов SHA1, RSA, 3DES в части генерации ключей, формирования и проверки ЭП;
- возможность встречной работы ПАКМ «КриптоПро HSM» с ПЭВМ, поддерживающим интерфейс «КриптоПро CSP»;
- уничтожение ключей;

- сопряжение с устройством доступа по криптографически защищенному каналу «K2». Канал «K2» — локальный защищенный канал, используется для защищенного обмена информацией между ПАКМ и устройством доступа к ПАКМ;
- сопряжение с устройством доступа по криптографически защищенному каналу «K». Канал «K» — локальный защищенный канал, используется для защищенного обмена информацией между ПАКМ и устройством доступа к ПАКМ²;
- регистрация событий в журнале аудита криптографических вызовов ПАКМ.

Примечания:

1. 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 ПАКМ, на выполнение предъявленных к ПАКМ требований по ТЗ, согласованному с 8 Центром ФСБ России.

2. Канал «K» используется только в рамках Головного удостоверяющего центра.

3.1. Операционные системы серверных приложений ПАКМ

Программные средства ПАКМ «КриптоПро HSM» функционируют на базе ОС Альт Линукс СПТ 7.0 с защитой ядра ОС средствами пакета GRSecurity.

Альт Линукс СПТ 7.0 Сертифицирован Федеральной службой по техническому и экспортному контролю (ФСТЭК России). Классификация по уровню контроля отсутствия недекларированных возможностей — 4 уровень. Показатели защищенности от несанкционированного доступа к информации — по 5 классу защищенности.

ПАКМ предназначен для использования с серверными приложениями и приложениями пользователей на базе операционных систем Unix/Linux и Windows, включая 64-разрядные их исполнения (список операционных систем приведен в документе «ЖТЯИ.00096-01 30 01. КриптоПро HSM. Формуляр»). На технических средствах ПЭВМ, имеющего подключенное СКЗИ ПАКМ «КриптоПро HSM», должна быть обеспечена антивирусная защита программных компонентов ПАКМ и СФ.

3.2. Ключевые носители ПАКМ

Ключевая система ПАКМ «КриптоПро HSM» включает в себя ключи ЭП, шифрования и обмена (экспорта ключей).

Ключи ЭП представляются ключевой парой: ключ ЭП — для формирования ЭП, ключ проверки ЭП.

Ключи шифрования: симметричные ключи сообщения (пакета), случайные или диверсифицированные из случайного ключа сессии по открытому заголовку сообщения (пакета).

Ключи обмена строятся на основе открытого распределения ключей по алгоритму Диффи-Хеллмана на базе ключевых пар закрытый/открытый ключи алгоритма ГОСТ Р 34.10-2012 (ГОСТ 34.10-2018).

На ключевых носителях ключи хранятся в формате ключевого контейнера. Ключевой контейнер содержит также информацию, необходимую для обеспечения криптографической защиты ключей и их целостности.

Закрытые ключи, хранящиеся в памяти ПАКМ, шифруются прямо или косвенно (через промежуточные ключи – ключи шифрования) на ключе активации ПАКМ.

Ключ активации ПАКМ использует схему разделения секрета с вводом ключевой информации с любых 3-х (k) из 5-ти (n) носителей для формирования функционального ключа. При этом обеспечивается защита функционального ключа при компрометации ключевой информации на любых не более k-1 носителях. В случае компрометации ключевой информации хотя бы с одного носителя, необходимо перевыпустить все 5 (n) ключей.

Интеллектуальные карты поставляются с ПАКМ «КриптоПро HSM» отформатированными, с предустановленным pin-кодом «11111111». При записи ключей на карту pin-код необходимо сменить.

Новый pin-код должен выбираться случайным образом и состоять из 8 цифр от «0» до «9».

Периодичность плановой смены pin-кода составляет 1 месяц.

При выпуске/записи карт с ключами на карту наносится тип карты: номер компоненты ключа защиты, символ «K» (для карты аутентификации Администратора UNIX/Linux Сервера – ПАК «КриптоПро HSM»), символ «K2» (для карты аутентификации Пользователь/Администратор Windows сервера – ПАК «КриптоПро HSM»). Кроме этого на карту записывается фамилия лица (пользователя-владельца), ответственного за данный ключевой носитель.

Надписи производятся разборчивым почерком (предпочтительно печатными буквами), фломастером типа Staedtler Lumocolor permanent № 318 (с водостойким красителем).

Описание ключевой системы и ключевых носителей представлено в документе «ЖТЯИ.00096-01 90 01. КриптоПро HSM. Инструкция по использованию».

4. Аппаратная компонента ПАКМ

4.1. Комплектация 1 Исполнение 1

В качестве аппаратной платформы ПАКМ «КриптоПро HSM» версия 2.0 Комплектация 1 Исполнение 1 (класс защиты KB/KB2) используется серверная платформа Advantech. Ее состав отображен в таблице 4.1.

Таблица 4.1. Состав аппаратной платформы ПАКМ «КриптоПро HSM» версия 2.0. Комплектация 1 Исполнение 1

Наименование	Модель	Описание
Корпус	ACP-2010MB	Серверный промышленный корпус Advantech для монтажа в стойку 19" высота 2U, глубиной 485 мм, с двумя вентиляторами охлаждения, смонтированными внутри.
Блок питания	DPS-500AB-9B	Advantech, Модель AC-120B Rev:01, 500 W двойной с горячей заменой
Материнская плата	ASMB-823I	Advantech, 19A6823I01-01 Rev.A1, Dual LGA 2011-R3
Процессоры	E5-2620V3	INTEL XEON E5-2620V3 2.4 GHz, гнездо LGA2011-3 (2 шт.)
ОЗУ	8G 1Rx4 DDR4 2133R	Модуль памяти AQD-D4U8GR21-HZ (4 шт.)
Вентиляторы	AVC DS06025B12U	Система охлаждения 1960055362N001 LGA2011 (2 шт.)
Флеш-диски	SQFlash 4GB	SATA DOM 530 4G SLC 4CH, SQF-SDMS4-4G-J6C (3 шт.)
LCD-панель	ЭЛКО SAPIС-E	Двухстрочный дисплей (2x20 символов, 4 кнопки управления, интерфейс RS-232).
Считыватель карт	CardMan 3921	OMNIKEY CardMan 3921 Reader Board (интерфейс USB)
Электронный замок	ПАК Соболев	ООО «Код Безопасности»
Сетевая карта	AT-29 Series SX/LC	Allied Telesis, двойной (1 шт.)
Считыватель iButton	iButton Probe	Контактная площадка идентификатора ключа ЭЗ «Соболев»
Защитная панель	P-PANEL	Защитная панель на задней части корпуса с креплением изнутри

4.2. Комплектация 1 Исполнение 2

В качестве аппаратной платформы ПАКМ «КриптоПро HSM» версия 2.0 Комплектация 1 Исполнение 2 (класс защиты КС3) используется серверная платформа AdvantiX модель IS-201849 (аналогично ПАКМ «КриптоПро HSM» версия 1.0 Исполнение 2 Модификация 1). Ее состав отображен в таблице 4.2.

Таблица 4.2. Состав аппаратной платформы ПАКМ «КриптоПро HSM» версия 2.0. Комплектация 1 Исполнение 2

Наименование	Модель	Описание
Корпус	RMC-2F-0-0-00L	Серверный промышленный корпус для монтажа в стойку 19" высота 2U, с дополнительным охлаждением FAN-8032-BR 80x80x32мм
Блок питания	R2G-5800V/EPS	Двойной блок питания мощностью 800 Вт каждый блок.
Материнская плата	DBS2600CP2	Серверная материнская плата INTEL DBS2600CP2
Процессоры	Хеон E5-2630 V2	Центральный процессор Intel Xeon E5-2630 V2, гнездо LGA2011 (2 шт.)
ОЗУ	PC12800	Модуль памяти 4GB PC12800 DDR3/ ECC REG KINGSTONE (8 шт.)
Вентиляторы	BXSTS200C	Thermal Solution (Combo) BXSTS200C (2 шт.)
Флеш-диски	D150QV	SATADOM 2GB, D150QV (DESIH-02GJ30AC1DB) (3 шт.)
LCD-панель	SAPIC-E	Двухстрочный дисплей (2x20 символов, 4 кнопки управления, интерфейс RS-232). Производство ООО «ЭЛКО Технологии СПб»
Считыватель карт	CardMan 3121 / PC Twin Reader	OMNIKEY CardMan 3121 Reader Board (интерфейс USB) Gemalto PC Twin Reader (интерфейс USB)
Электронный замок	ПАК Соболь	ООО «Код Безопасности»
Оптическая сетевая карта	EXPI9400PFBLK	Intel E1G42EFBLK897904
Считыватель iButton	iButton Probe	Контактная площадка для элемента энергонезависимой памяти (ключа ЭЗ Соболь)
Защитная панель	P-PANEL_1849	Защитная панель для IS-201849
Корпус	RMC-2F-0-0-00L	Серверный промышленный корпус для монтажа в стойку 19" высота 2U, с дополнительным охлаждением FAN-8032-BR 80x80x32мм

4.3. Комплектация 1 Исполнение 3

В качестве аппаратной платформы ПАКМ «КриптоПро HSM» версия 2.0 Комплектация 1 Исполнение 3 (класс защиты КС3) используется серверная платформа Аxiomtec модель AX61222TB (аналогично ПАКМ «КриптоПро HSM» версия 1.0 Исполнение 1). Ее состав отображен в таблице 4.3.

Таблица 4.3. Состав аппаратной платформы ПАКМ «КриптоПро HSM» версия 2.0. Комплектация 1 Исполнение 3

Наименование	Модель	Описание
Корпус	AX61222TB	Серверный промышленный корпус для монтажа в стойку 19" высота 2U, глубиной 450 мм с двумя вентиляторами охлаждения 1208BA DYNAEON смонтированными на передней панели.
Блок питания	FSP300-	FSP-300-60ATV(PFC)(RC) (ATX 300 W) - 1 шт.

	60ATV(PF)	140 x 150 x 86 mm
Объединительная шина	FAB105-V1 Rev: B0-RC	Стандарт PICMG 1.3 для корпусов 2U, 1 слот PICMG 1.3, 3 слота PCI (производитель Axiomtec).
Процессорная плата	SHB100VGGA	Стандарт PICMG 1.3, процессорное гнездо LGA775 (производитель Axiomtec).
Процессор	Core 2 Duo 6420	Двухъядерный 64 разрядный процессор с ядром Сонгое, тактовая частота ядра 2,13 ГГц, внешней шины 1066 МГц FSB, гнездо PLGA775 (1 шт.)
Вентилятор процессора	Nidec F09A- 12B1S2	Intel D60188-002 FC813331 1A011Q200 DC 12V 0,14A Faxconn
Оперативная память	KVR667	Kingstone KVR 667MHz DDR2 Non-ECC CL5/2GB (2 шт.)
Флеш-диски	D150QV 4GB	SATADOM 4Гбайт, вертикальный, серия D150QV, SLC (3 шт.)
LCD-панель	SAPIC-E	Двухстрочный дисплей (2x20 символов, 4 кнопки управления, интерфейс RS-232) Производство ООО «ЭЛКО Технологии СПб»
Считыватель карт	CardMan 3121 / PC Twin Reader	OMNIKEY CardMan 3121 Reader Board (интерфейс USB) Gemalto PC Twin Reader (интерфейс USB)
Электронный замок	ПАК Соболь	ООО «Код Безопасности»
Сетевая карта	DGE-550SX	D-Link DGE-550SX PCI-X 1000BASE-SX (SC) (2 шт.)
Считыватель iButton	iButton Probe	Контактная площадка для элемента энергонезависимой памяти (ключа ЭЗ Соболь)
Защитная панель		Защитная панель, блокирующая доступ к интерфейсам процессорной платы.

4.4. Комплектация 1 Исполнение 4

В качестве аппаратной платформы ПАКМ «КриптоПро HSM» версия 2.0 Комплектация 1 Исполнение 4 используется серверная платформа AdvantiX модель IS-201849 (аналогично ПАКМ «КриптоПро HSM» версия 1.0 Исполнение 2). Ее состав отображен в таблице 4.4.

Таблица 4.4. Состав аппаратной платформы ПАКМ «КриптоПро HSM» версия 2.0. Комплектация 1 Исполнение 4

Наименование	Модель	Описание
Корпус	RMC-2F-0-0-00L	Серверный промышленный корпус для монтажа в стойку 19" высота 2U, с дополнительным охлаждением FAN-8032-BR 80x80x32мм
Блок питания	R2W-6500P/EP5	Двойной блок питания мощностью 500 Вт каждый блок.
Материнская плата	S5500BC	INTEL BLUFF/CREEK S5500BC
Процессоры	Xeon E5540 / Xeon E5630	Центральный процессор Intel Xeon E5540 или Intel Xeon E5630, (2 шт.)
ОЗУ	PC12800	Модуль памяти 4GB PC12800 DDR3/ ECC REG KINGSTONE (8 шт.)

Вентиляторы	STS100C	Intel STS100C (2 шт.)
Флеш-диски	D150QV	SATADOM 2GB, D150QV (DESIH-02GJ30AC1DB) (3 шт.)
LCD-панель	SAPIC-E	Двухстрочный дисплей (2x20 символов, 4 кнопки управления, интерфейс RS-232). Производство ООО «ЭЛКО Технологии СПб»
Считыватель карт	CardMan 3121 / PC Twin Reader	OMNIKEY CardMan 3121 Reader Board (интерфейс USB) / Gemalto PC Twin Reader (интерфейс USB)
Электронный замок	ПАК Соболев	ООО «Код Безопасности»
Оптическая сетевая карта	EXPI9402PFBLK	Intel EXPI9402PFBLK
Считыватель iButton	iButton Probe	Контактная площадка для элемента энергонезависимой памяти (ключа ЭЗ Соболев)
Защитная панель	P-PANEL_1849	Защитная панель для IS-201849
Корпус	RMC-2F-0-0-00L	Серверный промышленный корпус для монтажа в стойку 19" высота 2U, с дополнительным охлаждением FAN-8032-BR 80x80x32мм

4.5. Комплектация 1 Исполнение 5

В качестве аппаратной платформы ПАКМ «КриптоПро HSM» версия 2.0 Комплектация 1 Исполнение 5 (класс защиты КС3) используются серверные платформы производства ООО «КРИПТО-ПРО».

Для ограничения возможности влияния аппаратных компонентов СВТ на функционирование СКЗИ необходимо проведение исследований ПО BIOS СВТ, на которых установлено СКЗИ, на соответствие действующим требованиям ФСБ России по исследованию ПО BIOS СВТ.

5. Состав компонент ПО ПАКМ

Программная компонента ПАКМ включает две составляющие:

- системное программное обеспечение — ОС «Альт Линукс СПТ 7.0»;
- программное обеспечение ПАКМ (ПО ПАКМ).

Штатные средства ОС «Альт Линукс СПТ 7.0» включают:

- МЭ — межсетевой экран;
- TCP/IP — стек протокола TCP/IP;
- RPC — процедура удаленных вызовов;
- SYS Log — модуль работы с log-файлами;
- ACL — система управления доступом;
- Драйвер COM-порта;
- Библиотека libc, обеспечивающая взаимодействие прикладного программного обеспечения с программными средствами ОС.

Состав устанавливаемых пакетов ОС «Альт Линукс СПТ 7.0» при изготовлении ПАКМ и настройка ОС описаны в документе «ЖТЯИ.00096-01 94 01. КриптоПро HSM. Описание процедуры сборки».

Программное обеспечение ПАКМ включает:

- «ЖТЯИ.00096-01 99 01. КриптоПро HSM. Базовые модули»;
- «ЖТЯИ.00096-01 99 02. КриптоПро HSM. Интерфейсные модули».

Базовые модули ПАКМ являются составной частью изделия ПАКМ и устанавливаются при изготовлении ПАКМ «КриптоПро HSM».

Интерфейсные модули поставляются на отдельном CDROM и служат для создания систем, использующих ПАКМ «КриптоПро HSM», подсистем криптографической защиты информации. Данные модули устанавливаются на ЭВМ, использующей ПАКМ «КриптоПро HSM».

В комплект поставки ПАКМ «КриптоПро HSM» входит CDROM «ЖТЯИ.00096-01 99 02. КриптоПро HSM. Интерфейсные модули».

На данном CDROM содержится необходимое ПО для платформ Windows, LSB-совместимых 32/64-разрядных ОС семейства Unix/Linux, которое устанавливается на ПЭВМ, использующую СКЗИ ПАКМ «КриптоПро HSM».

Описание интерфейсных модулей для ОС семейства Unix/Linux и их применения можно в документе «ЖТЯИ.00096-01 90 02. КриптоПро HSM. Использование интерфейсных модулей».

Описание интерфейсных модулей для ОС семейства Windows и их применения можно найти в документе «ЖТЯИ.00096-01 93 01. КриптоПро HSM. Руководство пользователя».

6. Инструкция по размещению технических средств

При размещении технических средств, имеющих в своем составе ПАКМ «КриптоПро HSM» (Таблица 4.1 «ЖТЯИ.00096-01 30 01. КриптоПро HSM. Формуляр»), следует руководствоваться требованиями и рекомендациями, приведенными ниже.

При эксплуатации ПАКМ необходимо обеспечить контролируемую зону¹ радиусом не менее 4 метров при работе с конфиденциальной информацией без применения дополнительных мер спецазащиты.

Примечание 1: Под контролируемой зоной здесь и далее понимается пространство (территория, здание, часть здания), в котором исключено неконтролируемое пребывание лиц, не имеющих постоянного или разового допуска, и посторонних транспортных средств.

Вспомогательные технические средства и системы (радиопередатчики, телефонные аппараты, аппаратура оперативно-командной связи, модемы, датчики пожарной и охранной сигнализации, батареи центрального отопления и др. оборудование), имеющие цепи, выходящие за пределы контролируемой зоны, необходимо располагать от ПАКМ на расстоянии не менее 0,7 м.

Кабели связи телефонных аппаратов, аппаратуры связи, модемов, посторонние провода и другие токопроводящие коммуникации, выходящие за пределы контролируемой зоны, или подключаемые к радиопередающим средствам, необходимо располагать от изделия ПАКМ на расстоянии не менее 0,1 м.

Должны быть приняты меры по исключению несанкционированного доступа в помещения, в которых установлен ПАКМ, посторонних лиц, по роду своей деятельности не являющихся персоналом, допущенным к работе в этих помещениях.

Входные двери режимных помещений должны быть оборудованы замками, гарантирующими надежное закрытие помещений в нерабочее время.

Окна и двери должны быть оборудованы охранной сигнализацией, связанной с пультом централизованного наблюдения за сигнализацией.

Электропитание ПАКМ от сети должно осуществляться одним из следующих способов:

- От трансформаторной подстанции, размещенной со всеми низковольтными цепями в пределах контролируемой зоны;
- Через сетевые помехоподавляющие фильтры, обеспечивающие затухание сигналов не менее 50 дБ в диапазоне частот 100-1000 МГц;
- От автономных источников электропитания (бензо/дизель-генераторов, аккумуляторов, источников бесперебойного питания (отключенных от сети)), расположенных в одной контролируемой зоне с изделием ПАКМ и не имеющих цепей, выходящих за пределы контролируемой зоны.

Фильтры должны располагаться от изделия ПАКМ на расстоянии не менее 0,7 м. Заземление фильтров должно производиться на контур заземления, расположенный в пределах контролируемой зоны.

Заземление изделия должно быть осуществлено на контур заземления, размещенный в пределах контролируемой территории. При эксплуатации изделия ПАКМ должен быть исключен посторонний, за исключением штатного, гальванический контакт изделия с шиной заземления.

Размещение оборудования, технических средств, предназначенных для обработки конфиденциальной информации, должно соответствовать требованиям техники безопасности, санитарным нормам, а также требованиям пожарной безопасности.

Внутренняя планировка и расположение рабочих мест в помещениях должны обеспечивать исполнителям работ сохранность доверенных им конфиденциальных документов и сведений.

По окончании рабочего дня помещения закрываются и опечатываются. Помещения с опечатанными входными дверями сдаются под охрану отделу безопасности или дежурному по предприятию (по установленному порядку) с указанием времени приема-сдачи с отметкой о включении и выключении охранной сигнализации в журнале учета.

Сдачу ключей и помещений под охрану, также получение ключей и вскрытие помещений производят сотрудники, работающие в этих помещениях, по утвержденному руководством учреждения списку с образцами подписей этих сотрудников, который находится у охраны или у дежурного по учреждению.

Перед вскрытием помещений должна быть проверена целостность оттисков печатей и исправность замков. При обнаружении нарушения целостности оттисков печатей, повреждения замков или других признаков, указывающих на возможное проникновение в эти помещения посторонних лиц, помещение не вскрывается, а о случившемся немедленно ставится в известность руководство и отдел безопасности.

В случае утраты ключа от входной двери помещения немедленно ставится в известность отдел безопасности учреждения.

На случай пожара, аварии или стихийного бедствия должны быть разработаны специальные инструкции, утвержденные руководством учреждения, в которых предусматривается порядок вызова администрации, должностных лиц, вскрытие помещений, очередность и порядок спасения конфиденциальных документов и дальнейшего их хранения.

Запрещается использовать в помещении, в котором размещен ПАКМ, радиотелефоны и другую радиоаппаратуру.

Запрещается размещение изделия ПАКМ в выделенных помещениях, где циркулирует речевая акустическая информация, содержащая сведения, составляющие государственную тайну или конфиденциального характера.

В случае планирования размещения средств ЭП в помещениях, в которых отсутствует речевая акустическая и визуальная информация, содержащая сведения, составляющие государственную тайну, и не установлены АС и системы приема, передачи, обработки, хранения и отображения информации, содержащей сведения, составляющие государственную тайну, решение о проведении проверок АС иностранного производства, входящих в состав средств ЭП класса KB2, принимается организацией, обеспечивающей эксплуатацию данных средств ЭП.

При эксплуатации изделия ПАКМ запрещается вносить изменения в состав, конструкцию, электрическую и монтажную схему ПАКМ.

Допускается организовать нешифрованный и/или не аутентифицированный канал K2s только при соблюдении требований по безопасности, включающих организационные меры по размещению ПАКМ и сервера в одной серверной стойке.

ПЭВМ, на которые устанавливается «КриптоПро HSM Client», должны быть допущены установленным порядком для обработки информации ограниченного доступа и иметь соответствующие Предписания на эксплуатацию.

Каналы связи ПЭВМ с «КриптоПро HSM» и/или «КриптоПро HSM Client», выходящие за пределы контролируемой зоны, должны быть защищены одним из следующих способов:

- Применением волоконно-оптических линий связи;
- Применением оптических развязывающих устройств (медиаконвертеров), устанавливаемых в канал передачи информации для создания оптоволоконного фрагмента сети;
- Применением специальных сертифицированных канальных СКЗИ для передачи информации по каналам связи.

При этом запрещается использование радиопередающих интерфейсов для передачи зашифрованной информации без проведения оценки защищенности радиоканала.

В случае планирования размещения ПЭВМ с установленным СКЗИ в помещениях, где присутствует речевая акустическая и/или визуальная информация, содержащая сведения, составляющие государственную тайну или конфиденциального характера, должны быть проведены специальные исследования и специальная проверка ПЭВМ в установленном порядке.

7. Требования к использованию аппаратных компонент ПАКМ

Для обеспечения безопасного функционирования ПАКМ «КриптоПро HSM» для всех его экземпляров необходимо обеспечить:

- отсутствие возможности бесконтрольного (несанкционированного) вскрытия персоналом корпуса с целью изменения состава аппаратных и программных средств.

Вскрытие корпуса допускается только в присутствии представителя организации Производителя ПАКМ;

- регулярный контроль за состоянием печатывающих наклеек;
- регулярный контроль за состоянием всех кабельных соединений;
- исключение организационными мерами нештатное и несанкционированное использование ПАКМ;

- осуществление электропитания ПАКМ через защищенную электрическую сеть, например, через ИБП (источник бесперебойного питания) с гальванической развязкой;

- строгое соблюдение инструкции по размещению технических средств (см. разд. 6 настоящего документа).

8. Защита от НСД

СКЗИ ПАКМ «КриптоПро HSM», обеспечивает защиту конфиденциальной информации, не содержащей сведений, составляющих государственную тайну, от внешнего и внутреннего нарушителя, осуществляющего создание способов и подготовку атак с привлечением специалистов, имеющих опыт разработки и анализа криптосредств. Привилегированные пользователи, имеющие доступ в контролируруемую зону, осуществляющие техническое обслуживание, настройку, конфигурирование ПАКМ и управление ключевой системой, относятся к потенциальным нарушителям. Возможность сговора между данными пользователями исключается.

Ролевая модель доступа к функциям ПАКМ

ПАКМ ЖТЯИ.00096-01 разработан с учетом того, что привилегированные пользователи ПАКМ (члены группы администраторов, имеющие доступ в контролируемую зону) могут являться потенциальными нарушителями. При этом возможность сговора между ними исключается.

Данное требование реализовано с использованием ролевой модели доступа к различным функциям ПАКМ. Это означает, что каждому отдельному члену административной группы дается доступ только к строго определенному набору административных функций, не позволяющих провести успешную атаку на получение контроля над ключами пользователей, хранящимися в ПАКМ «КриптоПро HSM».

Программное обеспечение ПАКМ «КриптоПро HSM» различает следующие роли:

- обычный пользователь СКЗИ ПАКМ «КриптоПро HSM»;
- администратор сервера, сервисы которого используют СКЗИ ПАКМ «КриптоПро HSM»;
- администратор ПАКМ «КриптоПро HSM»;
- аудитор ПАКМ «КриптоПро HSM»;
- администратор резервного копирования ПАКМ «КриптоПро HSM»;
- суперпользователь ПАКМ «КриптоПро HSM».

Признак того, что пользователю назначена та или иная роль хранится в сертификате ключа доступа к функциям ПАКМ, как специальное расширение (Extended Key Usage) сертификата. Доступ к ПАКМ (локальный или удаленный) осуществляется только с использованием данного сертификата ключа доступа и самого ключа.

Ключи и сертификат доступа к ПАКМ формируются ПАКМ и выдаются обычным пользователям администратором ПАКМ. Ключи и сертификат доступа к ПАКМ для привилегированных пользователей формируются ПАКМ и выдаются суперпользователем ПАКМ.

Подробнее ролевая модель доступа описана в п. 5 документа «ЖТЯИ.00096-01 91 01. КриптоПро HSM. Руководство администратора безопасности».

8.1. Принципы защиты информации от НСД

Защита информации от НСД в автоматизированной системе обеспечивается комплексом программно-технических средств и поддерживающих их организационных мер. В их числе:

- применение специальных программно-аппаратных средств защиты;
- организация системы контроля безопасности информации;

- физическая охрана ПЭВМ и ее средств;
- администрирование информационной безопасности, основанное на разделении ролей административной группы;
- учет носителей информации;
- сигнализация о попытках нарушения защиты;
- периодическое тестирование технических и программных средств защиты;
- использование сертифицированных программных и технических средств.

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.

Защита информации от НСД должна предусматривать контроль эффективности средств защиты от НСД. Этот контроль может быть либо периодическим, либо инициироваться по мере необходимости пользователем или контролирующими органами.

В организации, эксплуатирующей ПО ПАКМ, должна быть выпущена инструкция по защите от НСД к системе, разработанная на базе настоящего документа, руководящих документов Гостехкомиссии (ФСТЭК России), действующих нормативных документов самой эксплуатирующей организации.

В организации - пользователе системы должно быть выделено специальное должностное лицо - администратор безопасности, функции которого должны заключаться в выполнении процедур установки ПО, настройки системного окружения, установки, настройки, обслуживания и обеспечения функционирования средств защиты.

Администратор безопасности должен иметь возможность доступа ко всей информации, обрабатываемой на рабочем месте, кроме конфиденциальной.

Каждый исполнитель работ как пользователь сети конфиденциальной связи должен быть зарегистрирован у администратора.

Должны быть приняты меры, исключая возможность воздействия нарушителя на ПАКМ по каналам связи, выходящим за пределы контролируемой зоны.

8.2. Требования по защите от НСД

ПАКМ «КриптоПро HSM» должен соответствовать требованиям по условиям применения – документ «ЖТЯИ.00096-01 90 01. КриптоПро HSM. Инструкция по использованию».

Ремонт и сервисное обслуживание ПАКМ осуществляется в организациях, имеющих Лицензию ФСТЭК России с правом проведения работ по ремонту и сервисному обслуживанию средств защиты информации и средств информатизации в защищенном исполнении. При эксплуатации ПАКМ запрещается внесение изменений в состав компонент ПАКМ. После ремонта при необходимости проводится специальная проверка и специальные исследования отремонтированного оборудования.

8.3. Применяемая модель защиты

Субъектами, связанными с функционированием Серверов/рабочих станций, взаимодействующих с ПАКМ «КриптоПро HSM», и потенциально имеющими возможность осуществить НСД, являются:

- персонал службы безопасности - держатели ключей;

- привилегированные пользователи ПАКМ: администратор ПАКМ, аудитор ПАКМ, администратор резервного копирования ПАКМ (осуществляют установку, настройку ПАКМ «КриптоПро HSM» и поддержку его функционирования);
- администратор безопасности (осуществляет настройку подсистемы безопасности серверов приложений (например, Удостоверяющего центра) и поддержку организационных, организационно-технических и технических мер обеспечения безопасности);
- операторы ППО Серверов/рабочих станций;
- персонал, допущенный к работе на Сервере/рабочей станции;
- технический персонал информационной системы, не допущенный к работе на ПАКМ «КриптоПро HSM», на Серверах и рабочих станциях учреждения;
- пользователи ЛВС;
- пользователи глобальной сети.

Для защиты от НСД с учетом перечисленных потенциальных нарушителей должны использоваться следующие средства и меры:

- шифрование всей ключевой системы ПАКМ на ключах шифрования ПАКМ, которые, в свою очередь, зашифрованы на ключе активации ПАКМ, разделенном по схеме 3 из 5-ти. Защитные ключи с разделенными частями секретов формируются на смарт-картах и распределяются между отдельными привилегированными пользователями;
- парольная защита ключевого носителя на смарт-карте;
- парольная система входа в ОС администратора системы, администратора безопасности, оператора. Пароль входа в систему должен удовлетворять следующим требованиям: длина пароля не менее 6 символов, среди символов пароля встречаются заглавные символы, прописные символы, цифры и специальные символы, срок смены пароля не реже одного раза в месяц;
- штатные средства разграничения доступа ОС Серверов и рабочих станций и встроенной ОС ПАКМ;
- электронный замок «Соболь» для защиты от несанкционированного входа в систему;
- ролевое разграничение доступа к ПАКМ со стороны привилегированных пользователей ПАКМ;
- аутентификация на ПАКМ «КриптоПро HSM» с использованием системы аутентификации карты ОСКАР (карты канала К);
- аутентификация на ПАКМ «КриптоПро HSM» с использованием системы аутентификации протокола TLS (реализация канала К2) с использованием ключей обмена на ключевых носителях в виде смарт-карт Магистра на базе микроконтроллера ST23L80A, ОСКАР, USB-устройств (eToken, ruToken);
- локальная аутентификация привилегированных пользователей ПАКМ (из группы администраторов ПАКМ) с использованием системы аутентификации смарт-карты Магистра на базе микроконтроллера ST23L80A, ОСКАР (карты канала К2);

- штатные средства ОС ПАКМ - для защиты от воздействий со стороны Серверов/рабочих станций с целью НСД по каналам К, К2;
- межсетевой экран четвертого класса для защиты от воздействия по глобальной сети.

8.4. Организационные меры защиты

В данном разделе представлены основные рекомендации по организационным мерам защиты для обеспечения безопасности функционирования Серверов/рабочих станций (ПЭВМ), имеющих подключение к ПАКМ «КриптоПро HSM».

Правом доступа к ПЭВМ должны обладать только лица, прошедшие соответствующую подготовку. Администратор безопасности должен ознакомить каждого пользователя с документацией на ПАКМ «КриптоПро HSM» и с другими нормативными документами, созданными на ее основе.

Администратором безопасности должно быть проведено опечатывание системного блока ПЭВМ, исключающее возможность несанкционированного изменения аппаратной части.

При каждом включении ПЭВМ необходимо проверять сохранность печатей системного блока и разъемов.

Администратор безопасности должен периодически (не реже 1 раза в 2 месяца) проводить контроль целостности и легальности установленных копий ПО на ПЭВМ с помощью программ контроля целостности.

В случае обнаружения «посторонних» (не зарегистрированных) программ, нарушения целостности программного обеспечения, либо выявления факта повреждения печатей на системном блоке работа ПЭВМ должна быть прекращена. По данному факту должно быть проведено служебное расследование службой информационной безопасности организации-владельца ПЭВМ и организованы работы по анализу и ликвидации негативных последствий данного нарушения.

Пользователь должен запускать только те приложения, которые разрешены администратором безопасности.

ПО, установленное на ПЭВМ, не должно иметь встроенных средств разработки и отладки программ.

Пароли, назначаемые пользователям, должны отвечать требованиям соответствующих инструкций и нормативных документов.

На технических средствах ПЭВМ должно использоваться только лицензионное программное обеспечение фирм-производителей.

Необходимо исключить попадание в систему программ, позволяющих, пользуясь ошибками ОС, получать привилегии администратора.

Должны быть приняты меры по исключению несанкционированного доступа посторонних лиц в помещения, в которых установлены технические средства ПЭВМ, по роду своей деятельности не являющихся персоналом, допущенным к работе в указанных помещениях.

Из состава системы должно быть исключено все оборудование, которое может создавать угрозу безопасности ОС. Также следует избегать использования любых нестандартных аппаратных средств, имеющих возможность влиять на нормальный ход работы компьютера или ОС.

Если ПЭВМ подключена к общедоступным сетям связи, должны быть предприняты дополнительные меры, исключающие возможность несанкционированного доступа к системным ресурсам используемых операционных систем, к программному обеспечению, в окружении которого функционирует Сервер, и к компонентам ПЭВМ со стороны указанных сетей.

В случае компрометации ключей по факту компрометации должно быть проведено служебное расследование. Скомпрометированные ключи выводятся из действия.

Выведенные из действия, скомпрометированные ключевые носители после проведения служебного расследования уничтожаются, о чем делается запись в «Журнале пользователя сети» (подробнее см. «ЖТЯИ.00096-01 91 01. КриптоПро HSM. Руководство администратора безопасности»).

Не допускается:

Осуществлять несанкционированное копирование ключевых носителей.

Разглашать содержимое носителей ключевой информации или передавать сами носители лицам, к ним не допущенным, выводить ключевую информацию на дисплей и принтер.

Использовать ключевой носитель в режимах, не предусмотренных штатным режимом использования ключевого носителя.

Записывать на ключевой носитель постороннюю информацию.

Оставлять без контроля уполномоченных лиц вычислительные средства, входящие в состав ПАКМ/Серверов/рабочих станций, при включенном питании и загруженном программном обеспечении. При кратковременном перерыве в работе рекомендуется производить гашение экрана с возобновлением активности экрана по паролю доступа.

Подключать к Серверу и ПАКМ дополнительные устройства и соединители, не предусмотренные штатной комплектацией.

Эксплуатировать ПЭВМ и ПАКМ, если во время его начальной загрузки не проходит встроенный тест ОЗУ.

Вносить какие-либо изменения в программное обеспечение ПЭВМ и ПАКМ.

Изменять настройки, установленные программами установки ПАКМ «КриптоПро HSM» или администратором.

Использовать синхропосылки, вырабатываемые не средствами ПАКМ.

Обрабатывать на ПЭВМ информацию, содержащую государственную тайну.

Использовать бывшие в работе ключевые носители для записи новой информации без предварительного уничтожения на них ключевой информации средствами ПАКМ.

Осуществлять несанкционированное вскрытие системных блоков ПЭВМ и ПАКМ.

Запускать на ПЭВМ сервисы для удаленного входа пользователей из глобальной сети.

Устанавливать средства разработки и отладки ПО на ПЭВМ.

Приносить и использовать в помещении, где размещены ПЭВМ, радиотелефоны и другую радиопередающую аппаратуру.

8.5. Организационно-технические меры защиты

На ПЭВМ должны быть установлены последние обновления программных продуктов, касающиеся безопасности.

Должен быть реализован следующий комплекс организационно-технических мер защиты от НСД.

В системе регистрируются пользователи, обладающие правами администратора, на которых возлагается обязанность конфигурировать операционную систему ПЭВМ, настраивать безопасность ОС, а также конфигурировать оборудование ПЭВМ.

Для администратора выбирается надежный пароль входа в систему, удовлетворяющий следующим требованиям: длина пароля не менее 6 символов, среди символов пароля встречаются заглавные символы, прописные символы, цифры и специальные символы, срок смены пароля не реже одного раза в месяц, доступ к паролю должен быть обеспечен только администратору.

Контроль срока использования ключа ЭП, реализуемый в виде блокировки работы ПАКМ в случае попытки использования ключа дольше заданного срока, обеспечивается организационно-техническими мерами.

Всем пользователям, зарегистрированным в ОС, администратор в соответствии с политикой безопасности, принятой в организации, дает минимально возможные для нормальной работы права. Каждый пользователь ОС, не являющийся администратором, может просматривать и редактировать только свои установки в рамках прав доступа, назначенных ему администратором.

На ПЭВМ устанавливается только одна ОС. Не должны использоваться нестандартные, измененные или отладочные версии операционных систем.

Права доступа к каталогам ПЭВМ должны быть установлены в соответствии с политикой безопасности, принятой в организации.

Должна быть проведена установка атрибутов безопасности процессов и потоков в соответствии с требованиями безопасности всей системы в целом.

В случае подключения ПЭВМ с установленным ПАКМ к общедоступным сетям передачи данных должно быть исключено использование JavaScript, VBScript, ActiveX и других программных объектов, загружаемых из сети.

Должна быть отключена возможность удаленного администрирования ПЭВМ с установленным ПАКМ, а также Серверов и рабочих станций для всех пользователей.

Должен быть закрыт доступ ко всем не используемым портам.

При использовании удаленного рабочего места администрирования ПАКМ (раздел 12 «ЖТЯИ.00096-01 90 01. КриптоПро HSM. Инструкция по использованию») рабочая станция администратора должна быть размещена в контролируемой зоне. Сетевое подключение данной рабочей станции к ПАКМ должно осуществляться напрямую (без промежуточного сетевого оборудования) в свободный Ethernet порт ПАКМ. Если удаленное рабочее место администрирования ПАКМ не используется, необходимо программно отключать соответствующий порт при помощи штатных средств настройки ПАКМ.

Должны быть исключены исполнение и открытие файлов, полученных из общедоступных сетей передачи данных, без проведения соответствующих проверок на предмет содержания в них программных закладок и вирусов.

Должны быть удалены все общие ресурсы на ПЭВМ, которые не используются. Права доступа к используемым общим ресурсам должны быть заданы в соответствии с политикой безопасности, принятой в организации.

Должна быть разработана система назначения и смены паролей.

Должна использоваться система аудита в соответствии с политикой безопасности, принятой в организации, и организован регулярный анализ результатов аудита.

Должен проводиться регулярный просмотр сообщений в журналах событий ОС, ППО ПЭВМ и ПАКМ «КриптоПро HSM».

Должна быть исключена возможность создания аварийного дампа оперативной памяти ПЭВМ.

При загрузке ОС на ПЭВМ должен быть реализован контроль целостности программного обеспечения ПЭВМ.

Средствами BIOS должна быть исключена возможность отключения пользователями ISA и PCI устройств при использовании программно-аппаратных средств защиты от НСД, устанавливаемых в ISA и PCI разъем. В BIOS ПЭВМ определяются установки, исключающие возможность загрузки операционной системы, отличной от установленной на жестком диске: отключается возможность загрузки с гибкого диска, привода CD-ROM и прочие нестандартные виды загрузки ОС, включая сетевую загрузку. Не применяются ПЭВМ с BIOS, исключающим возможность отключения сетевой загрузки ОС для «КриптоПро HSM Client».

Вход в BIOS ПЭВМ должен быть защищен паролем, к которому предъявляются те же требования, что и к паролю администратора (длина пароля не менее 6 символов, среди символов пароля встречаются заглавные символы, прописные символы, цифры и специальные символы, срок смены пароля не реже одного раза в месяц). Пароль для входа в BIOS должен быть известен только администратору и быть отличным от пароля администратора для входа в ОС.

Средствами BIOS должна быть исключена возможность работы на ПЭВМ, если во время его начальной загрузки не проходят встроенные тесты.

Должно быть реализовано физическое затирание содержимого удаляемых файлов.

Должна быть исключена возможность использования в составе ПЭВМ аппаратных средств поддержки удаленного администрирования (Remote Insight Board/PCI всех модификаций).

Должна быть исключена возможность использования на ПЭВМ программного обеспечения, поддерживающего технологию удаленного управления (Compaq Insight Manager, Remote ROM Flash Setup Utility, COMPAQ MultiNIC Boot Utility и т.п.).

8.6. Ежесуточное нагрузочное тестирование ПАКМ

Для уменьшения вероятности криптографически опасных последствий Администратор безопасности должен периодически проводить контроль исправности аппаратного модуля ПАКМ путем его перезагрузки и осмотра.

Необходимо выполнять ежесуточное нагрузочное тестирование, обеспечивающее контроль работоспособности ПАКМ, корректного выполнения криптографических функций, контроль аппаратной части всей подсистемы криптографической защиты информации. Тест сводится к кратковременной максимальной загрузке подсистемы набором криптографических операций и

последующим её анализом администратором. Тест максимально загружает канал К, процессор и оперативную память ПАКМ.

Для этого на сервере или специально выделенной рабочей станции, подключенными к ПАКМ, запускается специальное задание, включающее:

- создание тестовых ключей в ПАКМ (выполняется один раз);
- выполнение процедуры нагрузочного тестирования (ежедневно).

Создание тестовых ключевых контейнеров в ПАКМ выполняется с использованием следующих команд:

```
csptest -keyset -newkeys -provtype 75 -cont "\\.\HSM\A"
```

```
csptest -keyset -newkeys -provtype 75 -cont "\\.\HSM\B"
```

При создании каждого ключа будет выдан запрос на значение pin-код для создаваемого ключа на LCD панель ПАКМ или на экран рабочей станции (в зависимости от используемого сертификата ключа доступа к ПАКМ на смарткарте). Задайте для этих значений, например, код 11111111.

Процедура нагрузочного тестирования запускается следующей командой:

```
csptest -perf -sign -hash -encrypt -data 512 -cycles
100,100,100,100,100,100,100,100,100,100,100,100,100,100 -sender "\\.\HSM\A" -recipient
"\\.\HSM\B" -thread 20 -repeat 20 -pinsend 11111111 -pinrecip 11111111
```

Исполняемый модуль csptest.exe находится в каталоге ...Program files\Crypto Pro\HSM

Результат теста выводится на стандартный вывод – консоль, который можно перенаправить в файл для последующего анализа. Положительный результат теста – отсутствие сообщений об ошибках как в выводе в окно теста на сервере, так и на LCD панели ПАКМ.

Пример успешного завершения теста можно посмотреть на следующем рисунке:

```

Командная строка
Performance statistic results:
Parameter          OperUsed <Oper/s>  d Oper/s  %  BandUsed <Kb/s>  d Kb/s
-----
ExportPublicKey    20      1707.9    216.1    13  none      ---      ---
ImportPublicKey    20      449.6     33.2     7   none      ---      ---
GenSessionKey      20      900.3     59.6     7   none      ---      ---
ExportSessionKey   20     1578.4    68.9     4   none      ---      ---
ImportSessionKey   20      915.3     3.6      0   none      ---      ---
Encrypt             20       27.1     0.7      2   20       277.7    6.9
Decrypt             20      424.0     8.0      2   20       217.1    4.1
SignHash            20      915.8     3.3      0   none      ---      ---
VerifySignature    20      696.4     28.3     4   none      ---      ---
HashData            20     1601.4    12.7     1   20       819.9    6.5
EncryptMessage      none      ---      ---      -   none      ---      ---
DecryptMessage      none      ---      ---      -   none      ---      ---
Total: SYS: 0.094 sec USR: 0.391 sec UTC: 118.406 sec
[ErrorCode: 0x00000000]
C:\Program Files\Crypto Pro\CSP>

```


8.7. Контроль целостности

ПАКМ «КриптоПро HSM» имеет встроенные функции контроля целостности ПО, которые выполняются периодически (один раз в сутки) и с каждым запуском ПЭВМ. В качестве средств контроля целостности используются:

- средства электронного замка «Соболь»;
- штатные средства программного обеспечения ПАКМ.

Контроль целостности ПО является 2-х этапным.

На первом этапе производится проверка всех файлов на загружаемом разделе /boot ПАКМ «КриптоПро HSM» (в данном разделе расположены загружаемое ядро ОС и конфигурационные файлы ОС, а также файлы с контрольными суммами всех файлов неизменяемого и монтируемого только на чтение раздела «/») - выполняется электронным замком «Соболь» до загрузки ОС. Проверка выполняется путем вычисления контрольной суммы и сравнением ее с предвычисленным значением.

На втором этапе (загрузка ОС) средствами ПО ПАКМ «КриптоПро HSM» вычисляются контрольные суммы всех файлов корневого раздела и сравниваются с вычисленными в момент изготовления ПАКМ, хранящимися в разделе /boot, контролируемым электронным замком.

Контроль целостности в ПАКМ «КриптоПро HSM» охватывает:

- модули криптопровайдера;
- ядро, модули ядра ОС, их конфигурационные файлы;
- модули канала «К», «К2»;
- драйверы устройств и портов ввода ГМД, смарт-карт, com-порта, сети Ethernet;
- драйвер электронного замка «Соболь».

Кроме этого проверка контрольных сумм файлов осуществляется по расписанию (один раз в сутки).

В случае, если проверка дала отрицательный результат, ПАКМ останавливается (выполняется процедура halt).

Сервера и рабочие станции также имеют средства контроля целостности. В качестве средств контроля целостности на ПЭВМ используются:

- средства электронного замка;
- средства ПО ПАКМ, обеспечивающего интерфейс к функциям ПАКМ.

Контроль целостности ПО является 2-х этапным.

На первом этапе производится проверка файлов средствами электронного замка до загрузки ОС (для исполнений с уровнем защиты КС2 и выше). Проверка выполняется путем вычисления контрольной суммы и сравнением ее с предвычисленным значением.

На втором этапе средствами ПО ПАКМ проверяются контрольные суммы файлов, входящих в комплект Сервера/рабочей станции. Контрольные суммы вычисляются при изготовлении дистрибутива для каждого файла отдельно и записываются в его заголовок по определенному смещению. Он охватывает:

- драйверы устройств и портов ввода смарт-карт, com-порта, сети Ethernet;
- программы отображения log-файлов;
- драйвер электронного замка.

Если в результате периодического контроля целостности или при загрузке операционной системы появляются сообщения о нарушении целостности контролируемого файла, пользователь обязан прекратить работу и обратиться к администратору безопасности.

Администратор безопасности, проанализировав причину, приведшую к нарушению целостности, должен переустановить ПО Сервера/рабочей станции с дистрибутива.

8.8. Электронный замок

Система Электронного замка в ПАКМ предназначена для организации защиты компьютера от входа посторонних пользователей (защита от НСД). Под посторонними пользователями понимаются все лица, не зарегистрированные в системе электронного замка как пользователи данного компьютера.

Система электронного замка обеспечивает:

- регистрацию пользователей компьютера и назначение им персональных идентификаторов и паролей на вход в систему;
- запрос персонального идентификатора и пароля пользователя при загрузке компьютера;
- возможность блокирования входа в систему зарегистрированного пользователя;
- ведение системного журнала, в котором производится регистрация событий, имеющих отношение к безопасности системы;
- контроль целостности файлов на жестком диске;
- контроль целостности физических секторов жесткого диска;
- аппаратную защиту от несанкционированной загрузки операционной системы с гибкого диска и CD-ROM диска.

Так же система электронного замка включает в себя физический датчик случайных чисел, используемый криптографическими функциями ПАКМ «КриптоПро HSM».

Установка и настройка электронного замка ПАКМ производится предприятием-изготовителем. Настройка исключает возможность вмешательства пользователя в процессы загрузки операционной системы и прикладного ПО и проверки целостности программной среды.

Ключи электронного замка, без которых невозможна загрузка ПАКМ, должны находиться в ведении администратора безопасности.

Ключи электронного замка разделяются на 2 вида:

- ключи администратора;
- ключи пользователя (в заказанном количестве, не менее 2 шт.).

Ключи администратора используются только сотрудниками предприятия-изготовителя для осуществления изготовления (настройки) ПАКМ и уничтожаются после завершения изготовления изделия.

Ключи пользователя администратор безопасности должен выдавать лицам, выполняющим работу с использованием ПАКМ, осуществляющим включение и загрузку ПАКМ.

Ключи электронного замка относятся к категории ключевых носителей, подлежат соответствующему учету и хранению.

8.9. Защита от вскрытия корпуса ПАКМ

Все ключи в ПАКМ «КриптоПро HSM» хранятся на диске в зашифрованном виде. Ключи шифрования, в свою очередь, зашифрованы на ключе активации ПАКМ, а ключ активации, разделенный по схеме 3 из 5-ти собирается в момент ввода его частей с внешних носителей – смарт карт. Таким образом, проникновение в ПАКМ, находящийся в выключенном/неактивированном состоянии, с целью компрометации ключей пользователей не представляет особой угрозы.

При активации ПАКМ, ключи пользователей по мере обращения к ним считываются в оперативную память и там расшифровываются. В таком состоянии существует угроза вскрытия ПАКМ, быстрого изъятия и заморозки модулей памяти с последующим исследованием банков памяти на предмет поиска последовательностей, содержащих образы закрытых ключей.

Корпус ПАКМ «КриптоПро HSM» оборудован датчиком вскрытия корпуса. В случае его срабатывания происходит моментальная очистка памяти, используемая для открытых контекстов ключей, в журнал заносится информация о данном событии, и ПАКМ выключается.

Дальнейшая загрузка ПАКМ становится невозможной.

В случае возникновения подобной ситуации необходимо обратиться в службу поддержки компании изготовителя.

9. Резервирование и восстановление ПАКМ

Для обеспечения надежности ПАКМ «КриптоПро HSM» имеет возможность резервирования ключевой информации и, при необходимости, её восстановления.

9.1. Холодное резервирование ПАКМ

Данный режим применяется, в основном, для сохранения важной, редко изменяющейся ключевой информации, например, ключа уполномоченного лица удостоверяющего центра, предназначенного для подписи сертификатов открытых ключей пользователей.

Основной режим холодного резервирования – внутренний. Он подразумевает создание резервной копии состояния ПАКМ, включая настройки ПАКМ и текущее состояние всей ключевой системы, и сохранение её внутри ПАКМ на отдельном, специально предназначенном для хранения резервных копий, флэш-диске.

В случае порчи ключевой информации в основном разделе файловой системы ПАКМ, её можно попытаться восстановить из ранее созданной резервной копии. В случае порчи флэш-диска с основным разделом ПАКМ такой, что восстановление на этом диске невозможно, только этот флэш-диск можно заменить, после чего заново развернуть ПАКМ и восстановиться с резервной копии.

Если требуется ещё более надежное хранение ключа уполномоченного лица, например, в другом помещении или даже районе, имеется возможность выгрузки резервной копии из ПАКМ и транспортировка её в нужное место для хранения. Обратную загрузку в ПАКМ такого файла резервной копии может выполнить привилегированный пользователь, которому назначена роль Администратора резервного копирования, через интерфейс удаленного рабочего места Администратора ПАКМ. При этом Администратором ПАКМ должен быть выставлен специальный параметр ПАКМ «Enable Upload» в значение, разрешающее производить такую загрузку файлов. В целях безопасности в обычном режиме функционирования ПАКМ данный параметр должен быть отключен. Таким образом, в системах, требующих очень высокий уровень доступности, можно изготовить резервный ПАКМ и в случае выхода из строя основного, быстро переключиться на резервный.

Загрузка файла резервной копии не означает восстановление. Данная операция только сохраняет файл с резервной копией в соответствующем разделе. Для восстановления ПАКМ из данной резервной копии используйте соответствующий режим LCD меню ПАКМ.

Резервную копию следует делать каждый раз при смене ключевой информации ПАКМ.

Для создания резервной копии предусмотрена роль - «Администратор резервного копирования». Чтобы получить доступ к его функциям, в сертификате ключа доступа к ПАКМ привилегированного пользователя должно присутствовать специальное расширение (см. п. «Ролевая модель доступа к функциям ПАКМ» документа «ЖТЯИ.00096-01 90 01. КриптоПро HSM. Инструкция по использованию»).

Кроме того, что вся ключевая информация, хранящаяся в ПАКМ, зашифрована в конечном итоге на разделенном на 5 частей ключе активации ПАКМ, и именно в таком виде она попадает в файл с резервной копией, получаемый файл резервной копии подписывается для контроля целостности и после этого ещё раз шифруется на уникальном для данной копии ключе, формируемом на смарт-карте. На открытый ключ формируется самоподписанный сертификат ключа, изданный ПАКМ, и тоже помещается

на карту. Данная смарт-карта формируется в момент создания резервной копии Администратором резервного копирования и должна храниться только у него.

Выгрузить файл резервной копии и/или запустить процедуру восстановления ПАКМ из резервной копии может только Администратор ПАКМ, но без ключа шифрования и подписи резервной копии (смарт-карты), хранящейся у Администратора резервного копирования, процедура восстановления невозможна. Таким образом, чтобы провести атаку с использованием резервной копии, необходим сговор, как минимум двух привилегированных пользователей категории II (имеющих доступ в контролируемую зону). Кроме того, все подобные операции отражаются в журнале аудита ПАКМ, доступном для очистки только Суперпользователю ПАКМ, либо Аудитору совместно с Администратором ПАКМ.

Необходимо помнить, что восстановление с резервной копии не имеет смысла, если утеряны или неработоспособны, как минимум, 3 из 5 карт с защитными ключами ключа активации ПАКМ, действовавшего на момент создания резервной копии. Таким образом, после каждой успешной смены ключа активации ПАКМ, необходимо пересоздать резервную копию.

Если в резервировании информации нет необходимости, то в регламенте не должно быть предусмотрено роли «Администратора резервного копирования». Пользователь с такими привилегиями просто не должен создаваться.

9.2. Горячее резервирование ПАКМ

Данный режим резервирования может применяться в системах высокой степени доступности с часто меняющейся ключевой информацией пользователей ПАКМ.

Он требует наличия, как минимум, одного дополнительного ПАКМ «КриптоПро HSM», включенного в кластер с основным ПАКМ.

Горячее резервирование ПАКМ «КриптоПро HSM» подразумевает перенос новых и изменённых в основном ПАКМ данных в резервный ПАКМ в онлайн режиме. Фактически, изменения отображаются в резервном ПАКМ сразу после обновления данных в основном ПАКМ.

Это относится ко всем данным ПАКМ (информация о пользователях ПАКМ, ключи пользователей, служебные ключи ПАКМ, настройки ПАКМ), за исключением настроек сетевых интерфейсов, настроек горячего резервирования, ключа активации и ключей шифрования ПАКМ, хранилищ сертификатов служебных ключей ПАКМ.

Таким образом, можно говорить, что в момент отказа в работе основного ПАКМ, резервные ПАКМ будут содержать информацию максимально близкую к информации основного ПАКМ на момент отказа. Для восстановления работы системы необходимо лишь переключить клиента на резервный ПАКМ.

Механизм горячего резервирования основывается на механизмах репликации баз данных. При этом основной ПАКМ «КриптоПро HSM» называется MASTER сервером репликации, а ПАКМ-ы горячего резерва называются SLAVE серверами.

SLAVE сервера отслеживают события изменения данных на MASTER сервере и при их обнаружении «забирают» и отображают все изменения у себя.

Между SLAVE и MASTER сервером создается зашифрованный канал, аналогичный каналу K2, т.е. все данные передаются при помощи TLS (ГОСТ Р 34.10-2012) протокола с взаимной аутентификацией сторон при помощи сертификатов открытых ключей обмена.

Для организации этого канала используются закрытые ключи и сертификаты открытых ключей обмена TLS серверов, т.е. те же самые, которые используются для организации канала K2 при взаимодействии с клиентами ПАКМ.

Т.к. эти ключи зашифрованы на ключах шифрования ПАКМ, которые в свою очередь зашифрованы на ключе активации ПАКМ, то механизм репликации может работать только после активации ПАКМ (в режимах ACTIVE или ADMIN_ONLY). При правильно настроенной репликации она автоматически запускается при входе в один из этих режимов и останавливается при переводе ПАКМ в неактивное (INACTIVE) состояние.

Изначально между двумя разными ПАКМ нет никакого доверия, сертификаты ключей TLS серверов неизвестны друг, т.к. изданы различными (каждый своим) ПАКМ-ами, что не позволяет установить доверенный канал между ними, даже для синхронизации данных. Поэтому для того, чтобы развернуть сервер горячего резервирования его необходимо клонировать с MASTER сервера ПАКМ, используя механизм холодного резервирования ПАКМ, описанный в предыдущей главе.

Сразу после создания резервной копии MASTER сервера, следует объявить его (настроить) MASTER сервером, создать правила фаервола, разрешающие подключение к нему SLAVE серверов через порт репликации (1504) с заданных IP адресов и, перестартовав сетевые сервисы, включить в работу (перевести в состояние ACTIVE). Удаленные пользователи уже могут начинать штатную работу.

На SLAVE серверах необходимо произвести инициализацию обычным способом. Временно настроить их таким образом, чтобы была возможность подключиться к ним по Web интерфейсу, дополнительно разрешив загрузку файлов резервных копий в них. Далее необходимо выгрузить файл резервной копии из MASTER сервера и загрузить его в SLAVE сервера, восстановить SLAVE сервера из данной резервной копии. В результате получим клоны MASTER сервера на момент создания его резервной копии, включая разделенный ключ активации MASTER сервера, ключевую систему привилегированных и обычных пользователей, сетевые настройки и настройки брэндмауэра. Данные настройки на SLAVE серверах необходимо изменить (как минимум IP адреса сетевых интерфейсов), после чего выполнить процедуру регенерации ключа и сертификата сервера (TLS). В настройках репликации необходимо указать тип сервера (SLAVE), уникальный номер SLAVE сервера (от 1 до 9), IP адрес MASTER сервера.

Все настройки выполняются в соответствии с инструкциями, описанными в документе «ЖТЯИ.00096-01 90 01 КриптоПро HSM. Инструкция по использованию».

Если всё сделано правильно, то после активации SLAVE серверов все изменения данных, произошедшие на MASTER сервере с момента создания на нём резервной копии, использовавшейся для клонирования, отобразятся на SLAVE серверах, а информация о состоянии репликации, отображаемая в Web интерфейсе, не будет содержать данных о возникшей ошибке.

Не следует без надобности использовать SLAVE сервера HSM для штатной работы с ними клиентов HSM. Изменения реплицируются только в одну сторону - с MASTER сервера на SLAVE сервера. Частое переключение клиентов с MASTER на SLAVE и обратно в конечном итоге может привести к потере данных. SLAVE сервера должны использоваться только в качестве горячего резерва.

В случае необходимости SLAVE сервера можно останавливать, деактивировать, перегружать. Их можно использовать для периодического создания образов HSM в виде обычных резервных копий. Только необходимо помнить, что данные резервные копии будут содержать информацию и о типе сервера (SLAVE) и соответствующих настройках,

и после восстановления ПАКМ из такой копии, возможно придется данные настройки сразу изменить.

После активации SLAVE серверов после простоя все изменения, произошедшие на MASTER сервере с момента остановки SLAVE сервера, автоматически будут реплицированы.

Объем накапливаемых изменений на MASTER сервере ограничен, поэтому не следует останавливать надолго SLAVE сервера. Чтобы не забивать дисковую память, файлы с накопленными изменениями на Master сервере автоматически будут удаляться через 3 суток после их ротации (смены на новый файл изменений).

Изменения, связанные со сменой ключа активации, ключа подписи ПАКМ, не реплицируются. Поэтому после проведения данных процедур необходимо заново пересобрать репликацию.

Количество используемых SLAVE серверов диктуется требованиями к надежности системы и бюджетом эксплуатирующей организации. В общем случае достаточно одного сервера горячего резерва. Необходимо иметь в виду, что отказ оборудования может наступить и на SLAVE сервере. И если используется только один SLAVE сервер, для ввода нового сервера горячего резерва придется заново пересобрать репликацию (если прошло более 3 суток с момента создания резервной копии на MASTER сервере).

В случае сбоя на MASTER сервере (например, отключения питания), его база данных может оказаться в разрушенном состоянии. Программное обеспечение ПАКМ будет пытаться осуществить максимально возможное восстановление данных. Но может оказаться так, что часть данных будет потеряна. В таком случае это может стать поводом для переключения работы клиентов HSM на один из резервных (SLAVE) серверов, равно как и отказ оборудования MASTER сервера.

Сразу после такого переключения следует остановить репликацию на SLAVE сервере и изменить его тип, сделав MASTER сервером. Быстрое переключение на резервный сервер дает возможность продолжить штатную работу клиентам ПАКМ. Если использовалось более одного SLAVE сервера, то оставшиеся сервера можно остановить. Они становятся мало функциональными. Необходимо запланировать работы по настройке новой системы горячего резервирования. Для этого потребуются кратковременная остановка работы клиентов ПАКМ для создания на новом MASTER сервере резервной копии, которая будет использоваться для клонирования новых SLAVE серверов.

В настоящий момент ПАКМ не поддерживает автоматическое переключение с основного сервера на резервный. С ПАКМ «КриптоПро HSM» может работать несколько устройств доступа (серверов, рабочих станций пользователей), которые ничего не знают друг о друге. Отказ какому-либо устройству в доступе к ПАКМ, не означает, что ПАКМ нефункционален и данному устройству следует переключаться на резервный ПАКМ. ПАКМ может быть просто перегружен в данный момент (превышение количества соединений и т.п.) и абсолютное большинство пользователей успешно работают с ним в данный момент.

Чтобы избежать потери данных, переключение на резервный ПАКМ «КриптоПро HSM» должно осуществляться синхронно для всех клиентов (устройств доступа) ПАКМ.

Работоспособность ПАКМ должна отслеживаться Администратором ПАКМ.

Для сетевого соединения между собой MASTER и SLAVE серверов в целях репликации следует использовать выделенные сетевые интерфейсы. При наличии только одного SLAVE сервера рекомендуется подключать сервера напрямую. При

наличии нескольких SLAVE серверов следует использовать выделенный оптический коммутатор, используя при этом выделенный сегмент сети.

Настройки репликации (IP адрес MASTER сервера, типы и номера серверов репликации, а также настройки брандмауэра, связанные с репликацией) выполняются привилегированным пользователем с правами Администратора ПАКМ. Репликацию невозможно создать без предварительной процедуры клонирования ПАКМ, а соответственно без участия привилегированного пользователя с правами «Администратора резервного копирования ПАКМ».

10. Защита от вскрытия корпуса ПАКМ

Все ключи в ПАКМ «КриптоПро HSM» хранятся на диске в зашифрованном виде. Ключи шифрования, в свою очередь, зашифрованы на ключе активации ПАКМ, а ключ активации, разделенный по схеме 3 из 5-ти, собирается в момент ввода его частей с внешних носителей – смарт карт. Таким образом, проникновение в ПАКМ, находящийся в выключенном/неактивированном состоянии, с целью компрометации ключей пользователей не представляет особой угрозы.

При активации ПАКМ, ключи пользователей по мере обращения к ним считываются в оперативную память и там расшифровываются. В таком состоянии существует угроза вскрытия ПАКМ, быстрого изъятия и заморозки модулей памяти с последующим исследованием банков памяти на предмет поиска последовательностей, содержащих образы закрытых ключей.

Корпус ПАКМ «КриптоПро HSM» оборудован датчиком вскрытия корпуса. В случае его срабатывания происходит моментальная очистка памяти, используемая для открытых контекстов ключей, в журнал заносится информация о данном событии, и ПАКМ выключается.

Дальнейшая загрузка ПАКМ становится невозможной.

В случае возникновения подобной ситуации необходимо обратиться в службу поддержки компании изготовителя.

11. Эксплуатация ПАКМ

Эксплуатация ПАКМ должна производиться в соответствии с инструкцией, содержащейся в документе «ЖТЯИ.00096-01 90 01. КриптоПро HSM. Инструкция по использованию».

Администратор безопасности Серверов и ПАКМ «КриптоПро HSM» должен вести следующие журналы:

- «Журнал регистрации администраторов безопасности и пользователей»;
- «Журнал пользователя сети».

Описание журналов представлено в документе «ЖТЯИ.00096-01 90 01. КриптоПро HSM. Инструкция по использованию».

Установка ПАКМ и его включение описываются в документе «ЖТЯИ.00096-01 90 01. КриптоПро HSM. Инструкция по использованию».

Администрирование ПАКМ и удаленное администрирование ПАКМ описываются в документе «ЖТЯИ.00096-01 90 01. КриптоПро HSM. Инструкция по использованию».

Порядок выполнения технического обслуживания и ремонта ПАКМ описываются в документе «ЖТЯИ.00096-01 90 01. КриптоПро HSM. Инструкция по использованию».

Необходимо использовать усиленный контроль использования ключей (см. п. 4.2 «ЖТЯИ.00096-01 90 01. КриптоПро HSM. Инструкция по использованию»).

12. Рекомендации по использованию ПАКМ

ПАКМ «КриптоПро HSM» предназначен для предоставления пользователю доступа к криптографическим функциям. Правильность построения функций управления возлагается на специалиста, осуществляющего встраивание ПАКМ «КриптоПро HSM» в конкретную защищаемую систему связи.

При каждом использовании ключа проверки подписи (сертификата ключа проверки подписи) или сертификата открытого ключа обмена должна проводиться проверка подлинности ключа (сертификата) и поиск ссылки на данный ключ (сертификат) в «списке отозванных (аннулированных) сертификатов».

Пользователь несет персональную ответственность за хранение ключевых документов.

Режим простой замены необходимо использовать только для зашифрования/расшифрования ключевой информации.

Алгоритм открытого распределения ключей обеспечивает распределение сеансовых ключей между участниками информационного обмена, не обеспечивая при этом аутентификацию участников. В связи с этим данный алгоритм должен использоваться совместно с протоколами аутентификации.

Для всех режимов алгоритма шифрования следует обеспечить отсутствие перекрытий ключей.

Кроме того, рекомендуется:

- на открытые данные вычислять имитовставку в соответствии с ГОСТ 28147-89;
- формирование синхропосылки следует производить с использованием встроенного в ПАКМ датчика случайных чисел;
- предусмотреть в формате данных, подлежащих шифрованию, информацию, защищающую от повторов ранее переданных сообщений.

13. Встраивание ПАКМ

При встраивании ПАКМ ЖТЯИ.00096-01 (собственно ПАКМ, клиентская компонента ПАКМ «КриптоПро HSM Client») в прикладные системы необходимо по Техническому заданию, согласованному с 8 центром ФСБ России, проводить оценку влияния аппаратных, программно-аппаратных и программных средств сети (системы) конфиденциальной связи, совместно с которыми предполагается штатное функционирование ПАКМ, на выполнение предъявленных к ПАКМ требований:

- если информация конфиденциального характера подлежит защите в соответствии с законодательством Российской Федерации;

- при организации криптографической защиты информации конфиденциального характера в федеральных органах исполнительной власти, органах исполнительной власти субъектов Российской Федерации (далее – государственные органы);

- при организации криптографической защиты информации конфиденциального характера в организациях независимо от их организационно-правовой формы и формы собственности при выполнении ими заказов на поставку товаров, выполнение работ или оказание услуг для государственных нужд (далее – организации, выполняющие государственные заказы);

- если обязательность защиты информации конфиденциального характера возлагается законодательством Российской Федерации на лиц, имеющих доступ к этой информации или наделенных полномочиями по распоряжению сведениями, содержащимися в данной информации;

- при обработке информации конфиденциального характера, обладателем которой являются государственные органы или организации, выполняющие государственные заказы, в случае принятия ими мер по охране ее конфиденциальности путем использования средств криптографической защиты;

- при обработке информации конфиденциального характера в государственных органах и в организациях, выполняющих государственные заказы, обладатель которой принимает меры к охране ее конфиденциальности путем установления необходимости криптографической защиты данной информации.

В указанных выше случаях, если встраивание СКЗИ производится в прикладные системы, в которых функции создания и/или проверки электронной подписи не являются автоматическими, в том числе необходимо проводить оценку соответствия прикладной системы п.п. 8 и/или 9 Приложения 1 к Приказу ФСБ РФ от 27 декабря 2011 г. № 796 «Об утверждении Требований к средствам электронной подписи и Требований к средствам удостоверяющего центра».

В остальных случаях рекомендуется проводить установленным порядком оценку влияния среды функционирования на СКЗИ с целью оценки обоснованности и достаточности мер, принятых для защиты информации, обрабатываемой СКЗИ.

Разработка программного обеспечения на основе «КриптоПро HSM Client» с учетом п. 1.5 Формуляра ЖТЯИ.00096-01 30 01 может производиться без создания новых СКЗИ в случае использования вызовов из приведенного в Приложении 1 перечня в соответствии с документацией.

В случае использования вызовов, не входящих в перечень Приложения 1 документа «ЖТЯИ.00096-01 95 01. КриптоПро HSM. Правила пользования», необходимо производить разработку отдельного СКЗИ на базе ПАКМ «КриптоПро HSM» версия 2.0

(с проведением соответствующих тематических исследований) в соответствии с действующей нормативной базой (в частности, с Постановлением Правительства Российской Федерации от 16 апреля 2012 г. №313 и Положением о разработке, производстве, реализации и эксплуатации шифровальных (криптографических) средств защиты информации (Положение ПКЗ-2005)).

Следующие приложения из состава ПАКМ «КриптоПро HSM» версия 2.0:

- приложение командной строки для подписи и шифрования файлов cryptsp;
- приложение командной строки для работы с сертификатами certmgr;
- приложение для создания TLS-туннеля stunnel.

можно использовать без проведения оценки влияния указанных приложений на программные интерфейсы ПАКМ, но требуется проведение оценки влияния компонентов среды функционирования (прикладных систем, программных продуктов и т.п.), вызывающих указанные приложения, на их штатное функционирование.

Проведение тематических исследований приложений, указанных выше, не требуется.

Приложение 1. Перечень вызовов, использование которых при разработке систем на основе ПАКМ «КриптоПро HSM» всех комплектаций с учетом п.1.5 Формуляра ЖТЯИ.00096-01 30 01 возможно без дополнительных тематических исследований

Таблица 1 — Перечень функций интерфейса CryptoAPI

Функция	Описание	Ограничения на использование функции
Функции инициализации и настройки провайдера		
CryptAcquireContext	Функция CryptAcquireContext используется для создания дескриптора криптопровайдера с именем ключевого контейнера, определённым параметром pszContainer	Полученный дескриптор криптопровайдера должен быть в обязательном порядке освобождён с помощью вызова функции CryptReleaseContext (за исключением вызовов с флагом CRYPT_DELETEKEYSET).
CryptReleaseContext	Функция CryptReleaseContext используется для удаления дескриптора 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	Управляет счетчиком дескрипторов созданного CryptAcquireContext.	
CryptEnumProviders	Перечисление установленных криптопровайдеров	
CryptEnumProviderTypes	Перечисление установленных типов криптопровайдеров	
CryptGetDefaultProvider	Получение контекста провайдера, установленного в	

	системе по умолчанию	
CryptGetProvParam	Функция CryptGetProvParam получает параметры криптопровайдера.	
CryptSetProvParam	Функция CryptSetProvParam устанавливает параметры криптопровайдера.	
FreeCryptProvFromCertEx	Функция используется для удаления дескриптора криптопровайдера, созданного CryptAcquireContext.	
CryptInstallDefaultContext, CryptSetProvider, CryptSetProviderEx, CryptUninstallDefaultContext	Функции управления контекстом провайдера по умолчанию	
Функции генерации и обмена ключами, создание конфигурирование и удаление ключей		
CryptGenKey	Функция CryptGenKey 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 CryptDestroyKey удаляет ключ, передаваемый через параметр hKey. После удаления ключ (дескриптор ключа) не может использоваться.	
CryptDeriveKey	Функция CryptDeriveKey производит криптографические ключи сессии на основе значения хэш-функции.	Разрешено использование при следующих условиях: 1. передаваемый объект функции хэширования был создан функцией CryptCreateHash с одним из символьных аргументов: CALG_PBKDF2_94_256,

		<p>CALG_PBKDF2_2012_256, CALG_PBKDF2_2012_512;</p> <p>2. работа с передаваемым объектом функции хэширования производилась только с использованием вызовов функций из текущего Перечня.</p>
CryptExportKey	<p>Функция CryptExportKey используется для экспорта криптографических ключей из ключевого контейнера криптопровайдера, сохраняя их в защищённом виде.</p>	<p>Разрешено экспортировать только открытые ключи (PUBLICKEYBLOB).</p>
CryptGenRandom	<p>Функция CryptGenRandom заполняет буфер случайными байтами.</p>	
CryptGetKeyParam	<p>Функция CryptGetKeyParam возвращает параметры ключа.</p>	
CryptGetUserKey	<p>Функция CryptGetUserKey возвращает дескриптор одной из долговременных ключевых пар в ключевом контейнере.</p>	<p>Полученный дескриптор ключа должен в обязательном порядке быть удалён с помощью вызова функции CryptDestroyKey до вызова функции CryptReleaseContext для рабочего дескриптора криптопровайдера.</p>
CryptImportKey	<p>Функция CryptImportKey используется для импорта криптографического ключа из ключевого блока в контейнер криптопровайдера.</p>	<p>Разрешено импортировать только открытые ключи (PUBLICKEYBLOB) при нулевом значении флагов.</p> <p>Полученный дескриптор ключа должен в обязательном порядке быть удалён с помощью вызова функции CryptDestroyKey до вызова функции CryptReleaseContext для</p>

		рабочего дескриптора криптопровайдера.
CryptSetKeyParam	Функция CryptSetKeyParam устанавливает параметры ключа.	Разрешено использование только со следующими символьными аргументами: KP_CERTIFICATE, KP_CIPHEROID, KP_DHOID, KP_HASHOID.
Функции обработки криптографических сообщений		
CryptSignMessage	Функция CryptSignMessage создает хэш определенного содержания, подписывает хэш и затем производит закодирование и текста исходного сообщения, и подписанного хэша	
CryptVerifyMessageSignature	Функция CryptVerifyMessageSignature проверяет электронно-цифровую подпись подписанного сообщения.	
CryptVerifyDetachedMessageSignature	Функция CryptVerifyDetachedMessageSignature проверяет подписанное сообщение, содержащее отсоединенную (detached) подпись или подписи	
CryptDecodeMessage	Функция декодирует, расшифровывает и проверяет сообщение	
CryptDecryptAndVerifyMessageSignature	Функция декодирует и проверяет сообщение	
CryptEncryptMessage	Функция CryptEncryptMessage зашифровывает и производит закодирование сообщения. Аутентичность сообщения не обеспечивается.	
CryptDecryptMessage	Функция CryptDecryptMessage производит раскодирование и расшифрование сообщения. Проверка аутентичности	

	сообщения не производится. Примечание: Не допускается автоматический анализ результата работы функции, направленный на проверку корректности сообщения.	
CryptGetMessageCertificates	Функция возвращает хранилище сертификатов и списки аннулированных сертификатов из сообщения	
CryptGetMessageSignerCount	Функция возвращает количество подписавших сообщение	
CryptHashMessage	Функция создает хэшированное сообщение	
CryptSignAndEncryptMessage	Функция создает подписанное и зашифрованное сообщение	
CryptSignMessageWithKey	Функция создает подписанное сообщение	
CryptVerifyDetachedMessageHash	Функция проверяет открепленный хэш	
CryptVerifyMessageHash	Функция проверяет хэшированное сообщение	
CryptVerifyMessageSignatureWithKey	Функция проверяет подписанное сообщение	
CryptMsgCalculateEncodedLength	Функция CryptMsgCalculateEncodedLength вычисляет максимальное количество байтов, необходимое для закодированного криптографического сообщения, заданного типом сообщения, параметрами кодирования и общей длиной информации, которая должна быть закодирована.	
CryptMsgOpenToEncode	Функция CryptMsgOpenToEncode открывает криптографическое сообщение для кодирования и возвращает дескриптор открытого сообщения.	
CryptMsgOpenTo-Decode	Функция CryptMsgOpenToDecode	

	открывает криптографическое сообщение для раскодирования и возвращает дескриптор открытого сообщения.	
CryptMsgUpdate	Функция CryptMsgUpdate пополняет текст криптографического сообщения.	
CryptMsgGetParam	Функция CryptMsgGetParam получает параметр сообщения после того, как криптографическое сообщение было раскодировано или закодировано.	
CryptMsgControl	Функция CryptMsgControl выполняет контрольное действие.	
CryptMsgClose	Функция CryptMsgClose закрывает дескриптор криптографического сообщения.	
CryptMsgDuplicate	Функция CryptMsgDuplicate дублирует дескриптор криптографического сообщения путем увеличения счетчика ссылок	
Функции работы с алгоритмами хэширования		
CryptCreateHash	Функция CryptCreateHash инициализирует дескриптор нового объекта функции хэширования потока данных.	<p>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.</p> <p>Полученный дескриптор объекта хэширования</p>

		должен в обязательном порядке быть удалён с помощью вызова функции CryptDestroyHash до вызова функции CryptReleaseContext для рабочего дескриптора криптопровайдера.
CryptDestroyHash	Функция CryptDestroyHash удаляет объект функции хэширования.	
CryptDuplicateHash	Функция CryptDuplicateHash 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 CryptGetHashParam возвращает параметры объекта функции хэширования и значение функции хэширования.	Запрещено использование с символьным аргументом HP_OPAQUEBLOB.
CryptHashData	Функция CryptHashData передаёт данные указанному объекту функции хэширования.	
CryptSetHashParam	Функция CryptSetHashParam устанавливает параметры объекта хэширования.	Разрешено использование только с символьными аргументами HP_PBKDF2_SALT, HP_PBKDF2_PASSWORD, HP_PBKDF2_COUNT, HP_HASHSTARTVECT (при условии, что передаваемый объект функции хэширования был создан функцией CryptCreateHash с символьным аргументом

		CALG_GR3411), HP_OID/KP_HASHOID, HP_OPEN.
CryptSignHash	Функция CryptSignHash возвращает значение электронной цифровой подписи от значения функции хэширования.	Разрешено использование только с ключевыми контейнерами, полученными ранее с помощью вызова CryptAcquireCertificatePrivateKey либо с помощью вызова CertGetCertificateContextProperty из сертификата, проверенного с помощью функции CertVerifyCertificateChainPolicy
CryptVerifySignature	Функция CryptVerifySignature осуществляет проверку цифровой подписи.	Разрешено использование только с дескрипторами ключей, полученными ранее с помощью вызова CryptImportPublicKeyInfo (CryptImportPublicKeyInfo Ex) из сертификата, проверенного с помощью функции CertVerifyCertificateChainPolicy
Функции работы с сертификатами, списками аннулированных сертификатов, хранилищем сертификатов		
Списки аннулированных сертификатов		
CertAddCRLContextToStore	Функция CertAddCRLContextToStore добавляет контекст СОС в хранилище сертификатов.	
CertAddCRLLinkToStore	Функция создает ссылку на список аннулированных сертификатов в другом хранилище	
CertAddEncodedCRLToStore	Функция CertAddEncodedCRLToStore создает контекст СОС из	

	закодированного СОС и добавляет его в хранилище сертификатов. Функция создает копию контекста СОС перед добавлением его в хранилище.	
CertEnumCRLsInStore	Функция CertEnumCRLsInStore получает первый или следующий СОС в хранилище. Эта функция используется в цикле для того, чтобы последовательно получить все СОС в хранилище.	
CertFreeCRLContext	Функция CertFreeCRLContext освобождает контекст СОС, уменьшая счетчик ссылок на единицу. Когда счетчик ссылок обнуляется, функция CertFreeCRLContext освобождает память, выделенную под контекст СОС.	
CertCreateCRLContext	Функция CertCreateCRLContext создает контекст СОС из закодированного СОС. Созданный контекст не помещается в хранилище сертификатов. В созданном контексте функция размещает копию закодированного СОС.	
CertDeleteCRLFromStore	Функция удаляет список аннулированных сертификатов из хранилища	
CertDuplicateCRLContext	Функция CertDuplicateCRLContext дублирует контекст СОС, увеличивая счетчик ссылок на СОС на единицу.	
CertFindCRLInStore	Функция CertFindCRLInStore находит первый или следующий контекст СОС 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 СОС в хранилище сертификатов, удовлетворяющие заданному критерию поиска.	
CertDeleteCertificateFromStore	Функция CertDeleteCertificateFromStore удаляет определенный контекст СОС из хранилища сертификатов.	
CertFindCertificateInCRL	Функция осуществляет поиск заданного сертификата в списке аннулированных сертификатов	
CertGetCRLFromStore	Функция CertGetCRLFromStore получает первый или следующий контекст СОС для определенного издателя сертификата из хранилища сертификатов. Эта функция также осуществляет возможную проверку СОС.	
CertSerializeCRLStoreElement	Функция сериализации списка аннулированных сертификатов со своими свойствами	
Расширенные свойства сертификата списка отозванных (СОС) сертификатов и СТЛ		
CertGetCRLContextProperty	Функция CertGetCRLContextProperty получает расширенные свойства определенного контекста СОС.	
CertSetCRLContextProperty	Функция CertSetCRLContextProperty устанавливает расширенные свойства определенного контекста СОС.	
CertGetCertificateContextProperty	Функция CertGetCertificateContextProperty получает информацию, содержащуюся в расширенных свойствах контекста сертификата.	
CertEnumCertificateContextProperties	Функция CertEnumCertificateContextProperties позволяет перечислить информацию, содержащуюся в	

	расширенных свойствах контекста сертификата.	
CertSetCertificateContextProperty	Функция CertSetCertificateContextProperty устанавливает расширенные свойства для определенного контекста сертификата.	
CertEnumCRLContextProperties	Перечисление расширенных свойств списка аннулированных сертификатов	
CertEnumCTLContextProperties	Перечисление расширенных свойств CTL	
CertGetCTLContextProperty	Получение расширенного свойства CTL	
CertSetCTLContextProperty	Установка расширенных свойств CTL	
CertAddCTLContextToStore	Функция CertAddCTLContextToStore добавляет контекст CTL в хранилище сертификатов	
CertCreateCTLContext	Функция CertCreateCTLContext создает контекст закодированного CTL. Созданный контекст не помещается в хранилище сертификатов. Функция делает копию CTL в созданном контексте.	
CertDuplicateCTLContext	Функция CertDuplicateCTLContext дублирует контекст CTL, увеличивая счетчик ссылок на CTL на единицу.	
CertFreeCTLContext	Функция CertFreeCTLContext освобождает контекст CTL, уменьшая счетчик ссылок на единицу. Когда счетчик ссылок обнуляется, функция CertFreeCTLContext освобождает память, выделенную под контекст CTL.	
Функции работы с сертификатами		

CertAddCertificateContextToStore	Функция CertAddCertificateContextToStore добавляет контекст сертификата в хранилище сертификатов.	
CertAddCertificateLinkToStore	Добавляет ссылку на сертификат в другом хранилище	
CertAddEncodedCertificateToStore	Функция CertAddEncodedCertificateToStore создает контекст сертификата из закодированного сертификата и добавляет его в хранилище сертификатов. Созданный контекст не содержит никаких расширенных свойств.	
CertEnumCertificatesInStore	Функция CertEnumCertificatesInStore 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 CertFreeCertificateContext освобождает контекст сертификата, уменьшая счетчик ссылок на единицу.	
CertCreateCertificateContext	Функция CertCreateCertificateContext создает контекст сертификата из закодированного сертификата. Созданный контекст не помещается в хранилище сертификатов. В созданном контексте функция размещает копию закодированного сертификата.	
CertDuplicateCertificateContext	Функция CertDuplicateCertificateContext дублирует контекст сертификата, увеличивая	

	счетчик ссылок на единицу.	
CertFindCertificateInStore	Функция CertFindCertificateInStore находит первый или следующий контекст сертификата в хранилище сертификатов, который соответствует критерию поиска, определяемому параметром dwFindType и связанным с ним pvFindPara.	
CertDeleteCertificateFromStore	Функция CertDeleteCertificateFromStore удаляет определенный контекст сертификата из хранилища сертификатов.	
CertGetSubjectCertificateFromStore	Функция CertGetSubjectCertificateFromStore получает контекст сертификата из хранилища сертификатов, однозначно определяемый его издателем и серийным номером	
CertGetIssuerCertificateFromStore	Поиск сертификатов издателей заданного сертификата	
CertGetSubjectCertificateFromStore	Поиск сертификата по серийному номеру и издателю	
CertGetValidUsages	Поиск пересечения KeyUsage для массива сертификатов	
CertSerializeCertificateStoreElement	Сериализация элемента хранилища	
CertComparePublicKeyInfo	Функция CertComparePublicKeyInfo сравнивает два открытых ключа и определяет, являются ли они идентичными	
CertFindExtension	Функция CertFindExtension находит расширение в массиве и возвращает указатель на него.	
CertGetPublicKeyLength	Функция CertGetPublicKeyLength возвращает длину ключа в битах.	

CertGetIntendedKeyUsage	Функция CertGetIntendedKeyUsage получает назначение ключа из сертификата.	
CertCompareCertificateName	Функция CertCompareCertificateName сравнивает два сертификата и определяет, являются ли они идентичными.	
OCSP		
CertAddRefServerOcsponse	Увеличение счетчика ссылок на OCSP ответ	
CertAddRefServerOcsponseContext	Увеличение счетчика ссылок на контекст OCSP ответа	
CertCloseServerOcsponse	Закрытие дескриптора OCSP ответа	
CertGetServerOcsponseContext	Получение контекста OCSP ответа	
CertOpenServerOcsponse	Открытие дескриптора OCSP ответа для заданной цепочки сертификатов	
Оконные функции		
CertSelectCertificate	Отображение диалога выбора сертификата по заданным критериям	
CryptUIDlgCertMgr	Отображение диалога управления сертификатами	
CryptUIDlgSelectCertificate	Отображение диалога выбора сертификата	
CryptUIDlgSelectCertificateFromStore	Отображение диалога выбора сертификата из хранилища	
CryptUIDlgViewCertificate	Отображение диалога со свойствами сертификата	
CryptUIDlgViewContext	Отображение сертификата, списка аннулированных сертификатов или CTL	
CryptUIDlgViewSignerInfo	Отображение диалога с информацией о подписавшем	

CertSelectionGetSerializedBlob	Сериализация сертификата из структуры, используемой для отображения	
GetFriendlyNameOfCert	Преобразование имени сертификата к «читаемому» виду	
Функции проверки цепочек		
CertVerifyCertificateChainPolicy	Функция CertVerifyCertificateChainPolicy проверяет цепочку сертификатов на достоверность, включая соответствие критерию истинности.	
CertGetCertificateChain	Функция CertGetCertificateChain строит цепочку сертификатов, начиная с последнего сертификата, в обратном направлении до доверенного корневого сертификата, если это возможно.	
CertFreeCertificateChain	Функция CertFreeCertificateChain освобождает цепочку сертификатов путем уменьшения счетчика ссылок. Если счетчик ссылок равен нулю, то память, выделенная под цепочку, освобождается.	
CertCreateCertificateChainEngine	Функция CertCreateCertificateChainEngine создает контекст HCERTCHAINENGINE, который позволяет изменять параметры механизма построения цепочки сертификатов. Позволяет ограничивать множество доверенных сертификатов.	
CertFreeCertificateChainEngine	Функция CertFreeCertificateChainEngine освобождает контекст HCERTCHAINENGINE.	
CertCreateCTLEntryFromCertificateContextProperties	Создание CTL на основе свойств атрибутов контекста сертификата	

CertDuplicateCertificateChain	Дублирование контекста цепочки.	
CertFindChainInStore	Функция построения цепочки по заданным критериям из хранилища	
CertFreeCertificateChainList	Функция освобождения массива цепочек	
CertIsValidCRLForCertificate	Функция проверки наличия сертификата в списке аннулированных сертификатов	
CertSetCertificateContextPropertiesFromCTLEntry	Установка свойств в контекст сертификата на основе CTL	
Расширенные свойства сертификата (EKU)		
CertGetEnhancedKeyUsage	Функция CertGetEnhancedKeyUsage получает информацию о расширенном использовании ключа из соответствующего расширения или из расширенных свойств сертификата. Расширенное использование ключа служит признаком правомерного использования сертификата.	
CryptAcquireCertificatePrivateKey	Функция CryptAcquireCertificatePrivateKey получает дескриптор HCRYPTPROV и параметр dwKeySpec для определенного контекста сертификата.	
Функции работы с идентификаторами		
CryptFindOIDInfo	Функция CryptFindOIDInfo получает первую предопределенную или зарегистрированную структуру CRYPT_OID_INFO, согласованную с определенным типом ключа и с ключом.	
CryptEnumOIDInfo	Перечисление зарегистрированных идентификаторов и получение	

	информации для них	
Функции работы с хранилищем		
CertOpenStore	Функция CertOpenStore открывает хранилище сертификатов, используя заданный тип провайдера.	
CertDuplicateStore	Функция CertDuplicateStore дублирует дескриптор хранилища, увеличивая счетчик ссылок на хранилища на единицу.	
CertOpenSystemStore	Функция CertOpenSystemStore используется для открытия наиболее часто используемых хранилищ сертификатов.	
CertCloseStore	Функция CertCloseStore закрывает дескриптор хранилища сертификатов и уменьшает счетчик ссылок на хранилища на единицу.	
CertAddStoreToCollection	Добавление хранилища в коллекцию	
CertControlStore	Установка нотификации при различиях в закешированном хранилище и физическом хранилище	
Функции, используемые для работы с открытыми данными и объектами		
CryptImportPublicKeyInfoEx2	Функция CryptImportPublicKeyInfoEx2 импортирует информацию об открытом ключе и возвращает дескриптор открытого ключа.	
CryptImportPublicKeyInfoEx	Функция CryptImportPublicKeyInfoEx импортирует информацию об открытом ключе в CSP и возвращает дескриптор открытого ключа.	
CryptImportPublicKeyInfo	Функция CryptImportPublicKeyInfo преобразовывает и импортирует	

	информацию об открытом ключе в провайдер и возвращает дескриптор открытого ключа.	
CryptExportPublicKeyInfoEx	Функция CryptExportPublicKeyInfoEx экспортирует информацию об открытом ключе, связанную с соответствующим секретным ключом провайдера.	
CryptExportPublicKeyInfo	Функция CryptExportPublicKeyInfo экспортирует информацию об открытом ключе, ассоциированную с соответствующим секретным ключом провайдера.	
CertCompareCertificate	Функция CertCompareCertificate сравнивает два сертификата для того, чтобы определить, являются ли они идентичными.	
CertCompareIntegerBlob	Функция CertCompareIntegerBlob сравнивает два целочисленных блока для определения того, представляют ли они собой два равных числа.	
CryptExportPublicKeyInfoFromBCryptKeyHandle	Экспортирует информацию об открытом ключе, ассоциированную с соответствующим секретным ключом провайдера.	
Функции кодирования/декодирования		
CryptDecodeObject	Функция CryptDecodeObject используется для декодирования сертификатов, списков аннулированных сертификатов (СОС) и запросов на сертификаты.	
CryptDecodeObjectEx	Функция CryptDecodeObjectEx используется для декодирования сертификатов, списков аннулированных сертификатов и запросов на	

	сертификаты	
CryptEncodeObject	Функция CryptEncodeObject используется для кодирования сертификатов, списков аннулированных сертификатов и запросов на сертификаты.	
CryptEncodeObjectEx	Функция CryptEncodeObjectEx используется для кодирования сертификатов, списков аннулированных сертификатов и запросов на сертификаты.	
Получение объектов из удаленных источников		
CryptRetrieveObjectByUrlA	Функция CryptRetrieveObjectByUrlA получает объект инфраструктуры открытых ключей по заданному URL.	
CryptRetrieveObjectByUrlW	Функция CryptRetrieveObjectByUrlW 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.	
CertFindAttribute	Функция производит поиск атрибута сертификата по идентификатору.	
CertGetNameString	Функция получает имя владельца или издателя сертификата.	
CertNameToStr	Функция производит раскодирование имени из ASN структуры в DN (RFC1779).	
CertSaveStore	Функция производит запись хранилища 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 ASN1 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).	
CryptMsgVerifyCountersignatureEncoded	Функция проверяет добавочную подпись. (декодирует структуру SignerInfo, как определено в PKCS#7).	
CryptMsgVerifyCountersignatureEncodedEx	Функция проверяет добавочную подпись. (декодирует структуру SignerInfo, как определено в PKCS#7).	

Таблица 2 — Перечень вызовов КриптоПро JavaCSP

Метод	Описание	Ограничения на использование функции
getInstance() класса KeyPairGenerator	Метод getInstance() создает объект генерации ключевой пары ЭП и VKO (генератор)	
getInstance() класса KeyFactory	Метод getInstance() создаёт объект, создающий объекты секретных и открытых ключей ЭП и VKO на основе ключевых спецификаций, представляющих собой некоторое внутреннее представление ключевой информации	

Метод	Описание	Ограничения на использование функции
getInstance() класса KeyStore	Метод создает объект, осуществляющий доступ к ключевому хранилищу.	
getInstance() класса SecureRandom	Метод создает объект класса генератора случайных чисел	
getInstance() класса Signature	Метод создает объект формирования ЭП в соответствии с указанным идентификатором алгоритма подписи.	
getInstance() класса MessageDigest	Метод создает объект функции хэширования в соответствии с указанным идентификатором алгоритма хэширования	
getInstance() класса KeyAgreement	Метод создает объект, вырабатывающий ключи согласования	
getInstance() класса Mac	Метод создает объект, осуществляющий имитозащиту данных по алгоритму, соответствующему указанному идентификатору.	Разрешена при указании параметра «HMAC_GOSTR3411», «HMAC_GOSTR3411_2012_256» или «HMAC_GOSTR3411_2012_512»
Initialize() класса KeyPairGenerator	Метод initialize() осуществляет операцию изменения набора параметров вырабатываемых ключевых пар ЭП или VKO (алгоритм ЭП/VKO, параметры используемой группы точек эллиптической кривой, алгоритм хэширования, узла замены для алгоритма хэширования).	
generateKeyPair() класса	Метод generateKeyPair() класса	

Метод	Описание	Ограничения на использование функции
KeyPairGenerator	KeyPairGenerator осуществляет генерацию ключевой пары	
generatePublic() класса KeyFactory	Метод создает объект открытого ключа на основе спецификации открытого ключа (объекта класса PublicKeySpec), содержащей ключевой материал, или закодированного ключа (в виде объекта класса X509EncodedKeySpec).	
generatePrivate() класса KeyFactory	Метод создаёт объект секретного ключа на основе спецификации секретного ключа (объекта класса PrivateKeySpec), содержащей ключевой материал	
Load() класса KeyStore	Метод осуществляет загрузку содержимого стандартного хранилища JCA в память.	
getName() класса KeyStore	Метод возвращает имя хранилища JCA.	
getEntry() класса KeyStore	Метод возвращает дескриптор ключа для дальнейшей работы с ключом при указании верного пароля.	
setKeyEntry() класса KeyStore	Метод осуществляет запись секретного ключа на носитель	
store() класса KeyStore	Метод осуществляет перезапись стандартного хранилища JCA	
getKey() класса KeyStore	Метод осуществляет чтение ключа ЭП с носителя	
setCertificateEntry() класса	Метод осуществляет запись	

Метод	Описание	Ограничения на использование функции
KeyStore	сертификата ключа проверки ЭП на носитель Запись сертификата в хранилище	
getCertificate() класса KeyStore	Метод осуществляет чтение сертификата ключа проверки ЭП с носителя Чтение сертификата из хранилища	
deleteEntry() класса KeyStore	Метод производит удаление ключевого контейнера с носителя	
reset() класса MessageDigest	Метод переинициализирует объект хэширования для повторного использования после получения хэш-значения предыдущего сообщения. Метод может также изменять параметры хэширования (OID) (при использовании алгоритма хэширования ГОСТ Р 34.11-94)	
Clone() класса MessageDigest	Метод создает точную копию существующего объекта хэширования	
Update() класса MessageDigest	Метод осуществляет обработку хэшируемых данных	
read() класса DigestInputStream	Метод осуществляет обработку хэшируемых данных, получаемых из потока	
digest() класса MessageDigest	Метод завершает операцию хэширования и получает хэш-значение сообщения	
valid() класса	Метод, проверяющий срок	

Метод	Описание	Ограничения на использование функции
PrivateKeyUsageExtension	действия ключа электронной подписи	
initSign() класса Signature	Метод устанавливает секретный ключ подписания и параметры ЭП	
initVerify() класса Signature	Метод устанавливает открытый ключ проверки электронной подписи и параметры ЭП	Метод разрешается использовать только для объектов открытых ключей, полученных с помощью механизмов РКIX.
setParameter() класса Signature	Метод устанавливает используемую хэш-функцию и её параметры в соответствии с переданным идентификатором.	
update() класса Signature	Метод осуществляет обработку переданных данных хэш-функцией.	
sign() класса Signature	Метод производит завершающее преобразование хэш-функции и производит подпись всего накопленного сообщения.	
verify() класса Signature	Метод производит завершающее преобразование хэш-функции и осуществляет проверку подписи	
generateCertificate() класса CertificateFactory	Метод производит генерацию X.509 сертификатов	
getEncoded() класса Certificate, наследуется X509Certificate	Метод осуществляет кодирование существующего сертификата	
getPublicKey() класса Certificate, наследуется	Метод возвращает ключ проверки ЭП из сертификата	

Метод	Описание	Ограничения на использование функции
X509Certificate		
checkValidity() класса X509Certificate	Метод осуществляет проверку срока действия сертификата X509	
getVersion() класса X509Certificate	Метод возвращает номер версии сертификата X509	
getSerialNumber() класса X509Certificate	Метод возвращает серийный номер сертификата X509	
getIssuerDN() класса X509Certificate	Метод возвращает DN эмитента сертификата	
getIssuerX500Principal() класса X509Certificate	Метод возвращает DN эмитента сертификата в формате X500	
getSubjectDN() класса X509Certificate	Метод возвращает DN владельца сертификата	
getSubjectX500Principal() класса X509Certificate	Метод возвращает DN владельца сертификата в формате X500	
getNotBefore() класса X509Certificate	Метод возвращает дату начала действия сертификата	
getNotAfter() класса X509Certificate	Метод возвращает дату окончания действия сертификата	
getTBSCertificate() класса X509Certificate	Метод осуществляет DER-кодирование сертификата	
getSignature() класса X509Certificate	Метод возвращает подпись под сертификатом	
getSigAlgName() класса X509Certificate	Метод возвращает название алгоритма подписи под сертификатом	

Метод	Описание	Ограничения на использование функции
getSigAlgOID() класса X509Certificate	Метод возвращает OID алгоритма подписи под сертификатом	
getSigAlgParams() класса X509Certificate	Метод возвращает параметры алгоритма подписи под сертификатом в DER-кодировке	
getIssuerUniqueID() класса X509Certificate	Метод возвращает уникальный ID эмитента сертификата	
getSubjectUniqueID() класса X509Certificate	Метод возвращает уникальный ID владельца сертификата	
getKeyUsage() класса X509Certificate	Метод возвращает набор разрешенных областей использования ключа	
getExtendedKeyUsage() класса X509Certificate	Метод возвращает области расширенного использования ключа	
getBasicConstraints() класса X509Certificate	Метод возвращает длину расширения BasicConstraints	
getIssuerAlternativeNames() класса X509Certificate	Метод возвращает список альтернативных имён эмитента сертификата	
getSubjectAlternativeNames() класса X509Certificate	Метод возвращает список альтернативных имён владельца сертификата	
getOID() интерфейса ParamsInterface	Метод возвращает объектный идентификатор параметров алгоритма ЭП/ВКО/хэширования/узлов замены шифрования.	
getDefault() интерфейса ParamsInterface	Метод возвращает значение объектного идентификатора,	

Метод	Описание	Ограничения на использование функции
	установленного в контрольной панели, параметров алгоритма ЭП/ВКО/хэширования/узлов замены шифрования.	
setDefaultAvailable() интерфейса ParamsInterface	Метод осуществляет проверку наличия необходимых прав для установки новых параметров по умолчанию в контрольную панель.	
setDefault(OID def) интерфейса ParamsInterface	Метод устанавливает объектный идентификатор по умолчанию для параметров алгоритма ЭП/ВКО/хэширования/узлов замены шифрования.	
getOIDs() интерфейса ParamsInterface	Метод получает список допустимых объектный идентификаторов параметров для алгоритма ЭП/ВКО/хэширования/узлов замены шифрования.	
getDefaultSignParams() класса AlgIdSpec	Метод возвращает параметры алгоритма подписи по умолчанию	
getDefaultDigestParams() класса AlgIdSpec	Метод возвращает параметры алгоритма хэширования по умолчанию	
getDefaultCryptParams класса AlgIdSpec	Метод возвращает параметры алгоритма шифрования по умолчанию	
setKeyUsage() класса GostCertificateRequest	Метод устанавливает значение поля keyUsage в запросе на сертификат, описывающее разрешенные области использования ключа.	

Метод	Описание	Ограничения на использование функции
addExtKeyUsage() класса GostCertificateRequest	Метод устанавливает значение поля extKeyUsage в запросе на сертификат, описывающее дополнительные области использования ключа.	
addExtension() класса GostCertificateRequest	Метод устанавливает дополнительное расширение в список расширений.	
setPublicKeyInfo() класса GostCertificateRequest	Метод осуществляет кодирование и запись в структуру запроса параметров и значения передаваемого открытого ключа субъекта	
setSubjectInfo() класса GostCertificateRequest	Метод определяет имя субъекта в формате X.500, устанавливает новое имя субъекта.	
encodeAndSign() класса GostCertificateRequest	Метод выполняет кодирование запроса в DER-кодировку и подпись сертификата.	
getEncodedCert() класса GostCertificateRequest	Метод отправляет запрос центру сертификации и получает соответствующий запросу сертификат	
getEncodedCertFromDER() класса GostCertificateRequest	Метод отправляет запрос в DER-кодировке центру сертификации и получает соответствующий запросу сертификата	
printToDER() класса GostCertificateRequest	Метод осуществляет печать подписанного запроса в DER-кодировке в передаваемый PrintStream	
getEncodedCertFromBASE64() класса GostCertificateRequest	Метод осуществляет отправку запроса в кодировке BASE64	

Метод	Описание	Ограничения на использование функции
	центру сертификации и получение соответствующего запросу сертификата	
printToBASE64() класса GostCertificateRequest	Метод осуществляет печать подписанного запроса в BASE64-кодировке в передаваемый PrintStream	
getEncodedRootCert() класса GostCertificateRequest	Метод запрашивает и получает корневой сертификат центра сертификации	
getRootCertList() класса CA15GostCertificateRequest	Метод получает список корневых сертификатов УЦ (CA15)	
sendCertificateRequest () класса CA15GostCertificateRequest	Метод отправляет в УЦ запрос на сертификат.	
getCertificateRequestList () класса CA15GostCertificateRequest	Метод получает список запросов на сертификат и статуса их обработки.	
checkCertificateStatus () класса CA15GostCertificateRequest	Метод получает статус обработки УЦ ранее отправленного запроса на сертификат.	
getCertificateRequestId() класса CA15GostCertificateRequest	Метод получает строки-идентификаторы запроса на сертификат.	
getCertificateByRequestId() класса CA15GostCertificateRequest	Метод получает выпущенный сертификат по идентификатору запроса.	
init() класса Mac	Метод, устанавливающий алгоритм шифрования	

Метод	Описание	Ограничения на использование функции
clone() класса Mac	Метод, производящий копирование объекта имитозащиты	
update() класса Mac	Метод, осуществляющий обработку данных в процессе вычисления имитовставки	
doFinal() класса Mac	Метод, завершающий вычисление значения имитовставки	
reset() класса Mac	Метод, осуществляющий переинициализацию объекта класса Mac после окончания вычисления значения имитовставки предыдущего сообщения	
verify() класса CAdESSignature	Метод осуществляет проверку всех подписей CAdES	
getCAdESSignerInfos() класса CAdESSignature	Метод получает список всех подписантов	
setCertificateStore() класса CAdESSignature	Метод устанавливает набор сертификатов, которые будут упакованы в подписанное сообщение	
setCRLStore() класса CAdESSignature	Метод устанавливает набор списков отозванных сертификатов, которые будут упакованы в подписанное сообщение	
addSigner() класса CAdESSignature	Метод осуществляет добавление подписи в формируемое подписанное сообщение.	
open() класса CAdESSignature	Метод осуществляет открытие	

Метод	Описание	Ограничения на использование функции
	потока подписываемых данных	
update() класса CAdESSignature	Метод осуществляет обработку порции данных	
close() класса CAdESSignature	Метод осуществляет завершение обработки подписываемых данных.	
replaceSigners() класса CAdESSignature	Метод, осуществляющий изменения данных подписывающих в упакованном подписанном сообщении.	
addRecipient() класса EnvelopedSignature	Метод, добавляющий информацию о получателе сообщения, и подготавливающий сообщение к зашифрованию.	
addKeyTransRecipient() класса EnvelopedSignature	Метод добавляет переданный сертификат в список информации о получателе CMS сообщения в виде структуры key_transport.	Метод разрешается использовать только при осуществлении проверки цепочки сертификата с помощью механизмов РКIX и проверки области действия сертификата.
addKeyAgreeRecipient() класса EnvelopedSignature	Метод добавляет переданный сертификат в список информации о получателе CMS сообщения в виде структуры key_agreement.	Метод разрешается использовать только при осуществлении проверки цепочки сертификата с помощью механизмов РКIX и проверки области действия сертификата.
open() класса EnvelopedSignature	Метод, осуществляющий открытие потока зашифрования сообщения	
update() класса EnvelopedSignature	Метод, осуществляющий шифрование порции данных.	

Метод	Описание	Ограничения на использование функции
close() класса EnvelopedSignature	Метод, осуществляющий шифрование финальной порции данных и закрывающий поток.	
getRecipients() класса EnvelopedSignature	Метод, возвращающий список получателей сообщения	
decrypt() класса EnvelopedSignature	Метод, позволяющий получателю расшифровать сообщение	
getCAAdESSignatureType() класса CAAdESType	Метод возвращает тип CAAdES сообщения	
getSignerInfo() класса CAAdESigner	Метод возвращает информацию о подписанте	
getSignerSignedAttributes() класса CAAdESigner	Метод возвращает список подписываемых атрибутов сообщения	
getSignerUnsignedAttributes() класса CAAdESigner	Метод возвращает список неподписываемых атрибутов сообщения	
getSignatureTimestampToken() класса CAAdESigner	Метод возвращает внутренний штамп (signature-timestamp) времени сообщения	
getCAAdESCTimestampToken() класса CAAdESigner	Метод возвращает внешний штамп (CAAdES-C-timestamp) времени сообщения	
getCAAdESCertificates() класса CAAdESigner	Метод возвращает список сертификатов (certificate-values)	
getSignerCertificate() класса CAAdESigner	Метод возвращает сертификат подписавшего сообщение	
getSignatureType() класса	Метод возвращает тип подписи	

Метод	Описание	Ограничения на использование функции
CAdESSigner		
getCAdESCountersignerInfos() класса CAdESSigner	Метод возвращает список заверителей сообщения	
setCertificateStore() класса CAdESSigner	Метод устанавливает хранилище сертификатов, из которого позже может быть достан сертификат подписи	
enhance() класса CAdESSigner	Метод осуществляет «усовершенствование» подписи CAdES-BES до CAdES-X Long Type 1	
addCountersigner() класса CAdESSigner	Метод, осуществляющий добавление заверяющей подписи к отдельному подписанту.	
verify() класса CAdESSigner	Метод, осуществляющий проверку одной отдельной подписи CAdES	
build() класса CertPathBuilder	Метод, осуществляющий построение цепочки сертификатов	
getAlgorithm() класса CertPathValidator	Метод, возвращающий имя алгоритма, осуществляющего проверку цепочек сертификатов.	
getDefaultType() класса CertPathValidator	Метод, возвращающий имя алгоритма, осуществляющего проверку цепочек сертификатов, принятого по умолчанию.	
getInstance() класса CertPathValidator	Метод, возвращающий объект класса CertPathValidator.	

Метод	Описание	Ограничения на использование функции
getProvider() класса CertPathValidator	Метод, осуществляющий получение объекта провайдера алгоритма, осуществляющего проверку цепочек сертификатов, принятого по умолчанию.	
validate() класса CertPathValidator	Метод, осуществляющий проверку цепочек сертификатов.	
addCertPathChecker() класса CertPathParameters	Метод, устанавливающий дополнительные правила проверки сертификатов.	
addCertStore() класса CertPathParameters	Метод, устанавливающий дополнительные хранилища сертификатов и CRL.	
getCertPathCheckers() класса CertPathParameters	Метод, возвращающий список установленных дополнительных правил проверки сертификатов.	
getCertStores() класса CertPathParameters	Метод, возвращающий список хранилищ сертификатов и CRL.	
getDate() класса CertPathParameters	Метод, возвращающий дату, на которую проверяется верность цепочки сертификатов.	
getInitialPolicies() класса CertPathParameters	Метод, возвращающий список OID'ов политик, которые должны быть указаны в сертификатах, из которых строится цепочка.	
getPolicyQualifiersRejected() класса CertPathParameters	Метод, возвращающий флаг PolicyQualifiersRejected.	
getSigProvider() класса	Метод, получающий имя	

Метод	Описание	Ограничения на использование функции
CertPathParameters	провайдера ЭП.	
getTargetCertConstraints() класса CertPathParameters	Метод, возвращающий ограничения на целевой сертификат.	
getTrustAnchors() класса CertPathParameters	Метод, получающий список доверенных сертификатов.	
isAnyPolicyInhibited() класса CertPathParameters	Метод, возвращающий признак обработки всех политик, указанных в сертификате.	
isExplicitPolicyRequired() класса CertPathParameters	Метод, возвращающий признак наличия явных описаний политик.	
isPolicyMappingInhibited() класса CertPathParameters	Метод, возвращающий признак подавления PolicyMapping.	
isRevocationEnabled() класса CertPathParameters	Метод, устанавливающий признак необходимости проверки сертификатов с помощью OCSP/CRL.	
setAnyPolicyInhibited() класса CertPathParameters	Метод, устанавливающий признак обработки всех политик, указанных в сертификате.	
setCertPathCheckers() класса CertPathParameters	Метод, устанавливающий набор дополнительных правил проверки сертификатов.	
setCertStores() класса CertPathParameters	Метод, устанавливающий набор дополнительных хранилищ сертификатов и CRL.	
setDate() класса CertPathParameters	Метод, устанавливающий дату, на которую должна производиться проверка	

Метод	Описание	Ограничения на использование функции
	цепочки сертификатов.	
setExplicitPolicyRequired() класса CertPathParameters	Метод, устанавливающий признак наличия явных описаний политик.	
setInitialPolicies() класса CertPathParameters	Метод, устанавливающий список OID'ов политик, которые должны быть указаны в сертификатах, из которых строится цепочка.	
setPolicyMappingInhibited() класса CertPathParameters	Метод, устанавливающий признак подавления PolicyMapping.	
setPolicyQualifiersRejected() класса CertPathParameters	Устанавливает флаг PolicyQualifiersRejected.	
setRevocationEnabled() класса CertPathParameters	Метод, устанавливающий признак необходимости проверки сертификатов с помощью OCSP/CRL.	
setTargetCertConstraints() класса CertPathParameters	Метод, устанавливающий ограничения на целевой сертификат.	
setTrustAnchors() класса CertPathParameters	Метод, устанавливающий список доверенных сертификатов.	
getMaxPathLength() класса PKIXBuilderParameters	Метод, получающий максимальную возможную длину строимой цепочки.	
setMaxPathLength() класса PKIXBuilderParameters	Метод, устанавливающий максимальную возможную длину строимой цепочки.	
getXAdESSignerInfos() класса XAdESSignature	Метод получает список всех подписантов	

Метод	Описание	Ограничения на использование функции
addSigner() класса XAdESSignature	Метод осуществляет добавление подписи в формируемое подписанное XML сообщение.	
open() класса XAdESSignature	Метод осуществляет открытие потока подписываемых данных	
update() класса XAdESSignature	Метод осуществляет обработку порции данных	
close() класса XAdESSignature	Метод осуществляет завершение обработки подписываемых данных.	
verify() класса XAdESSignature	Метод осуществляет проверку всех подписей XAdES	
getSignatureType() класса XAdESSigner	Метод, возвращающий типа XAdES-подписи.	
getEarliestValidSignatureTimeStampToken() класса XAdESSignerT	Метод, возвращающий самый ранний валидный внутренний штамп времени.	
verify() класса XAdESSigner	Метод, осуществляющий проверку одной отдельной подписи XAdES	
getSignerInfo() класса XAdESSigner	Метод, возвращающий узел подписи в документе.	
getSignerCertificate() класса XAdESSigner	Метод, возвращающий сертификат ключа ЭП данного подписанта.	
getElement() класса XAdESSigner	Метод, возвращающий подписываемый узел.	
getDocument()	Метод, возвращающий	

Метод	Описание	Ограничения на использование функции
класса XAdESSigner	подписываемый документ.	
getPrivateKey() класса JCPPPrivateKeyEntry	Метод, возвращающий дескриптор ключа privKey.	
isExportable() класса JCPPPrivateKeyEntry	Метод, возвращающий значение флага экспортируемости закрытого ключа/ключа ЭП.	
getCertificateChain() класса JCPPPrivateKeyEntry	Метод, возвращающий цепочку сертификатов.	
getCertificate() класса JCPPPrivateKeyEntry	Метод, возвращающий клиентский сертификат (первый в цепочке сертификатов).	
toString() класса JCPPPrivateKeyEntry	Метод, возвращающий строковое представление цепочки сертификатов.	
isSilentMode() класса JCPPProtectionParameter	Метод, возвращающий режим открытия ключевого контейнера.	
isAllowEmptyChain() класса JCPPProtectionParameter	Метод, возвращающий разрешение использовать null вместо цепочки сертификатов.	
getKeyType() класса JCPPProtectionParameter	Метод, возвращающий тип ключа.	

Метод	Описание	Ограничения на использование функции
sign() класса XMLSignature	Метод создает ЭП сообщения.	Разрешено использование только со следующими URI/URN: <ul style="list-style-type: none">• http://www.w3.org/2001/04/xmldsig-more#gostr34102001-gostr3411• urn:ietf:params:xml:ns:cpxmlsec:algorithms:gostr34102001-gostr3411• urn:ietf:params:xml:ns:cpxmlsec:algorithms:gostr34102012-gostr34112012-256• urn:ietf:params:xml:ns:cpxmlsec:algorithms:gostr34102012-gostr34112012-512
checkSignatureValue() класса XMLSignature	Метод осуществляет проверку ЭП сообщения.	Метод разрешается использовать только при осуществлении проверки сертификата открытого ключа с помощью механизмов РКIX.