

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



УТВЕРЖДЕНЫ
ЖТЯИ.00096-03 95 01-ЛУ

Средство Криптографической Защиты Информации	КриптоПро HSM версия 2.0 R3 Комплектации 1 и 2 Правила пользования
---	---

ЖТЯИ.00096-03 95 01

Листов 70

2023 г.

Содержание

Аннотация.....	3
Перечень используемых терминов и сокращений.....	4
1 Назначение ПАКМ и его основные характеристики	5
1.1 Назначение и основные функции СКЗИ.....	5
1.2 Комплектации ПАКМ.....	6
1.3 Аппаратный компонент ПАКМ.....	7
1.3.1 Комплектация 1 Исполнение 1К	7
1.3.2 Комплектация 1 Исполнение 5К	7
1.4 Программный компонент ПАКМ	7
2 Ключевая система и ключевые документы.....	9
2.1 Типы ключей	9
2.2 Сроки действия ключей.....	9
2.3 Хранение и использование ключей.....	9
2.3.1 Требования к PIN ключевых носителей.....	10
2.3.2 Маркировка ключевых носителей.....	10
2.3.3 Хранение и использование ключевых носителей	10
2.3.4 Уничтожение ключей	10
2.3.5 Контроль сроков действия ключей	11
2.3.6 Режим усиленного контроля использования ключей.....	11
2.3.7 Порядок действий при компрометации ключей	11
3 Требования по размещению технических средств.....	13
4 Требования к использованию аппаратных компонент ПАКМ	16
5 Требования по обеспечению безопасности при эксплуатации ПАКМ.....	17
5.1 Общие требования по защите от НСД.....	17
5.1.1 Принципы защиты информации от НСД	17
5.1.2 Применяемая модель защиты.....	17
5.2 Требования к аутентификации и разграничению доступа	18
5.2.1 Ролевая модель доступа к функциям ПАКМ	19
5.3 Требования по выполнению технического обслуживания и ремонта ПАКМ	19
5.4 Требования по обеспечению целостности.....	20
5.4.1 Контроль целостности ПО ПАКМ.....	20
5.5 Требования к ПЭВМ с установленным клиентским компонентом ПАКМ	21
5.6 Ежесуточное нагрузочное тестирование ПАКМ.....	21
5.7 Использование аппаратно-программного модуля доверенной загрузки	22
5.8 Защита от вскрытия и проникновения в корпус ПАКМ	23
6 Эксплуатация ПАКМ	24
7 Резервирование и восстановление ПАКМ.....	25
7.1 Холодное резервирование ПАКМ	25
7.2 Горячее резервирование ПАКМ	26
8 Контроль установок времени	30
9 Требования по криптографической защите.....	31
10 Требования к встраиванию СКЗИ в прикладные системы и проведению исследований	
СФ СКЗИ.....	32
10.1 Применение ПАКМ без исследований СФ	33
10.2 Применение ПАКМ с проведением исследований по оценке влияния.....	33
10.3 Применение ПАКМ с проведением тематических исследований.....	34
10.4 Требования при встраивании ПАКМ в прикладные системы	34
Литература	37
Приложение 1. Перечень вызовов, использование которых при разработке систем на основе	
ПАКМ «КриптоПро HSM» всех комплектаций с учетом п.1.5 Формуляра ЖТЯИ.00096-03 30 01 возможно	
без дополнительных тематических исследований	38
Интерфейс CryptoAPI.....	38
Интерфейс JCA/JCE (КриптоПро JavaCSP)	56

Аннотация

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

Документ предназначен для администраторов информационной безопасности учреждений, осуществляющих установку, обслуживание и контроль за соблюдением требований к эксплуатации ПАКМ, 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.

Перечень используемых терминов и сокращений

АС	—	аппаратные средства
АПМДЗ	—	аппаратно-программный модуль доверенной загрузки
ДСЧ	—	датчик случайных чисел
ИБП	—	источник бесперебойного питания
ИС	—	информационная система
НСД	—	несанкционированный доступ
ОЗУ	—	оперативное запоминающее устройство
ОС	—	операционная система
ПАК	—	программно-аппаратный комплекс
ПАКМ	—	программно-аппаратный криптографический модуль
ПО	—	программное обеспечение
ПЭВМ	—	персональная электронно-вычислительная машина
СВТ	—	средства вычислительной техники
СКЗИ	—	средство криптографической защиты информации
СФ	—	среда функционирования
ТЗ	—	техническое задание
ЭП	—	электронная подпись

1 Назначение ПАКМ и его основные характеристики

1.1 Назначение и основные функции СКЗИ

ПАКМ «КриптоПро HSM» версии 2.0 R3 — 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 «КриптоПро HSM» может быть использован в качестве СКЗИ в различных системах/подсистемах криптографической защиты информации, поддерживающих криптографические интерфейсы «КриптоПро CSP».

ПАКМ «КриптоПро HSM» версии 2.0 R3 предназначен для выполнения следующих функций:

- создание, хранение и управление ключевой информацией;
- шифрование/расшифрование файлов и областей памяти;
- вычисление имитовставки для файлов и областей памяти;
- создание/проверка электронной подписи;
- вычисление значения хэш-функции для файлов и областей памяти;
- защита TLS-соединений;
- криптографическая аутентификация абонентов при установлении соединения;
- обеспечение защиты чувствительной информации, обрабатываемой в технических средствах информационной инфраструктуры платежной системы.

При этом ПАКМ «КриптоПро HSM» обеспечивает:

- реализацию криптографических функций и интерфейса взаимодействия ПАКМ с серверами и клиентскими компонентами ПАКМ пользователей;
- интерфейс к прикладным криптографическим функциям в соответствии со спецификацией интерфейса СКЗИ «КриптоПро CSP»;
- возможность использования функций ПАКМ через интерфейсы Microsoft CryptoAPI, JCA/JCE;
- идентификацию и аутентификацию пользователей при локальном и удаленном доступе к ПАКМ;
- проверку целостности критичного к безопасному функционированию ПО при инициализации ПАКМ;
- генерацию случайных чисел с использованием аппаратного ДСЧ;
- создание ключей ЭП и шифрования с использованием исходного материала, предоставленного уполномоченной организацией;
- сопряжение ПАКМ с устройством доступа по криптографически защищенным каналам «К»¹, «K2», «K2s»;

¹ Канал «К» используется только в рамках Головного удостоверяющего центра

- сопряжение ПАКМ с сервером/группой серверов по отдельному сегменту Ethernet;
- возможность удаленного Web администрирования ПАКМ;
- запись и ввод ключей с ключевых носителей на смарт-картах;
- создание и проверка ЭП согласно ГОСТ Р 34.10-2012 (ГОСТ 34.10-2018), проверка ЭП согласно ГОСТ Р 34.10-2001;
- вычисление значений хэш-функции согласно ГОСТ Р 34.11-94 и ГОСТ Р 34.11-2012 (ГОСТ 34.11-2018);
- шифрование и имитозащита данных согласно ГОСТ 28147-89, ГОСТ Р 34.12-2015 (ГОСТ 34.12-2018), ГОСТ Р 34.13-2015 (ГОСТ 34.13-2018);
- сетевая аутентификация на базе протокола TLS версий 1.0, 1.1, 1.2 в соответствии с методическими рекомендациями МР 26.2.001-2013 и рекомендациями по стандартизации Р 1323565.1.020-2018, Р 1323565.1.020-2020;
- формирование защищенных сообщений в формате CMS в соответствии с методическими рекомендациями МР 26.2.002-2013 и рекомендациями по стандартизации Р 1323565.1.025-2019;
- возможность встречной работы ПАКМ «КриптоПро HSM» с СКЗИ «КриптоПро CSP»;
- регистрацию событий в журнале аудита криптографических вызовов ПАКМ;
- инициализацию и эмиссию платёжных карт с магнитной полосой, бесконтактных и смарт-карт, включая генерацию PIN/CVC/CVP и печать PIN-конвертов, смену и проверку PIN;
- управление платёжными ключами и обеспечение безопасности ключей на всех этапах жизненного цикла (включая генерацию, печать ключевых компонент, формирование ключей из компонент, диверсификацию ключей, импорт/экспорт, трансляцию и удаление из хранилища);
- обработку платёжных транзакций, банковских транзакций от платёжных устройств.

1.2 Комплектации ПАКМ

ПАКМ «КриптоПро HSM» версии 2.0 R3 представлен в следующих комплектациях, сертифицированных по различным уровням защиты:

1. Комплектация 1 (ПАКМ)

- Исполнение 1К — соответствует требованиям к СКЗИ² по классу КВ, требованиям к средствам ЭП³ по классу KB2
- Исполнение 5К — соответствует требованиям к СКЗИ² по классу КСЗ, требованиям к средствам ЭП³ по классу КСЗ

2. Комплектация 2 (ПАКМ и клиентский компонент ПАКМ)

- Исполнение 1К — соответствует требованиям к СКЗИ² по классу КС1, требованиям к средствам ЭП³ по классу КС1

² Требования к средствам криптографической защиты информации, предназначенным для защиты информации, не содержащей сведений, составляющих государственную тайну

³ Требования к средствам электронной подписи, утверждённые приказом ФСБ России от 27 декабря 2011г. № 796

- Исполнение 2К — соответствует требованиям к СКЗИ² по классу КС2, требованиям к средствам ЭП³ по классу КС2
- Исполнение 3К — соответствует требованиям к СКЗИ² по классу КС3, требованиям к средствам ЭП³ по классу КС3

Состав комплектаций и исполнений ПАКМ описан в разделе «Комплектность» документа «ЖТЯИ.00096-03 30 01. Формуляр».

1.3 Аппаратный компонент ПАКМ

1.3.1 Комплектация 1 Исполнение 1К

В качестве аппаратной платформы ПАКМ «КриптоПро HSM» версия 2.0 R3 Комплектация 1 Исполнение 1К (класс защиты КВ/КВ2) используется серверная платформа Advantech. Ее технические характеристики отображены в таблице ниже.

Корпус	Серверный промышленный корпус Advantech для монтажа в стойку 19", высота 2U, глубина 52 см, с двумя вентиляторами охлаждения, смонтированными внутри
Процессоры	INTEL XEON E5-2620V3 2.4 GHz, 2 шт.
Сетевая карта	Allied Telesis, Gigabit Ethernet Fiber Adapter, 1000-Base-SX двойной
Блок питания	Advantech, 500 Вт двойной с горячей заменой
Разъемы электропитания	C14, 2 шт.
Энергопотребление	До 300 Вт
Напряжение	220 В

1.3.2 Комплектация 1 Исполнение 5К

В качестве аппаратной платформы ПАКМ «КриптоПро HSM» версия 2.0 R3 Комплектация 1 Исполнение 5К (класс защиты КС3) используются серверные платформы производства ООО «КРИПТО-ПРО».

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

1.4 Программный компонент ПАКМ

Программный компонент ПАКМ включает:

- системное программное обеспечение — ОС «Альт Линукс СПТ 7.0» с защитой ядра ОС средствами пакета GRSecurity;
- программное обеспечение ПАКМ (ПО ПАКМ).

Комплектация 1 (ПАКМ) включает следующее ПО ПАКМ:

1. Базовые программные модули

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

2. Модуль обеспечения защиты платежных систем

Модуль реализует интерфейс поддержки операций системы платёжных карт и предназначен для обеспечения защиты чувствительной информации, обрабатываемой в платёжной системе, и безопасного хранения криптографических платёжных ключей.

3. Модуль обеспечения защиты систем ДЭГ

Модуль расширяет интерфейс базовых модулей ПАКМ в части работы с ключевой информацией, хранящейся в ПАКМ. Данное расширение интерфейса (а также специальным образом использующее данный интерфейс клиентское ПО, не входящее в комплект поставки ПАКМ) необходимо для обеспечения защиты систем ДЭГ.

Комплектация 2 (ПАКМ и клиентский компонент) в дополнение к указанным выше модулям также включает следующее ПО ПАКМ:

1. ПО «Клиент КриптоПро HSM» (КриптоПро HSM Client)

«Клиент КриптоПро HSM» представляет собой отдельную программу для установки в ОС семейства Windows, предназначенную для реализации взаимодействия приложений, установленных на рабочих станциях/серверах под управлением ОС Windows, с ПАКМ «КриптоПро HSM».

Инструкция по установке и настройке ПО «Клиент КриптоПро HSM» под управлением ОС семейства Windows приведена в документе «ЖТЯИ.00096-03 93 01. Руководство пользователя».

2. Утилита командной строки HsmGenArm

Утилита предназначена для формирования и экспорта ключей в формате PKCS#12, функционирует в ОС семейств Windows и Linux. Инструкция по использованию утилиты приведена в Приложении 2 документа «ЖТЯИ.00096-03 90 02. Использование интерфейсных модулей».

3. СКЗИ «КриптоПро CSP»

СКЗИ «КриптоПро CSP» устанавливается в составе клиентского компонента ПАКМ и реализует криптографический интерфейс обмена данными между устройствами доступа и ПАКМ.

4. ПО «КриптоПро JavaCSP», «КриптоПро JavaTLS» (только Исполнения 1К, 2К)

ПО «КриптоПро JavaCSP», «КриптоПро JavaTLS» функционирует под управлением Java-машины и реализует стандартный интерфейс JCA/JCE в соответствии с российскими криптографическими алгоритмами.

2 Ключевая система и ключевые документы

2.1 Типы ключей

Ключевая система ПАКМ «КриптоПро HSM» включает:

1. Ключи ЭП

Ключи ЭП представляются ключевой парой: ключ ЭП (для формирования ЭП) и ключ проверки ЭП.

2. Ключи шифрования

Симметричные ключи сообщения (пакета), случайные или диверсифицированные из случайного ключа сессии по открытому заголовку сообщения (пакета).

3. Ключи обмена (экспорта ключей)

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).

По назначению различают ключи пользователей ПАКМ, используемые для выполнения криптографических операций по запросу пользователей, и служебные ключи самого ПАКМ, обеспечивающие безопасное хранение и управление ключевой информацией. Описание ключей ПАКМ приведено в документе «ЖТЯИ.00096-03 91 01. Руководство администратор безопасности».

2.2 Сроки действия ключей

При эксплуатации ПАКМ должны соблюдаться следующие сроки использования⁴ пользовательских закрытых ключей и сертификатов:

- ключ ЭП (неэкспортируемый) — не более 3 лет;
- ключ проверки ЭП — не более 15 лет после окончания срока действия соответствующего ключа ЭП;
- открытый ключ обмена — не более 3 лет;
- закрытый ключ обмена (неэкспортируемый) — не более 3 лет;
- иные ключи — не более 1 года 3 месяцев.

2.3 Хранение и использование ключей

Ключи могут храниться в памяти ПАКМ или на ключевых носителях.

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

Ключи, хранящиеся на ключевых носителях, используются для удаленной и локальной аутентификации пользователей ПАКМ, а также для защиты ключа активации ПАКМ и LMK. На ключевых носителях ключи хранятся в формате ключевого

⁴ 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.

контейнера. Ключевой контейнер содержит также информацию, необходимую для обеспечения криптографической защиты ключей и их целостности.

2.3.1 Требования к PIN ключевых носителей

Смарт-карты поставляются с ПАКМ «КриптоПро HSM» отформатированными, с предустановленным PIN = 11111111. При записи ключей на карту PIN необходимо сменить.

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

Периодичность плановой смены PIN составляет 1 месяц.

2.3.2 Маркировка ключевых носителей

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

2.3.3 Хранение и использование ключевых носителей

Личные ключевые носители пользователей (с ключами доступа к ПАКМ) рекомендуется хранить в сейфе. Пользователь несет персональную ответственность за хранение личных ключевых носителей.

Администратор безопасности организации, эксплуатирующей СКЗИ, при централизованном хранении ключевых носителей несет персональную ответственность за хранение личных ключевых носителей пользователей. Личные ключевые носители администратора безопасности должны храниться в его личном сейфе.

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

2.3.4 Уничтожение ключей

Ключи на ключевых носителях (смарт-картах), срок действия которых истек, уничтожаются с использованием ПО СКЗИ. Ключевые носители могут быть использованы в дальнейшем при условии записи на них новой ключевой информации.

Ключи ЭП/закрытые ключи, хранящиеся внутри ПАКМ, уничтожаются пользователями с использованием штатных средств ПО клиентского компонента ПАКМ.

2.3.5 Контроль сроков действия ключей

В ПАКМ «КриптоПро HSM» реализован контроль сроков действия ключей. При попытке использования ключа дольше заданного срока операции подписи и шифрования на нем будут заблокированы. При превышении срока действия служебных (внутренних) ключей ПАКМ может стать нефункциональным.

При превышении срока действия ключа в журнале событий ПАКМ отображается следующая информация:

```
cryptsrv_hsm[41244]: <cpcsp>[0x66973adf6700]SetupAndCheckKeyTimeValidity Key expired:
signing will be forbidden
cryptsrv_hsm[41244]: <cpcsp>[0x66973adf6700]SetupAndCheckKeyTimeValidity Key expired:
encryption will be forbidden
```

2.3.6 Режим усиленного контроля использования ключей

Режим усиленного контроля использования ключей обеспечивает осуществление контроля срока действия долговременных ключей ЭП и ключевого обмена, контроля доверенности ключей проверки ЭП и контроля корректного использования программного ДСЧ.

Использование клиентского компонента ПАКМ с выключенным режимом усиленного контроля использования ключей **не допускается**. Управление режимом усиленного контроля использования ключей (StrengthenedKeyUsageControl) осуществляется в соответствии с документацией на СКЗИ «КриптоПро CSP», входящее в состав клиентского компонента ПАКМ.

2.3.7 Порядок действий при компрометации ключей

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

- 1) потеря ключевых носителей;
- 2) потеря ключевых носителей с их последующим обнаружением;
- 3) увольнение сотрудников, имевших доступ к ключевой информации;
- 4) нарушение правил хранения и уничтожения (после окончания срока действия) закрытого ключа;
- 5) возникновение подозрений на утечку информации или ее искажение в системе конфиденциальной связи;
- 6) нарушение печати на сейфе с ключевыми носителями;
- 7) случаи, когда нельзя достоверно установить, что произошло с ключевыми носителями (в том числе случаи, когда ключевой носитель вышел из строя и доказательно не опровергнута возможность того, что данный факт произошел в результате несанкционированных действий злоумышленника).

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

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.00096-03 91 01. Руководство администратора безопасности»).

Скомпрометированные ключи подлежат замене. При перевыпуске ключей администратором выполняются действия, совершаемые при плановой смене ключей.

При компрометации ключа ЭП ПАКМ, использующегося для издания сертификатов ключей аутентификации пользователей, после смены ключа ЭП ПАКМ должны быть перевыпущены все сертификаты пользователей и соответствующие им ключи.

Ключ шифрования автоматически меняется при смене ключа активации ПАКМ.

3 Требования по размещению технических средств

При размещении технических средств, имеющих в своем составе ПАКМ «КриптоПро HSM» (в комплектации в соответствии с табл. 4.1 «ЖТЯИ.00096-03 30 01. Формуляр»), следует руководствоваться требованиями и рекомендациями, приведенными ниже.

При эксплуатации Комплектации 1 Исполнения 1К ПАКМ (класс защиты KB/KB2):

1. Необходимо обеспечить **контролируемую зону**⁵ радиусом не менее **4 метров** при работе с конфиденциальной информацией без применения дополнительных мер спецзащиты.

2. 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 метра**.

3. 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 метра**.

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 (бензиновых/дизельных 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 метра. 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.

⁵ 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.

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 дверьми 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, а 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 ПАКМ 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 ПАКМ, принимается организацией, обеспечивающей эксплуатацию ПАКМ.

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 клиентский компонент ПАКМ, должны быть допущены установленным порядком для обработки информации ограниченного доступа и иметь соответствующие Предписания на эксплуатацию.

Каналы связи ПЭВМ с ПАКМ и/или клиентским компонентом ПАКМ, 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.

При использовании удаленного рабочего места администрирования ПАКМ (раздел 6 «ЖТЯИ.00096-03 90 01. 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.

4 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 печатающихся наклеек;
- регулярный контроль за состоянием всех кабельных соединений;
- отсутствие возможности подключения к ПАКМ дополнительных устройств и соединителей, не предусмотренных штатной комплектацией;
- исключение организационными мерами нештатного и несанкционированного использования ПАКМ;
- осуществление электропитания ПАКМ через защищенную электрическую сеть, например, через ИБП с гальванической развязкой.

5 Требования по обеспечению безопасности при эксплуатации ПАКМ

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 контролируемую зону, осуществляющие техническое обслуживание, настройку, конфигурирование ПАКМ и управление ключевой системой, относятся к потенциальным нарушителям. Возможность сговора между данными пользователями исключается.

5.1 Общие требования по защите от НСД

5.1.1 Принципы защиты информации от НСД

Защита информации от НСД в автоматизированной системе обеспечивается комплексом программно-технических средств и поддерживающих их организационных мер, который включает в том числе:

- применение специальных программно-аппаратных средств защиты;
- организацию системы контроля безопасности информации;
- физическую охрану ПЭВМ и ее средств;
- администрирование информационной безопасности, основанное на разделении ролей административной группы;
- учет носителей информации;
- сигнализацию 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 ПАКМ, должна быть выпущена инструкция по защите от НСД к системе, разработанная на базе настоящего документа, рекомендаций ФСТЭК России и ФСБ России, действующих нормативных документов самой эксплуатирующей организации.

5.1.2 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 безопасности, который осуществляет настройку подсистемы безопасности серверов приложений (например, УЦ) 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 в ПАКМ используются следующие средства и меры защиты:

- шифрование всей ключевой системы ПАКМ на ключах шифрования ПАКМ, которые, в свою очередь, зашифрованы на ключе активации ПАКМ, разделенном по схеме 3 из 5;
- парольная защита ключевых носителей;
- парольная защита входа в ОС для администратора системы, администратора безопасности, оператора;
- штатные средства разграничения доступа ОС серверов и рабочих станций и встроенной ОС ПАКМ;
- ПАК «Соболь» для защиты от несанкционированного входа в систему;
- ролевое разграничение доступа к ПАКМ со стороны привилегированных пользователей ПАКМ;
- криптографическая аутентификация абонентов при установлении TLS-соединения с использованием ключей обмена на ключевых носителях;
- локальная аутентификация привилегированных пользователей ПАКМ (из группы администраторов ПАКМ) с использованием системы аутентификации смарт-карты;
- штатные средства ОС ПАКМ для защиты от воздействий со стороны серверов/рабочих станций с целью НСД по каналам сопряжения с устройством доступа;
- межсетевой экран 4 класса защиты в соответствии с классификацией ФСТЭК для защиты от воздействия по глобальной сети;
- датчик вскрытия корпуса и датчик движения (изменения объема) для защиты от проникновений в аппаратные средства ПАКМ.

5.2 Требования к аутентификации и разграничению доступа

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

Необходимо разработать и применить политику назначения и смены используемых паролей. Пароли должны удовлетворять следующим требованиям:

- длина пароля не менее 6 символов;
- в числе символов пароля обязательно должны присутствовать буквы в верхнем и нижнем регистрах, цифры и специальные символы (@, #, \$, &, *, % и т.п.);

- срок смены пароля не реже 1 раза в месяц;
- личный пароль пользователь не имеет права сообщать никому.

При использовании клиентского компонента ПАКМ должны выполняться требования к аутентификации и разграничению доступа, требования по защите от НСД, приведенные в документации на СКЗИ «КриптоПро CSP», входящее в состав клиентского компонента ПАКМ.

5.2.1 Ролевая модель доступа к функциям ПАКМ

ПАКМ 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 ПАКМ;
- администратор сервера, сервисы которого используют ПАКМ;
- администратор ПАКМ;
- аудитор ПАКМ;
- администратор резервного копирования ПАКМ;
- суперпользователь ПАКМ;
- администратор Статистики;
- администратор Платежной системы;
- держатель части разделенного секрета LMK.

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 разделе 5 документа «ЖТЯИ.00096-03 91 01. Руководство администратора безопасности».

5.3 Требования по выполнению технического обслуживания и ремонта ПАКМ

Все работы по выполнению технического обслуживания и ремонта изделия должны выполняться в соответствии с разделом «Порядок выполнения технического обслуживания и ремонта ПАКМ» документа «ЖТЯИ.00096-03 90 01. Инструкция по использованию».

При эксплуатации ПАКМ запрещается внесение изменений в состав компонент ПАКМ. После ремонта при необходимости проводится специальная проверка и специальные исследования отремонтированного оборудования.

5.4 Требования по обеспечению целостности

При использовании клиентского компонента ПАКМ должны выполняться требования по обеспечению целостности, приведенные в документации на СКЗИ «КриптоПро CSP», входящее в состав клиентского компонента ПАКМ.

При эксплуатации ПАКМ не допускается:

- вносить какие-либо изменения в программное обеспечение ПАКМ;
- эксплуатировать ПАКМ, если во время его начальной загрузки не проходит встроенный тест ОЗУ;
- осуществлять несанкционированное вскрытие системного блока ПАКМ.

5.4.1 Контроль целостности ПО ПАКМ

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 ПАК «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 «/»), которая выполняется ПАК «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 АПМДЗ.

Контроль целостности в ПАКМ «КриптоПро HSM» охватывает:

- модули криптопровайдера;
- ядро, модули ядра ОС, их конфигурационные файлы;
- модули каналов сопряжения с устройствами доступа;
- драйверы устройств и портов ввода смарт-карт, com-порта, сети Ethernet;
- драйвер ПАК «Соболь».

Кроме этого проверка контрольных сумм файлов осуществляется по расписанию (1 раз в сутки).

Если проверка дала отрицательный результат, ПАКМ останавливается (выполняется процедура Halt).

5.5 Требования к ПЭВМ с установленным клиентским компонентом ПАКМ

В организации должен быть назначен **администратор безопасности**, функции которого должны заключаться в выполнении процедур установки и настройки ПО клиентского компонента ПАКМ на ПЭВМ, настройки системного окружения, установки, настройки, обслуживания и обеспечения функционирования средств защиты, выработке соответствующих инструкций для пользователей ПАКМ и контроле за соблюдением требований по безопасности.

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

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

При использовании клиентского компонента ПАКМ должны выполняться требования по размещению технических средств с установленным СКЗИ, требования по установке СКЗИ и ПО на ПЭВМ, требования по обеспечению физической безопасности, требования по защите от НСД и требования по подключению СКЗИ для работы по общедоступным каналам передачи данных, приведенные в документации на СКЗИ «КриптоПро CSP», входящее в состав клиентского компонента ПАКМ.

5.6 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.

Периодичность перезагрузки ПАКМ — 1 месяц.

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 сопряжения ПАКМ с устройством доступа, 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 80 -cont "\\.\HSMDB\а"
```

```
csptest -keyset -newkeys -provtype 80 -cont "\\.\HSMDB\б"
```

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

Процедура нагрузочного тестирования запускается следующей командой:

```
csptest -perf -sign -hash -encrypt -data 512 -targettime 250 -provider "Crypto-Pro
GOST R 34.10-2012 HSM CSP" -provtype 80 -sender "\\.\HSMDB\A" -recipient
"\\.\HSMDB\B" -thread 20 -repeat 20 -pinsend 11111111 -pinrecip 11111111 -alghash
GOST12_256
```

Для отображения результатов теста используется стандартный вывод (консоль), который можно перенаправить в файл для последующего анализа. Положительный результат теста – отсутствие сообщений об ошибках как в выводе в окне теста на сервере, так и на LCD панели ПАКМ.

Пример успешного завершения теста можно посмотреть на следующем рисунке:

```
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 USB: 0.391 sec UTC: 118.406 sec
[ErrorCode: 0x00000000]

C:\Program Files\Crypto Pro\CSP>
```

5.7 Использование аппаратно-программного модуля доверенной загрузки

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

АПМДЗ 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 ОС.

Кроме того, АПМДЗ включает в себя физический датчик случайных чисел, используемый криптографическими функциями ПАКМ «КриптоПро HSM».

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

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

Ключи АПМДЗ 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 АПМДЗ относятся к категории ключевых носителей и подлежат соответствующему учету и хранению.

5.8 Защита от вскрытия и проникновения в корпус ПАКМ

Корпус ПАКМ «КриптоПро HSM» оборудован датчиком вскрытия корпуса и датчиком движения (изменения объема).

В случае срабатывания хотя бы одного из датчиков происходит завершение процессов ПАКМ, моментальная очистка оперативной памяти, используемой для открытых контекстов ключей, в журнал заносится информация о данном событии, и ПАКМ завершает работу (переход в состояние Halt).

Дальнейшая загрузка ПАКМ становится невозможной.

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

6 Эксплуатация ПАКМ

Эксплуатация ПАКМ должна производиться в соответствии с инструкцией, содержащейся в документе «ЖТЯИ.00096-03 90 01. Инструкция по использованию».

Порядок установки и включения ПАКМ описывается в разделах 3 и 4 документа «ЖТЯИ.00096-03 90 01. Инструкция по использованию».

Администрирование ПАКМ и удаленное администрирование ПАКМ описываются в разделах 5 и 6 документа «ЖТЯИ.00096-03 90 01. Инструкция по использованию».

Порядок выполнения технического обслуживания и ремонта ПАКМ описывается в разделе 8 документа «ЖТЯИ.00096-03 90 01. 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.00096-03 91 01. Руководство администратора безопасности».

В процессе эксплуатации ПАКМ необходимо выполнять периодический аудит журналов (регистрационных файлов) общесистемного и прикладного программного обеспечения рабочих станций/серверов (в том числе, журналов ПАКМ) с целью сбора информации об удачных и неудачных попытках доступа к объектам, применении привилегий и других событий безопасности и протоколирование этих событий для дальнейшего анализа.

7 Резервирование и восстановление ПАКМ

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

7.1 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 (сертификатов ключей проверки ЭП).

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 специальная роль — Администратор резервного копирования. Чтобы получить доступ к его функциям, в сертификате ключа доступа к ПАКМ привилегированного пользователя должно присутствовать специальное расширение (см. раздел 5 документа «ЖТЯИ.00096-03 91 01. Руководство администратора безопасности»).

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, 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.

7.2 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), за исключением настроек ПАКМ (включая сетевые), 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 шифрованный канал, аналогичный каналу K2, т.е. все данные передаются при помощи TLS-протокола с взаимной аутентификацией сторон при помощи сертификатов открытых ключей обмена ГОСТ Р 34.10-2012.

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 15), IP адрес MASTER сервера.

Все настройки выполняются в соответствии с инструкциями, описанными в документе «ЖТЯИ.00096-03 90 01. 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. 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, данные настройки потребуются 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 потребуется кратковременная остановка работы клиентов ПАКМ для создания на новом MASTER сервере резервной копии, которая будет использоваться для клонирования новых SLAVE серверов.

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 брэндмауэра, 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.

8 Контроль установок времени

В ходе эксплуатации необходимо контролировать установки времени на ПАКМ «КриптоПро HSM». Время, установленное в ПАКМ, не должно отличаться от установленного в Сервере более, чем на 5 минут.

Текущее время ПАКМ «КриптоПро HSM» отображается в окне панели управления «Клиент КриптоПро HSM» (КриптоПро HSM Client) на вкладке «Параметры HSM», на LCD-панели ПАКМ, а также на странице с информацией о ПАКМ в web-интерфейсе администратора. Время предустанавливается предприятием-изготовителем и корректируется администратором ПАКМ (процедура установки системного времени описана в п. 3.5 «ЖТЯИ.00096-03 90 01. Инструкции по использованию»).

При изменении системного времени ПАКМ на значительную величину (более 5 минут) ПАКМ необходимо перезагрузить.

9 Требования по криптографической защите

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

Пользователь несет персональную ответственность за хранение ключевых документов.

Режим простой замены (ECB) допускается использовать только для зашифрования/расшифрования ключевой информации.

Алгоритм открытого распределения ключей должен использоваться совместно с протоколами аутентификации.

Для всех режимов алгоритма шифрования следует обеспечить отсутствие перекрытий ключей.

Не допускается использовать синхропосылки, вырабатываемые не средствами ПАКМ.

Кроме того, рекомендуется:

- на открытые данные вычислять имитовставку в соответствии с ГОСТ 28147-89, ГОСТ Р 34.12-2015 (ГОСТ Р 34.13-2015);
- 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.

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

ПАКМ «КриптоПро HSM» версии 2.0 R3 (собственно ПАКМ, клиентский компонент ПАКМ) применяется в качестве платформы для построения на его основе программных, программно-аппаратных и аппаратных решений в области обеспечения информационной безопасности.

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

Возможны следующие варианты применения ПАКМ:

1) применение клиентского компонента ПАКМ в составе стандартного программного обеспечения Microsoft, ООО «КРИПТО-ПРО» и других компаний, использующих криптографический интерфейс в соответствии с архитектурой Microsoft;

2) встраивание собственно ПАКМ, клиентского компонента ПАКМ во вновь разрабатываемое или существующее прикладное программное обеспечение, программно-аппаратные и аппаратные решения.

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

Встраивание СКЗИ в СФ проводится в соответствии с порядком, определенным Положением ПКЗ-2005 [1], организациями, имеющими соответствующие лицензии [2] на указанные виды деятельности.

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

1) исследования СФ не требуются;

2) требуются исследования по оценке влияния СФ на СКЗИ;

3) требуется проведение тематических исследований СФ со встроенным СКЗИ как самостоятельного шифровального (криптографического) средства⁶.

Исследования по оценке влияния СФ на СКЗИ, а также тематические исследования СФ со встроенным СКЗИ как самостоятельного шифровального (криптографического) средства проводятся в соответствии с порядком, определенным Положением ПКЗ-2005 [1], организациями, имеющими соответствующие лицензии [2] на указанные виды деятельности и необходимую область аккредитации [3].

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

⁶ В соответствии с [2] термины «шифровальное (криптографическое) средство» и «средство криптографической защиты информации» (СКЗИ) являются эквивалентными и включают в себя, в том числе, термин «Средство электронной подписи».

10.1 Применение ПАКМ без исследований СФ

Исследования СФ не требуются при выполнении одного из следующих условий:

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

2) в качестве СФ клиентского компонента ПАКМ используется ПО, которое в соответствии с документацией на СКЗИ «КриптоПро CSP» (из состава клиентского компонента) допускается использовать в качестве СФ этого СКЗИ без исследований с учётом правил эксплуатации СКЗИ и самого ПО⁷;

3) применение прикладным ПО команд хоста Модуля обеспечения защиты платежных систем из состава собственно ПАКМ (подробнее см. ЖТЯИ.00096-03 92 04 и ЖТЯИ.00096-03 97 02);

4) применение консольных команд Модуля обеспечения защиты платежных систем из состава собственно ПАКМ (подробнее см. ЖТЯИ.00096-03 97 01).

10.2 Применение ПАКМ с проведением исследований по оценке влияния

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

- применение СКЗИ и СФ подпадает под случаи, приведенные в п. 3 Положения ПКЗ-2005 [1], или иными нормативными документами определены дополнительные условия применения шифровальных (криптографических) средств;
- в качестве СФ используются программные компоненты из следующего перечня (один пункт или несколько):
 - программные компоненты, использующие приложения (модули) cryptcp, certmgr, stunnel, cpverify, HsmGenArm в режиме «автоматического вызова»⁸;
 - программные компоненты, осуществляющие вызовы (напрямую или опосредовано через различные программные интерфейсы) функций ПАКМ, приведенных в Приложении 1 настоящего документа (подробнее об использовании программных интерфейсов ПАКМ см. разд. 10.4), или функций СКЗИ «КриптоПро CSP», входящего в состав клиентского компонента ПАКМ, приведенных в Приложении 2 соответствующих Правил пользования.

При реализации (построении) криптографических протоколов с использованием указанных выше вызовов функций или приложений (модулей) объем исследований определяется в Техническом задании.

При разработке ИС с использованием функций HsmGenArm необходимо руководствоваться Приложением 2 документа «ЖТЯИ.00096-03 90 02. Использование интерфейсных модулей».

⁷ Ручной вызов приложения HsmGenArm из состава клиентского компонента ПАКМ не допускается. Под режимом «ручного вызова» понимается вызов утилиты из командной строки непосредственно пользователем с визуальной обработкой результатов работы утилиты.

⁸ Под режимом «автоматического вызова» понимается вызов утилиты из компонентов прикладной системы, реализованных в исполняемых модулях, скриптах, bat-файлах и т.п.

10.3 Применение ПАКМ с проведением тематических исследований

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

- применение СКЗИ и СФ подпадает под случаи, приведенные в п. 3 Положения ПКЗ-2005, или иными нормативными документами определены дополнительные условия применения шифровальных (криптографических) средств;
- в качестве СФ используются программные компоненты, осуществляющие вызовы (напрямую или опосредовано через различные программные интерфейсы) функций ПАКМ, не приведенных в Приложении 1 настоящего документа, или функций СКЗИ «КриптоПро CSP», входящего в состав клиентского компонента ПАКМ, не приведенных в Приложении 2 соответствующих Правил пользования.

10.4 Требования при встраивании ПАКМ в прикладные системы

При разработке прикладного ПО платежных систем с использованием команд Модуля обеспечения защиты платежных систем из состава собственно ПАКМ необходимо руководствоваться документами ЖТЯИ.00096-03 92 04 и ЖТЯИ.00096-03 97 02. При использовании консольных команд управления Модуля обеспечения защиты платежных систем из состава собственно ПАКМ необходимо руководствоваться документом ЖТЯИ.00096-03 97 01.

При разработке пользовательских приложений с использованием функций интерфейса CryptoAPI 1.0 и 2.0 или методов классов интерфейса JCA/JCE рекомендуется использовать функции, указанные в Приложении 1 настоящего документа, и функции, приведенные в Приложении 2 Правил пользования на СКЗИ «КриптоПро CSP», входящего в состав клиентского компонента ПАКМ. При этом в части описания параметров функций необходимо руководствоваться документацией Microsoft (для CryptoAPI 1.0 и 2.0), Oracle (для JCA/JCE). В части использования специальных флагов и констант, поддерживаемых СКЗИ «КриптоПро CSP» из состава клиентского компонента ПАКМ, необходимо руководствоваться разделами документов ЖТЯИ.00096-03 92 01. Руководство программиста (CSP_5_0.chm), ЖТЯИ.00096-03 92 02. Руководство программиста JavaCSP, ЖТЯИ.00096-03 92 03. Руководство программиста JavaTLS.

При встраивании клиентского компонента ПАКМ в прикладные системы (прикладное ПО, СФ) должны выполняться следующие требования:

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

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

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

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

3) При вызове функций СКЗИ в обязательном порядке требуется указывать имя и тип провайдера из состава клиентского компонента ПАКМ, использование провайдера «по умолчанию» (NULL) не допускается. Для функций CryptoAPI допускается использование только провайдеров из состава клиентского компонента ПАКМ (типы 75, 80 и 81). Для методов JCA/JCE, в которых могут быть заданы алгоритмы и используемые криптопровайдеры, необходимо использовать отечественные алгоритмы ГОСТ и провайдеры типов JCP, JCSP, Crypto, RevCheck, JTLS.

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

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

6) Корневые сертификаты должны быть получены по доверенному каналу⁹. Размещение в хранилище корневых сертификатов должно проводиться администратором с контролем необходимости наличия конкретных корневых сертификатов, используемых для построения цепочек сертификатов.

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

8) При встраивании клиентского компонента ПАКМ в компоненты СФ, в которых функции создания и/или проверки электронной подписи не являются автоматическими, необходимо проводить оценку соответствия компонентов СФ п.п. 8 и/или 9

⁹ Под доверенным каналом понимается способ передачи, исключающий возможность подмены корневого сертификата (личная передача, криптографически защищенный канал связи и т.п.).

«Требований к средствам электронной подписи», утвержденных Приказом ФСБ России от 27 декабря 2011 г. № 796 «Об утверждении Требований к средствам электронной подписи и Требований к средствам удостоверяющего центра».

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

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

- по исходным кодам и эксплуатационной документации на СФ осуществляется анализ выполнения требований и рекомендаций по встраиванию СКЗИ, изложенных в документации на ПАКМ, а также анализ полноты и корректности эксплуатационной документации на компоненты СФ, включая отражение в указанной документации организационно-технических мер по защите информации с использованием ПАКМ и правил обращения с ключевыми документами;
- проверка корректности реализации в СФ работы с криптографическими ключами;
- проверка корректности реализации механизмов контроля целостности СФ и клиентского компонента ПАКМ;
- проверка корректности работы криптографических функций в штатном и не штатном режимах и в условиях искаженных и/или отсутствующих объектов PKI;
- проверка механизмов визуализации информации для пользователя (п.п. 8 и/или 9 Приложения 1 к Приказу ФСБ России от 27 декабря 2011 г. № 796 «Об утверждении Требований к средствам электронной подписи и Требований к средствам удостоверяющего центра»);
- в случае опосредованного использования функций программного интерфейса CryptoAPI 2.0, методов классов программного интерфейса JCA/JCE или утилиты HsmGenArm из других программных интерфейсов, необходимо проведение исследований прохождения вызовов и передаваемых данных между вызываемым интерфейсом и интерфейсами CryptoAPI 2.0 или JCA/JCE (построение трасс вызовов).

Литература

- [1] Приказ ФСБ России от 09.02.2005 № 66 «Об утверждении Положения о разработке, производстве, реализации и эксплуатации шифровальных (криптографических) средств защиты информации (Положение ПКЗ-2005)». <http://base.garant.ru/187947/>
- [2] Постановление Правительства РФ от 16.04.2012 N 313 «Об утверждении Положения о лицензировании деятельности по разработке, производству, распространению шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств, выполнению работ, оказанию услуг в области шифрования информации, техническому обслуживанию шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств (за исключением случая, если техническое обслуживание шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств, осуществляется для обеспечения собственных нужд юридического лица или индивидуального предпринимателя)». http://www.consultant.ru/document/cons_doc_LAW_128739/
- [3] Перечень организаций, аккредитованных в качестве испытательных лабораторий в Системе сертификации средств криптографической защиты информации. <http://clsz.fsb.ru/clsz/accred.htm>

Приложение 1. Перечень вызовов, использование которых при разработке систем на основе ПАКМ «КриптоПро HSM» всех комплектаций с учетом п.1.5 Формуляра ЖТЯИ.00096-03 30 01 возможно без дополнительных тематических исследований

Интерфейс CryptoAPI

<i>Функция</i>	<i>Описание</i>	<i>Ограничения на использование функции</i>
Функции инициализации и настройки провайдера		
CryptAcquireContext	Функция используется для создания дескриптора криптопровайдера с именем ключевого контейнера, определенным параметром pszContainer.	Полученный дескриптор криптопровайдера должен быть в обязательном порядке освобожден с помощью вызова функции CryptReleaseContext (за исключением вызовов с флагом CRYPT_DELETEKEYSET).
CryptReleaseContext	Функция 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	Функция управляет счетчиком дескрипторов созданного CryptAcquireContext.	
CryptEnumProviders	Функция перечисления установленных криптопровайдеров.	
CryptEnumProviderTypes	Функция перечисления установленных типов криптопровайдеров.	

CryptGetDefaultProvider	Функция получает контекст провайдера, установленного в системе по умолчанию.	
CryptGetProvParam	Функция получает параметры криптопровайдера.	
CryptSetProvParam	Функция 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 генерирует случайные криптографические ключи или ключевую пару (закрытый/открытый ключи).	Полученный дескриптор ключа должен в обязательном порядке быть удален с помощью вызова функции CryptDestroyKey до вызова функции CryptReleaseContext для рабочего дескриптора криптопровайдера.
CryptDestroyKey	Функция удаляет ключ, передаваемый через параметр hKey. После удаления ключ (дескриптор ключа) не может использоваться.	
CryptDeriveKey	Функция 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, CALG_PBKDF2_2012_256,

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

CryptSetKeyParam	Функция устанавливает параметры ключа.	Разрешено использование только со следующими символьными аргументами: KP_CERTIFICATE, KP_CIPHEROID, KP_DHOID, KP_HASHOID, KP_AUDIT_CONTROL.
Функции обработки криптографических сообщений		
CryptSignMessage	Функция создает хэш определенного содержания, подписывает хэш и затем производит закодирование и текста исходного сообщения, и подписанного хэша.	
CryptVerifyMessageSignature	Функция проверяет электронную подпись сообщения.	
CryptVerifyDetachedMessageSignature	Функция проверяет подписанное сообщение, содержащее отсоединенную (detached) подпись или подписи.	
CryptDecodeMessage	Функция декодирует, расшифровывает и проверяет сообщение.	
CryptDecryptAndVerifyMessageSignature	Функция декодирует и проверяет сообщение.	
CryptEncryptMessage	Функция зашифровывает и производит закодирование сообщения. Аутентичность сообщения не обеспечивается.	
CryptDecryptMessage	Функция производит раскодирование и расшифрование сообщения. Проверка аутентичности сообщения не производится. 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 вычисляет максимальное количество байтов, необходимое для закодированного криптографического сообщения, заданного типом сообщения, параметрами кодирования и общей длиной информации, которая должна быть закодирована.	
CryptMsgOpenToEncode	Функция открывает криптографическое сообщение для кодирования и возвращает дескриптор открытого сообщения.	
CryptMsgOpenToDecode	Функция открывает криптографическое сообщение для декодирования и возвращает дескриптор открытого сообщения.	
CryptMsgUpdate	Функция пополняет текст криптографического сообщения.	
CryptMsgGetParam	Функция получает параметр	

	сообщения после того, как криптографическое сообщение было раскодировано или закодировано.	
CryptMsgControl	Функция выполняет контрольное действие.	
CryptMsgClose	Функция закрывает дескриптор криптографического сообщения.	
CryptMsgDuplicate	Функция дублирует дескриптор криптографического сообщения путем увеличения счетчика ссылок.	
Функции работы с алгоритмами хэширования		
CryptCreateHash	Функция инициализирует дескриптор нового объекта функции хэширования потока данных.	Разрешено использование только со следующими символьными аргументами: CALG_GR3411, CALG_GR3411_2012_256, CALG_GR3411_2012_512, CALG_GR3411_HMAC, CALG_GR3411_2012_256_HMAC, CALG_GR3411_2012_512_HMAC, CALG_PBKDF2_94_256, CALG_PBKDF2_2012_256, CALG_PBKDF2_2012_512, CALG_SHAREDKEY_HASH. Полученный дескриптор объекта хэширования должен в обязательном порядке быть удален с помощью вызова функции CryptDestroyHash до вызова функции CryptReleaseContext для рабочего дескриптора криптопровайдера.
CryptDestroyHash	Функция удаляет объект функции хэширования.	
CryptDuplicateHash	Функция создает точную копию	Полученный дескриптор

	объекта функции хэширования, включая все его переменные, определяющие внутреннее состояние объекта функции хэширования.	объекта хэширования должен в обязательном порядке быть удален с помощью вызова функции CryptDestroyHash до вызова функции CryptReleaseContext для рабочего дескриптора криптопровайдера.
CryptGetHashParam	Функция возвращает параметры объекта функции хэширования и значение функции хэширования.	Запрещено использование с символьным аргументом HP_OPAQUEBLOB.
CryptHashData	Функция передает данные указанному объекту функции хэширования.	
CryptSetHashParam	Функция 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 возвращает значение электронной 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 осуществляет проверку электронной подписи.	Разрешено использование только с дескрипторами ключей, полученными ранее с помощью вызова CryptImportPublicKeyInfo (CryptImportPublicKeyInfoEx) из сертификата, проверенного с помощью функции CertVerifyCertificateChainPolicy.

Функции работы с сертификатами, списками аннулированных сертификатов, хранилищем сертификатов

Списки аннулированных сертификатов

CertAddCRLContextToStore	Функция добавляет контекст СОС в хранилище сертификатов.	
CertAddCRLLinkToStore	Функция создает ссылку на список аннулированных сертификатов в другом хранилище.	
CertAddEncodedCRLToStore	Функция создает контекст СОС из закодированного СОС и добавляет его в хранилище сертификатов. Функция создает копию контекста СОС перед добавлением его в хранилище.	
CertEnumCRLsInStore	Функция получает первый или следующий СОС в хранилище. Эта функция используется в цикле для того, чтобы последовательно получить все СОС в хранилище.	
CertFreeCRLContext	Функция освобождает контекст СОС, уменьшая счетчик ссылок на единицу. Когда счетчик ссылок обнуляется, функция освобождает память, выделенную под контекст СОС.	

CertCreateCRLContext	Функция 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 дублирует контекст СОС, увеличивая счетчик ссылок на СОС на единицу.	
CertFindCRLInStore	Функция 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 удаляет определенный контекст СОС из хранилища сертификатов.	
CertFindCertificateInCRL	Функция осуществляет поиск заданного сертификата в списке аннулированных сертификатов.	
CertGetCRLFromStore	Функция 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 CTL

CertGetCRLContextProperty	Функция получает расширенные свойства определенного контекста СОС.	
CertSetCRLContextProperty	Функция устанавливает расширенные свойства определенного контекста СОС.	
CertGetCertificateContextProperty	Функция получает информацию, содержащуюся в расширенных свойствах контекста сертификата.	
CertEnumCertificateContextProperties	Функция позволяет перечислить информацию, содержащуюся в расширенных свойствах контекста сертификата.	
CertSetCertificateContextProperty	Функция устанавливает расширенные свойства для определенного контекста сертификата.	
CertEnumCRLContextProperties	Функция перечисления расширенных свойств списка аннулированных сертификатов.	
CertEnumCTLContextProperties	Функция перечисления расширенных свойств CTL.	
CertGetCTLContextProperty	Функция получает расширенные свойства CTL.	
CertSetCTLContextProperty	Функция устанавливает расширенных свойств CTL.	
CertAddCTLContextToStore	Функция добавляет контекст CTL в хранилище сертификатов.	
CertCreateCTLContext	Функция создает контекст закодированного CTL. Созданный контекст не помещается в хранилище сертификатов. Функция делает копию CTL в созданном контексте.	
CertDuplicateCTLContext	Функция дублирует контекст CTL, увеличивая счетчик ссылок на CTL на единицу.	
CertFreeCTLContext	Функция освобождает контекст CTL, уменьшая счетчик ссылок	

	на единицу. Когда счетчик ссылок обнуляется, функция CertFreeCTLContext освобождает память, выделенную под контекст CTL.	
Сертификаты		
CertAddCertificateContextToStore	Функция добавляет контекст сертификата в хранилище сертификатов.	
CertAddCertificateLinkToStore	Функция добавляет ссылку на сертификат в другом хранилище.	
CertAddEncodedCertificateToStore	Функция создает контекст сертификата из закодированного сертификата и добавляет его в хранилище сертификатов. Созданный контекст не содержит никаких расширенных свойств.	
CertEnumCertificatesInStore	Функция получает первый или следующий сертификат в хранилище сертификатов. Эта функция используется в цикле для того, чтобы последовательно получить все сертификаты в хранилище сертификатов.	
CertFreeCertificateContext	Функция освобождает контекст сертификата, уменьшая счетчик ссылок на единицу.	
CertCreateCertificateContext	Функция создает контекст сертификата из закодированного сертификата. Созданный контекст не помещается в хранилище сертификатов. В созданном контексте функция размещает копию закодированного сертификата.	
CertDuplicateCertificateContext	Функция дублирует контекст сертификата, увеличивая счетчик ссылок на единицу.	
CertFindCertificateInStore	Функция находит первый или	

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

CertAddRefServerOcspResponseContext	Функция увеличения счетчика ссылок на контекст OCSP ответа.	
CertCloseServerOcspResponse	Функция закрытия дескриптора OCSP ответа.	
CertGetServerOcspResponseContext	Функция получения контекста OCSP ответа.	
CertOpenServerOcspResponse	Функция открытия дескриптора OCSP ответа для заданной цепочки сертификатов.	
Оконные функции		
CertSelectCertificate	Функция отображения диалога выбора сертификата по заданным критериям.	
CryptUIDlgCertMgr	Функция отображения диалога управления сертификатами.	
CryptUIDlgSelectCertificate	Функция отображения диалога выбора сертификата.	
CryptUIDlgSelectCertificateFromStore	Функция отображения диалога выбора сертификата из хранилища.	
CryptUIDlgViewCertificate	Функция отображения диалога со свойствами сертификата.	
CryptUIDlgViewContext	Функция отображения сертификата, списка аннулированных сертификатов или CTL.	
CryptUIDlgViewSignerInfo	Функция отображения диалога с информацией о подписавшем.	
CertSelectionGetSerializedBlob	Функция сериализации сертификата из структуры, используемой для отображения.	
GetFriendlyNameOfCert	Функция преобразования имени сертификата к «читаемому» виду.	
Проверка цепочек сертификатов		
CertVerifyCertificateChainPolicy	Функция проверяет цепочку сертификатов на достоверность, включая соответствие критерию	

	истинности.	
CertGetCertificateChain	Функция строит цепочку сертификатов, начиная с последнего сертификата, в обратном направлении до доверенного корневого сертификата, если это возможно.	
CertFreeCertificateChain	Функция освобождает цепочку сертификатов путем уменьшения счетчика ссылок. Если счетчик ссылок равен нулю, то память, выделенная под цепочку, освобождается.	
CertCreateCertificateChainEngine	Функция создает контекст HCERTCHAINENGINE, который позволяет изменять параметры механизма построения цепочки сертификатов. Позволяет ограничивать множество доверенных сертификатов.	
CertFreeCertificateChainEngine	Функция освобождает контекст HCERTCHAINENGINE.	
CertCreateCTLEntryFromCertificateContextProperties	Функция создает CTL на основе свойств атрибутов контекста сертификата.	
CertDuplicateCertificateChain	Функция дублирует контекст 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	Функция установки свойств в контекст сертификата на основе CTL.	
Расширенные свойства сертификата (EKU)		
CertGetEnhancedKeyUsage	Функция получает информацию о расширенном использовании	

	ключа из соответствующего расширения или из расширенных свойств сертификата. Расширенное использование ключа служит признаком правомерного использования сертификата.	
CryptAcquireCertificatePrivateKey	Функция получает дескриптор HCryptProv и параметр dwKeySpec для определенного контекста сертификата.	
Функции работы с идентификаторами		
CryptFindOIDInfo	Функция получает первую предопределенную или зарегистрированную структуру CRYPT_OID_INFO, согласованную с определенным типом ключа и с ключом.	
CryptEnumOIDInfo	Функция перечисления зарегистрированных идентификаторов и получение информации для них.	
Функции работы с хранилищем		
CertOpenStore	Функция открывает хранилище сертификатов, используя заданный тип провайдера.	
CertDuplicateStore	Функция дублирует дескриптор хранилища, увеличивая счетчик ссылок на хранилища на единицу.	
CertOpenSystemStore	Функция используется для открытия наиболее часто используемых хранилищ сертификатов.	
CertCloseStore	Функция закрывает дескриптор хранилища сертификатов и уменьшает счетчик ссылок на хранилища на единицу.	
CertAddStoreToCollection	Функция добавляет хранилище в коллекцию.	
CertControlStore	Функция устанавливает нотификации при различиях в	

	заэкшированным хранилище и физическом хранилище.	
Функции, используемые для работы с открытыми данными и объектами		
CryptImportPublicKeyInfoEx2	Функция импортирует информацию об открытом ключе и возвращает дескриптор открытого ключа.	
CryptImportPublicKeyInfoEx	Функция импортирует информацию об открытом ключе в CSP и возвращает дескриптор открытого ключа.	
CryptImportPublicKeyInfo	Функция преобразовывает и импортирует информацию об открытом ключе в провайдер и возвращает дескриптор открытого ключа.	
CryptExportPublicKeyInfoEx	Функция экспортирует информацию об открытом ключе, связанную с соответствующим секретным ключом провайдера.	
CryptExportPublicKeyInfo	Функция экспортирует информацию об открытом ключе, ассоциированную с соответствующим секретным ключом провайдера.	
CertCompareCertificate	Функция сравнивает два сертификата для того, чтобы определить, являются ли они идентичными.	
CertCompareIntegerBlob	Функция сравнивает два целочисленных блока для определения того, представляют ли они собой два равных числа.	
CryptExportPublicKeyInfoFrom BCryptKeyHandle	Функция экспортирует информацию об открытом ключе, ассоциированную с соответствующим секретным ключом провайдера.	
Функции кодирования/декодирования		
CryptDecodeObject	Функция используется для декодирования сертификатов,	

	списков аннулированных сертификатов (СОС) и запросов на сертификаты.	
CryptDecodeObjectEx	Функция используется для декодирования сертификатов, списков аннулированных сертификатов и запросов на сертификаты.	
CryptEncodeObject	Функция используется для кодирования сертификатов, списков аннулированных сертификатов и запросов на сертификаты.	
CryptEncodeObjectEx	Функция используется для кодирования сертификатов, списков аннулированных сертификатов и запросов на сертификаты.	
Получение объектов из удаленных источников		
CryptRetrieveObjectByUrlA	Функция получает объект инфраструктуры открытых ключей по заданному URL.	
CryptRetrieveObjectByUrlW	Функция является unicode версией функции CryptRetrieveObjectByUrlA.	
Дополнительные функции		
CryptBinaryToString	Функция переводит двоичную строку в строку Base64/HEX.	
CryptStringToBinary	Функция переводит строку Base64/HEX в двоичную строку.	
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).	

Интерфейс JCA/JCE (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() создает объект, создающий объекты закрытых и открытых ключей ЭП и VKO на основе ключевых спецификаций, представляющих собой внутреннее представление ключевой информации.	
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 KeyPairGenerator	Метод generateKeyPair() класса 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 создает 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 KeyStore	Метод осуществляет запись 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).	Запрещено использовать с параметром 1.2.643.2.2.30.0.
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 PrivateKeyUsageExtension	Метод, проверяющий срок 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 РКХ.
setParameter() класса Signature	Метод устанавливает используемую хэш-функцию и ее 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 X509Certificate	Метод возвращает ключ проверки ЭП из сертификата.	
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.	
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 имен эмитента сертификата.	
getSubjectAlternativeNames() класса X509Certificate	Метод возвращает список альтернативных имен 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, 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.	
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 сертификат.	
printToDER() класса GostCertificateRequest	Метод осуществляет печать подписанного запроса в DER-кодировке в передаваемый PrintStream.	
getEncodedCertFromBASE64() класса GostCertificateRequest	Метод отправляет запрос в кодировке BASE64 центру сертификации и получает соответствующий запросу сертификат.	
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 устанавливает алгоритм шифрования.	
clone() класса Mac	Метод производит копирование объекта имитозащиты.	
update() класса Mac	Метод осуществляет обработку данных в процессе вычисления имитовставки.	
doFinal() класса Mac	Метод завершает вычисление значения имитовставки.	
reset() класса Mac	Метод осуществляет 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 потока подписываемых данных.	
update() класса CAdESSignature	Метод осуществляет обработку порции данных.	
close() класса CAdESSignature	Метод осуществляет завершение обработки подписываемых данных.	
replaceSigners() класса CAdESSignature	Метод осуществляет изменения данных подписывающих в упакованном подписанном сообщении.	
addRecipient() класса EnvelopedSignature	Метод добавляет 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 PKIX 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 PKIX и проверки области действия сертификата.
open() класса EnvelopedSignature	Метод осуществляет открытие потока зашифрования сообщения.	
update() класса EnvelopedSignature	Метод осуществляет шифрование порции данных.	
close() класса EnvelopedSignature	Метод осуществляет шифрование финальной порции данных и закрывающий поток.	
getRecipients() класса EnvelopedSignature	Метод возвращает список получателей сообщения.	
decrypt() класса EnvelopedSignature	Метод позволяет получателю расшифровать сообщение.	

getCAdESSignatureType() класса CAdESType	Метод возвращает тип CAdES сообщения.	
getSignerInfo() класса CAdESSigner	Метод возвращает информацию о подписанте.	
getSignerSignedAttributes() класса CAdESSigner	Метод возвращает список подписываемых атрибутов сообщения.	
getSignerUnsignedAttributes() класса CAdESSigner	Метод возвращает список неподписываемых атрибутов сообщения.	
getSignatureTimestampToken()) класса CAdESSigner	Метод возвращает внутренний штамп (signature-timestamp) времени сообщения.	
getCAdESCTimestampToken() класса CAdESSigner	Метод возвращает внешний штамп (CAdES-C-timestamp) времени сообщения.	
getCAdESCertificates() класса CAdESSigner	Метод возвращает список сертификатов (certificate-values).	
getSignerCertificate() класса CAdESSigner	Метод возвращает сертификат подписавшего сообщение.	
getSignatureType() класса CAdESSigner	Метод возвращает тип подписи.	
getCAdESCountersignerInfos() класса CAdESSigner	Метод возвращает список заверителей сообщения.	
setCertificateStore() класса CAdESSigner	Метод устанавливает хранилище сертификатов, из которого позже может быть извлечен сертификат подписи.	
enhance() класса CAdESSigner	Метод осуществляет «усовершенствование» подписи CAdES-BES до CAdES-X Long Type 1.	
addCountersigner() класса CAdESSigner	Метод осуществляет добавление заверяющей подписи к отдельному подписанту.	
verify() класса CAdESSigner	Метод осуществляет проверку одной отдельной подписи CAdES	
build() класса CertPathBuilder	Метод осуществляет построение цепочки сертификатов.	

getAlgorithm() класса CertPathValidator	Метод возвращает имя алгоритма, осуществляющего проверку цепочек сертификатов.	
getDefaultType() класса CertPathValidator	Метод возвращает имя алгоритма, осуществляющего проверку цепочек сертификатов, принятого по умолчанию.	
getInstance() класса CertPathValidator	Метод возвращает объект класса CertPathValidator.	
getProvider() класса CertPathValidator	Метод осуществляет получение объекта провайдера алгоритма, осуществляющего проверку цепочек сертификатов, принятого по умолчанию.	
validate() класса CertPathValidator	Метод осуществляет проверку цепочек сертификатов.	
addCertPathChecker() класса CertPathParameters	Метод устанавливает дополнительные правила проверки сертификатов.	
addCertStore() класса CertPathParameters	Метод устанавливает дополнительные хранилища сертификатов и CRL.	
getCertPathCheckers() класса CertPathParameters	Метод возвращает список установленных дополнительных правил проверки сертификатов.	
getCertStores() класса CertPathParameters	Метод возвращает список хранилищ сертификатов и CRL.	
getDate() класса CertPathParameters	Метод возвращает дату, на которую проверяется верность цепочки сертификатов.	
getInitialPolicies() класса CertPathParameters	Метод возвращает список OID'ов политик, которые должны быть указаны в сертификатах, из которых строится цепочка.	
getPolicyQualifiersRejected() класса CertPathParameters	Метод возвращает флаг PolicyQualifiersRejected.	
getSigProvider() класса CertPathParameters	Метод получает имя провайдера ЭП.	
getTargetCertConstraints()	Метод возвращает ограничения	

класса CertPathParameters	на целевой сертификат.	
getTrustAnchors() класса CertPathParameters	Метод получает список доверенных сертификатов.	
isAnyPolicyInhibited() класса CertPathParameters	Метод возвращает признак обработки всех политик, указанных в сертификате.	
isExplicitPolicyRequired() класса CertPathParameters	Метод возвращает признак наличия явных описаний политик.	
isPolicyMappingInhibited() класса CertPathParameters	Метод возвращает признак подавления PolicyMapping.	
isRevocationEnabled() класса CertPathParameters	Метод устанавливает признак необходимости проверки сертификатов с помощью OCSP/CRL.	
setAnyPolicyInhibited() класса CertPathParameters	Метод устанавливает признак обработки всех политик, указанных в сертификате.	
setCertPathCheckers() класса CertPathParameters	Метод устанавливает набор дополнительных правил проверки сертификатов.	
setCertStores() класса CertPathParameters	Метод устанавливает набор дополнительных хранилищ сертификатов и CRL.	
setDate() класса CertPathParameters	Метод устанавливает дату, на которую должна производиться проверка цепочки сертификатов.	
setExplicitPolicyRequired() класса CertPathParameters	Метод устанавливает признак наличия явных описаний политик.	
setInitialPolicies() класса CertPathParameters	Метод устанавливает список OID'ов политик, которые должны быть указаны в сертификатах, из которых строится цепочка.	
setPolicyMappingInhibited() класса CertPathParameters	Метод устанавливает признак подавления PolicyMapping.	
setPolicyQualifiersRejected() класса CertPathParameters	Метод устанавливает флаг PolicyQualifiersRejected.	

setRevocationEnabled() класса CertPathParameters	Метод устанавливает признак необходимости проверки сертификатов с помощью OCSP/CRL.	
setTargetCertConstraints() класса CertPathParameters	Метод устанавливает ограничения на целевой сертификат.	
setTrustAnchors() класса CertPathParameters	Метод устанавливает список доверенных сертификатов.	
getMaxPathLength() класса PKIXBuilderParameters	Метод получает максимальную возможную длину строимой цепочки.	
setMaxPathLength() класса PKIXBuilderParameters	Метод устанавливает максимальную возможную длину строимой цепочки.	
getXAdESSignerInfos() класса XAdESSignature	Метод получает список всех подписантов.	
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 возвращает типа XAdES-подписи.	
getEarliestValidSignatureTimeStampToken() класса XAdESSignerT	Метод возвращает самый ранний валидный внутренний штамп времени.	
verify() класса XAdESSigner	Метод осуществляет проверку одной отдельной подписи XAdES.	
getSignerInfo() класса XAdESSigner	Метод возвращает узел подписи в документе.	

getSignerCertificate() класса XAdESSigner	Метод возвращает сертификат ключа ЭП данного подписанта.	
getElement() класса XAdESSigner	Метод возвращает подписываемый узел.	
getDocument() класса XAdESSigner	Метод возвращает подписываемый документ.	
getPrivateKey() класса JCPPPrivateKeyEntry	Метод возвращает дескриптор ключа privKey.	
isExportable() класса JCPPPrivateKeyEntry	Метод возвращает значение флага экспортируемости закрытого ключа/ключа ЭП.	
getCertificateChain() класса JCPPPrivateKeyEntry	Метод возвращает цепочку сертификатов.	
getCertificate() класса JCPPPrivateKeyEntry	Метод возвращает клиентский сертификат (первый в цепочке сертификатов).	
toString() класса JCPPPrivateKeyEntry	Метод возвращает строковое представление цепочки сертификатов.	
isSilentMode() класса JCPPProtectionParameter	Метод возвращает режим открытия ключевого контейнера.	
isAllowEmptyChain() класса JCPPProtectionParameter	Метод возвращает разрешение использовать null вместо цепочки сертификатов.	
getKeyType() класса JCPPProtectionParameter	Метод возвращает тип ключа.	
sign() класса XMLSignature	Метод создает ЭП сообщения.	Разрешено использование только со следующими URI/URN: • http://www.w3.org/2001/04/xmldsig-more#gostr34102001-gostr3411 • urn:ietf:params:xml:ns:cp-xmlsec:algorithms:gostr34102001-gostr3411 • urn:ietf:params:xml:ns:cp-xmlsec:algorithms:gostr34102001-gostr3411

		102012-gostr34112012-256 • urn:ietf:params:xml:ns:cp xmlsec: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.