СТАНДАРТ ПЛАТФОРМЫ ЦИФРОВОГО РУБЛЯ

УТВЕРЖДАЮ

Директор Департамента
информационной безопасности
Банка России

В.А. Уваров

СПЕЦИФИКАЦИЯ НА ПРОГРАММНЫЙ МОДУЛЬ

Оглавление

1 Лист регистрации изменений 4

2 Введение 5

2.1 Общие положения 5

2.1.1 Полное наименование системы и ее условное обозначение 5

2.1.2 Назначение и цели создания ПМ 5

2.1.3 Перечень нормативно-технических документов и методических материалов, использованных при разработке Спецификации 7

2.1.4 Определения, обозначения и сокращения 12

2.2 Структурная схема решения 16

2.3 Обзор функций ПМ 18

2.4 Требования к среде функционирования ПМ 25

2.5 Требования к интерфейсам ПМ 26

2.5.1 Пользовательские интерфейсы 26

2.5.2 Программные интерфейсы 26

3 Функциональные требования 28

3.1 Функциональная схема 28

3.1.1 Правила формирования названия потока 28

3.1.2 Список компонентов 29

3.1.3 Функции по информационным потокам 30

3.2 Описание классов и методов ПМ 32

3.2.1 Класс Core 33

3.2.2 Класс Rng 41

3.2.3 Класс Operations 41

3.2.4 Класс GostTlsSocket 61

3.2.5 Сервисные функции 67

3.2.6 Список ошибок 68

3.3 Порядок использования ПМ 71

3.3.1 Инициализация ПМ 71

3.3.2 Работа с криптографическими функциями 72

3.3.3 Загрузка САС 73

3.3.4 Смена пароля для доступа к хранилищу 73

3.3.5 Смена криптографических ключей пользователя 74

3.3.6 Сбор ЦО 74

3.3.7 Установка ГОСТ TLS соединения 76

3.4 Требования к логическому хранению данных 77

3.4.1 Требования к составу, структуре и способам организации данных в системе 77

3.4.2 Схема хранения ключевой информации 79

3.4.3 Требования к контролю, хранению, обновлению и восстановлению данных 90

3.4.4 Требования к программному обеспечению 90

3.4.5 Требования к форматам ЭП 93

3.4.6 Требования к встраиванию СКЗИ 93

4 Нефункциональные требования 95

4.1 Требования к производительности ПМ 95

4.2 Содержание, объем и организация работ по созданию ПМ 95

4.2.1 Общие требования 95

4.2.2 Требования к разработчику ПМ 96
4.2.3 Работы по подготовке и проведению оценки влияния ПМ на выполнение требований, предъявленных к СКЗИ.....96
4.2.4 Работы по внесению ПМ в Реестр.....98
4.3 Требования к программной документации 99
4.3.1 Общие требования к составу технической документации.....99
4.3.2 Требования к документации, предоставляемой для прохождения оценки влияния.....99
4.3.3 Требования к документации, предоставляемой для включения ПМ в Реестр 101
4.3.4 Требования к порядку актуализации документации ПМ.....103
4.4 Порядок контроля и приемки ПМ.....104
4.4.1 Виды, состав, объем и методы испытаний ПМ.....104
4.4.2 Общие требования к приемке работ.....105
4.5 Требования к порядку встраивания ПМ в МП.....106
4.5.1 Требования к организации работ.....106
4.5.2 Требования к разработчику МП 107
4.5.3 Требования к совместимости с МП.....107
4.5.4 Требования к использованию функций ПМ.....108
4.5.5 Требования к механизмам доставки транспортных сертификатов и сертификатов проверки ЭП.....108
4.6 Поддержка жизненного цикла ПМ.....109
4.7 Характеристики пользователей и персонала.....109
4.7.1 Требования к пользователям ПМ 109
4.7.2 Требования к персоналу, обеспечивающему техническую поддержку и модернизацию ПМ 110

1 Лист регистрации изменений

Дата	Автор	Описание

2 Введение

Документ содержит требования и рекомендации по разработке программного модуля, предназначенного для встраивания в мобильные приложения для организации дистанционных каналов обслуживания клиентов финансовых организаций. Программный модуль обеспечивает хранение криптографических ключей пользователя мобильного приложения и выполнение криптографических преобразований для совершения операций на Платформе Цифрового Рубля (далее — ПлЦР), а также (при необходимости) других финансовых операций.

Данный документ является спецификацией, включающей требования и рекомендации к интерфейсам и функциональности программного модуля, используемого в составе СКЗИ, его среде функционирования, а также к процессам разработки и поддержки жизненного цикла. Документ предназначен для использования в качестве руководства при проектировании и разработке программного модуля. Документ не ограничивает функциональность программного модуля, при необходимости и на усмотрение разработчика перечень функций программного модуля может быть расширен.

2.1 Общие положения

2.1.1 Полное наименование системы и ее условное обозначение

Программный модуль для мобильных финансовых приложений (далее — ПМ).

2.1.2 Назначение и цели создания ПМ

ПМ предназначен для встраивания в мобильные приложения (далее МП) финансовых организаций и должен обеспечивать взаимодействие Клиента с ПлЦР и, при необходимости, с другими информационными системами, выполняя криптографические преобразования.

ПМ должен предоставлять прикладной программный интерфейс для организации взаимодействия с МП и его компонентами.

В рамках настоящего документа рассматривается разработка ПМ для использования в операционных системах из списка:

- Google Android,
- Apple iOS,
- Аврора (Aurora),
- HarmonyOS.

ПМ предназначен для выполнения следующих криптографических операций:

- формирование и хранение криптографических ключей;
- формирование запроса на выпуск сертификата, организация хранения сертификатов и списка отозванных сертификатов;
- подписание электронных сообщений (далее — ЭС) с использованием криптографических алгоритмов в соответствии с ГОСТ Р 34.10-2012/34.10-2018;
- зашифрование ЭС с использованием криптографических алгоритмов в соответствии с ГОСТ Р 34.12-2015/34.12-2018/ ГОСТ Р 34.13-2015/34.13-2018;
- проверка подписи ЭС с использованием криптографических алгоритмов в соответствии с ГОСТ Р 34.10-2012/34.10-2018;
- расшифрование ЭС с использованием криптографических алгоритмов в соответствии с ГОСТ Р 34.12-2015/34.12-2018/ ГОСТ Р 34.13-2015/34.13-2018;
- организация односторонне и двусторонне аутентифицированного защищенного канала между МП и его сервисами, обеспечивающего целостность и конфиденциальность передаваемых данных, по протоколу TLS с использованием криптографических алгоритмов в соответствии с Р 1323565.1.020-2020/ Р 1323565.1.030-2020.

Полный список функций, которые реализует ПМ, приведен в разделе 2.3.

Цель создания ПМ — предоставить российское программное обеспечение для выполнения криптографических операций в МП для различных операционных систем.

Криптографические операции в ПМ должны выполняться посредством вызова в ПМ функций сертифицированного средства криптографической защиты информации (далее — СКЗИ). СКЗИ должен включаться в состав ПМ или ПМ должен сам являться сертифицированным СКЗИ.

Корректность выполнения криптографических преобразований должна обеспечиваться использованием сертифицированного СКЗИ и, если это необходимо, подтверждаться оценкой влияния в соответствии с документом [9].

ПМ, разработанный по данной спецификации, может быть внесен в Единый реестр российских программ для электронных вычислительных машин и баз данных (далее — Реестр).

МП, содержащее в своем составе ПМ, при необходимости может пройти оценку соответствия требованиям к оценочному уровню доверия (далее — ОУД) не ниже, чем ОУД 4, согласно требованиям национального стандарта Российской Федерации ГОСТ Р ИСО/15408-3-2013, или иметь возможность пройти сертификацию в системе сертификации Федеральной службы по техническому и экспортному контролю.

2.1.3 Перечень нормативно-технических документов и методических материалов, использованных при разработке Спецификации

Спецификация разработана с использованием следующих документов:

[1] ГОСТ 19.201.78, Единая система программной документации. Техническое задание. Требования к содержанию и оформлению.

[2] ГОСТ Р 34.10-2012, Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи.

[3] ГОСТ 34.12-2018, Информационная технология. Криптографическая защита информации. Блочные шифры.

[4] Рекомендации по стандартизации, Р 1323565.1.030-2020, Информационная технология. Криптографическая защита информации. Использование

российских криптографических алгоритмов в протоколе безопасности транспортного уровня (TLS 1.3).

[5] Рекомендации по стандартизации, Р 1323565.1.020-2020, Информационная технология. Криптографическая защита информации. Использование российских криптографических алгоритмов в протоколе безопасности транспортного уровня (TLS 1.2).

[6] ГОСТ Р ИСО/МЭК 15408-1-2012, Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 1: Введение и общая модель.

[7] ГОСТ Р ИСО/МЭК 15408-2-2013, Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 2: Функциональные компоненты безопасности.

[8] ГОСТ Р ИСО/МЭК 15408-3-2013, Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 3: Компоненты доверия к безопасности.

[9] Приказ ФСБ России от 9 февраля 2005 г. № 66 «Положение о разработке, производстве, реализации и эксплуатации шифровальных (криптографических) средств защиты информации (Положение ПКЗ-2005)».

[10] Банк России «Платформа Цифрового рубля. Правила заполнения полей сертификатов»[1].

[11] Банк России. Альбом электронных сообщений, используемых для взаимодействия Участников Платформы Цифрового рубля в актуальной редакции. — URL: https://www.cbr.ru/fintech/dr/doc_dr/albums_r/

[1] При необходимости использования данного документа необходимо обратиться в Банк России путем формирования официального запроса.

[12] Стандарт Банка России СТО БР БФБО-1.7-2023 «Безопасность финансовых (банковских) операций. Обеспечение безопасности финансовых сервисов с использованием технологии цифровых отпечатков».

[13] Постановление Правительства РФ от 16 ноября 2015 г. № 1236 «Об установлении запрета на допуск программного обеспечения, происходящего из иностранных государств, для целей осуществления закупок для обеспечения государственных и муниципальных нужд».

[14] Постановление Правительства РФ от 23 марта 2017 г. № 325 «Об утверждении дополнительных требований к программам для электронных вычислительных машин и базам данных, сведения о которых включены в реестр российского программного обеспечения, и внесении изменений в Правила формирования и ведения единого реестра российских программ для электронных вычислительных машин и баз данных».

[15] Приказ Минкомсвязи России № 62 «Об утверждении административного регламента предоставления Министерством цифрового развития, связи и массовых коммуникаций Российской Федерации государственной услуги по формированию и ведению единого реестра российских программ для электронных вычислительных машин и баз данных и единого реестра евразийских программ для электронных вычислительных машин и баз данных».

[16] Министерство цифрового развития, связи и массовых коммуникаций РФ «Методические рекомендации по работе с Федеральной государственной информационной системой «Реестры программ для электронных вычислительных машин и баз данных» (ФГИС Реестры ПО)».

[17] ГОСТ Р 56939-2016, Защита информации. Разработка безопасного программного обеспечения. Общие требования.

[18] Постановление Правительства РФ от 16 апреля 2012 г. № 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)».

[19] ГОСТ Р 59792-2021, Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем.

[20] ГОСТ 34.10-2018, Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи.

[21] ГОСТ Р 34.13-2015, Информационная технология (ИТ). Криптографическая защита информации. Режимы работы блочных шифров.

[22] ГОСТ 34.13-2018, Информационная технология (ИТ). Криптографическая защита информации. Режимы работы блочных шифров.

[23] ГОСТ Р 34.12-2015, Информационная технология. Криптографическая защита информации. Блочные шифры.

[24] Р 1323565.1.012-2017, «Информационная технология. Криптографическая защита информации. Принципы разработки и модернизации шифровальных (криптографических) средств защиты информации.

[25] Положение Банка России от 17.04.2019 № 683-П «Об установлении обязательных для кредитных организаций требований к обеспечению защиты информации при осуществлении банковской деятельности в целях противодействия осуществлению переводов денежных средств без согласия клиента».

[26] Банк России «Методический документ «Профиль защиты прикладного программного обеспечения автоматизированных систем и приложений кредитных организаций и некредитных финансовых организаций»» — URL: https://www.cbr.ru/content/document/file/132666/inf_note_feb_0422.pdf.

[27] ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств».

[28] Стандарт платформы цифрового рубля «Требования операционно-технологического взаимодействия на платформе цифрового рубля» — URL: https://www.cbr.ru/fintech/dr/doc_dr/standarts.

[29] Положение Банка России от 07.12.2023 № 833-П «О требованиях к обеспечению защиты информации для участников платформы цифрового рубля».

[30] ГОСТ Р 34.11-2012, «Информационная технология. Криптографическая защита информации. Функция хэширования».

[31] Банк России 25 декабря 2023г. «Временные требования по обеспечению информационной безопасности для автоматизации выпуска сертификатов пользователя платформы цифрового рубля»[2].

[32] Федеральный закон от 06.04.2011 №63-ФЗ «Об электронной подписи».

[33] Стандарт платформы цифрового рубля «Требования и рекомендации к пользовательским интерфейсам при совершении операций с цифровым рублем» — URL: https://www.cbr.ru/fintech/dr/doc_dr/standarts/.

[34] Стандарт платформы цифрового рубля «Порядок проведения работ по оценке влияния аппаратных, программно-аппаратных и программных средств сети (системы) конфиденциальной связи, совместно с которыми предполагается штатное функционирование программного модуля Банка России, на выполнение

[2] При необходимости использования данного документа необходимо обратиться в Банк России путем формирования официального запроса.

предъявленных к входящему в его состав средству криптографической защиты информации требований» — URL: https://www.cbr.ru/fintech/dr/doc_dr/standarts/.

[35] Приказ ФСБ России от 27 декабря 2011 г. № 796 «Об утверждении Требований к средствам электронной подписи и Требований к средствам удостоверяющего центра».

2.1.4 **Определения, обозначения и сокращения**

В спецификации используется список обозначений и сокращений, используемый в документах [11]. Список сокращений приведен в Таблице 1.

Таблица 1 — Принятые сокращения

Сокращение	**Расшифровка**
ППИ	Прикладной программный интерфейс
HTTP	Hypertext Transfer Protocol
SDK	Software development kit
TLS	Transport layer security — протокол защиты транспортного уровня
CMS	Cryptographic Message Syntax — синтаксис криптографических сообщений
БДСЧ	Биологический датчик случайных чисел
ГОСТ TLS	Протокол TLS, адаптированный для использования с российскими криптографическими стандартами ГОСТ [4], [5], обеспечивающий безопасность передачи данных с использованием национальных алгоритмов шифрования, хэширования и цифровой подписи.
ЕБС	Единая биометрическая система
ЕСИА	Единая система идентификации и аутентификации
ИС	Информационная система
КК	Контур контроля
КО	Контур обработки
МП	Мобильное приложение
ОП	Оператор Платформы Цифрового рубля
ОС	Операционная система
ПДСЧ	Программный датчик случайных чисел

Сокращение	Расшифровка
ПО	Программное обеспечение
ПлЦР	Платформа Цифрового рубля
ПМ	Программный модуль для мобильных финансовых приложений
ПУЦ	Подчиненный удостоверяющий центр
РОРД	Регистрационный, операционный и расчетный депозитарий
CAC, CRL	Список отозванных сертификатов
СКЗИ	Средство криптографической защиты информации
СКПЭП	Сертификат ключа проверки электронный подписи
СФ	Среда функционирования
ТЗ	Техническое задание на проведение исследований по оценке влияния специального программного обеспечения ПМ
УЦ	Удостоверяющий центр
УНЭП	Усиленная неквалифицированная электронная подпись
УЦ БР	Корневой Удостоверяющий центр ПлЦР Банка России
УЦ Безопасности	Удостоверяющий центр на стороне участница ПлЦР, используемый для выдачи TLS-сертификатов
ПУЦ УНЭП	Подчиненный Удостоверяющий центр на стороне участника ПлЦР, являющийся подчиненным к корневому УЦ БР, используемый для выдачи сертификатов УНЭП пользователям ПлЦР
ХПМ	Хранилище ПМ
ХМП	Хранилище МП
ХСКЗИ	Хранилище СКЗИ
ЦО	Цифровой отпечаток устройства
ЦР	Цифровой рубль
ЭП	Электронная подпись
ЭС	Электронное сообщение

Список терминов и их определений, используемых в документе, приведен в Таблице 2.

Таблица 2 — Определения

Термин	Определение
Биологический датчик случайных чисел	Датчик, который формирует случайную последовательность чисел на основе случайных испытаний. Испытания основаны на случайном характере многократного взаимодействия человека с СКЗИ и средой функционирования СКЗИ.
Криптографический провайдер	ПО, предоставляющее реализацию криптографических алгоритмов и функций для обеспечения безопасности данных и операций с ними.
Мобильное приложение	ПО для мобильного устройства, которое используется клиентами финансовой организации для дистанционного доступа к финансовым услугам и сервисам, в том числе для осуществления операций с ЦР.
Пароль	Криптографический ключ, принимающий значения из множества малой мощности. Как правило, представляется в виде конечной последовательности символов из фиксированного алфавита и используется для аутентификации субъекта доступа.
Пользователь ПлЦР	Пользователи Платформы ЦР — физические и юридические лица, индивидуальные предприниматели, а также физические лица, применяющие специальный налоговый режим «Налог на профессиональный доход».
Платформа ЦР	Информационная система, посредством которой взаимодействуют Оператор, Участники ПлЦР и Пользователи ПлЦР в соответствии с правилами ПлЦР.
Программный модуль	Программное обеспечение, имеющее в своем составе сертифицированное СКЗИ для осуществления криптографических преобразований, встраиваемое в мобильное финансовое приложение.
Программный датчик случайных чисел	Датчик, вырабатывающий псевдослучайную последовательность путем детерминированного преобразования инициализирующей последовательности (исходной ключевой информации).
Сбор энтропии	Механизм выработки последовательности случайных чисел для генерации пары криптографических ключей с использованием биометрического датчика случайных чисел (БДСЧ).
Участник ПлЦР	Клиент Банка России (кредитная организация, некредитная финансовая организация), предоставляющий доступ к сервисам Платформы ЦР

Термин	Определение
	своим Клиентам, либо использующий сервисы Платформы ЦР для выполнения операций с ЦР.
Хранилище СКЗИ (Защищенное хранилище)	Область памяти для хранения криптографических ключей, доступ к которой защищен паролем и имеет защиту от подбора пароля.
Хранилище ПМ (Внутреннее хранилище)	Область памяти, которая выделяется для обеспечения работы ПМ.
Хранилище МП	Область памяти, которая выделяется мобильной ОС для обеспечения работы МП.
Хранилище сертификатов СКЗИ	Защищенная область памяти, предназначенная для хранения сертификатов, которые используются для криптографических операций.
Цифровой отпечаток устройства (device fingerprint)	Уникальный идентификатор, который формируется в виде производного значения из значений параметров устройства, и позволяет идентифицировать устройство пользователя при получении им финансовых услуг.
Электронное сообщение	Совокупность информации, имеющая криптографическую защиту и передаваемая между субъектами взаимодействия при выполнении бизнес-процессов.
CMScert	Подпись типа CMS (Cryptographic Message Syntax) содержит сертификат открытого ключа, соответствующего закрытому ключу, на котором произведено формирование ЭП.
CMSid	Подпись типа CMS (Cryptographic Message Syntax) содержит идентификатор открытого ключа, соответствующего закрытому ключу, на котором произведено формирование ЭП.
RAW	ЭП, представленная в виде массива байт без сертификата или идентификатора ключа
Big-endian	Формат хранения и передачи последовательности байтов в порядке от старшего к младшему.
Little-endian	Формат хранения и передачи последовательности байтов в порядке от младшего к старшему.

2.2 Структурная схема решения

Раздел содержит высокоуровневое описание структуры решения, включающее основные компоненты и взаимосвязи между ними. Схема решения приведена на Рисунке 1.

На схеме Мобильное устройство включает в себя несколько ключевых компонентов. Мобильное финансовое приложение (МП) представляет собой программное обеспечение, используемое для дистанционного доступа к финансовым услугам и сервисам, включая операции с цифровыми рублями (ЦР). Ядро ОС мобильного устройства управляет аппаратными ресурсами для функционирования мобильного приложения.

В МП на схеме выделены следующие архитектурные слои:

- Графический пользовательский интерфейс;
- Доменный уровень, в котором реализована основная функциональность приложения, а именно: бизнес-логика, обработка данных и управление взаимодействием между различными частями приложения;
- Слой данных, который обеспечивает взаимодействие доменного-уровня и хранилища данных, гарантируя целостность и безопасность при обработке и хранении информации. В этом слое также расположен ПМ, отвечающий за предоставление доступа к криптографическим функциям и обеспечение дополнительной защиты данных.

ПМ реализует интерфейсы для выполнения криптографических операций и для установления защищенного канала связи с внешними сервисами, обеспечивая таким образом защиту данных при передаче. Доменный уровень МП при необходимости вызывает эти интерфейсы, а ПМ использует интерфейсы встроенного в него СКЗИ для обеспечения защиты информации.

При передаче данных во внешние системы Доменный уровень МП обращается к интерфейсам ГОСТ TLS ПМ для установления защищенного канала связи с внешними сервисами в сети интернет.

Во время установки МП создается хранилище МП, которое содержит данные для приложения и его компонентов, в том числе ПМ и СКЗИ. МП не может получить прямой доступ к системным файлам и каталогам, а также файлам других приложений.

Хранилище ПМ используется для хранения служебных данных, необходимых для работы ПМ. Хранилище СКЗИ предназначено для хранения криптографических ключей и других конфиденциальных данных в защищенном виде. Управление этими хранилищами осуществляется через ядро ОС, которое отвечает за сохранность и безопасность всех данных мобильного устройства.

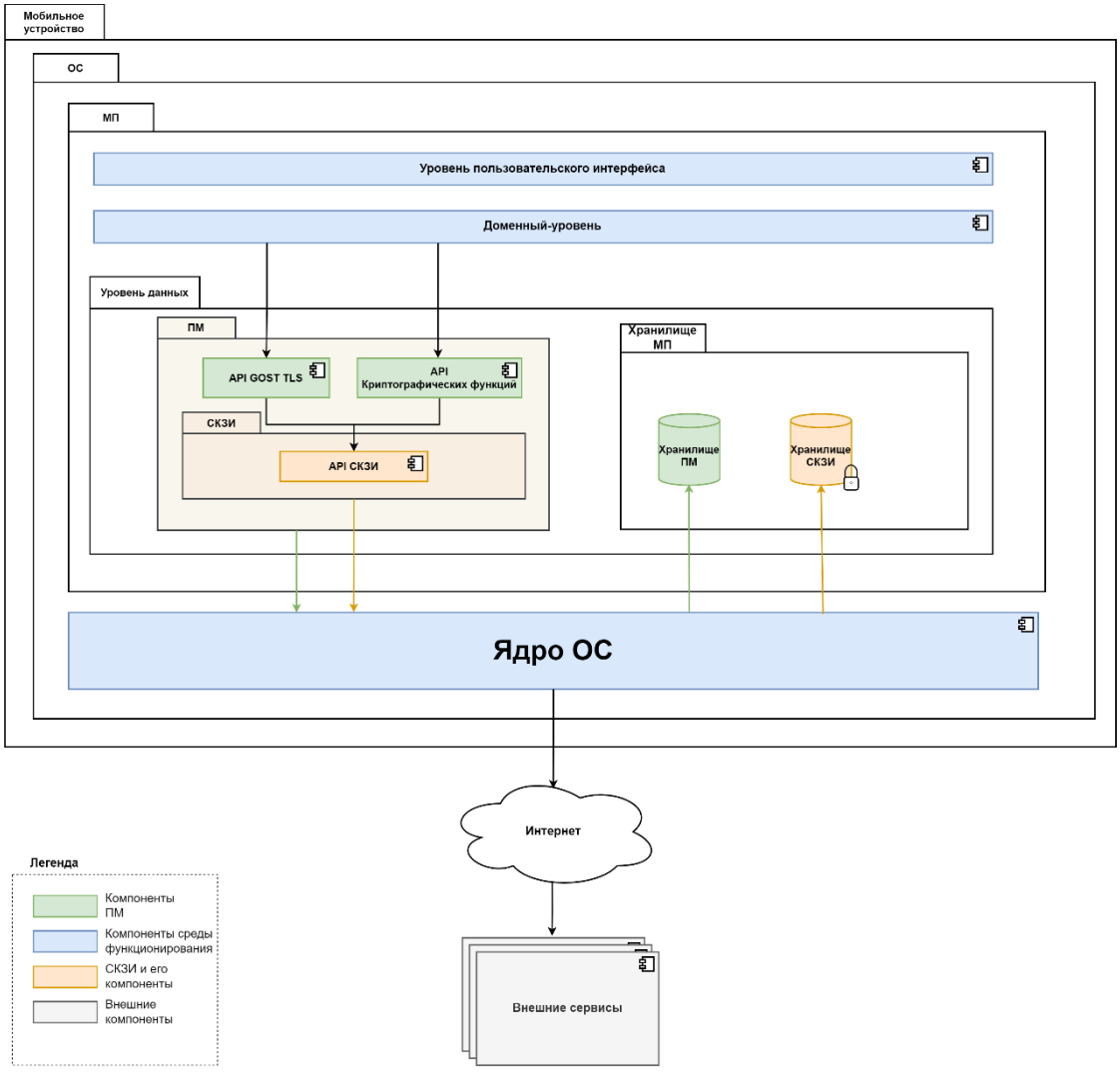

Рисунок 1 — Структурная схема решения

2.3 Обзор функций ПМ

ПМ должен предоставлять интерфейс для взаимодействия с МП и его компонентами. В Таблице 3 перечислен набор функций ПМ. Подробное описание методов МП, реализующих функции ПМ, содержится в пункте 3.2.

Таблица 3 — Функции ПМ

Номер функции	**Краткое описание функции**	**Доступность**	**Раздел, описывающий требования к реализации метода, соответствующего функции**
FN-1.1	Инициализация ПМ	Всегда	Пункт 3.2.1.1
FN-1.2	Инициализация работы с хранилищем СКЗИ	Должен быть выполнен метод «Инициализация ПМ»	Пункт 3.2.1.2
FN-1.3	Проверка контроля целостности	Всегда	Пункт 3.2.1.3
FN-1.4	Получение времени, оставшегося до окончания блокировки вызовов криптографических функций	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.1.4
FN-1.5	Получение даты и времени срока истечения действия пароля	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.1.5
FN-1.6	Смена пароля хранилища	Должны быть выполнены методы: • «Инициализация ПМ».	Пункт 3.2.1.6

Номер функции	Краткое описание функции	Доступность	Раздел, описывающий требования к реализации метода, соответствующего функции
		• «Инициализация работы с хранилищем СКЗИ».	
FN-1.7	Деинициализация хранилища	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.1.7
FN-1.8	Получение журнала использования ПМ	Должны быть выполнены методы: «Инициализация ПМ».	Пункт 3.2.1.8
FN-1.9	Регистрация установки СКЗИ	Должны быть выполнены методы: «Инициализация ПМ».	Пункт 3.2.1.9
FN-1.10	Настройка интерфейса работы с БДСЧ	Должны быть выполнены методы: «Инициализация ПМ».	Пункт 3.2.2.1
FN-1.11	Сохранение сертификатов криптографических ключей	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.3.5
FN-1.12	Сохранение САС	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с	Пункт 3.2.3.6

Номер функции	Краткое описание функции	Доступность	Раздел, описывающий требования к реализации метода, соответствующего функции
		хранилищем СКЗИ».	
FN-1.13	Формирование и хранение криптографических ключей пользователя	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.3.7
FN-1.14	Формирование и хранение транспортных криптографических ключей	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.3.8
FN-1.15	Создание запроса на СКПЭП пользователя	Должны быть выполнены методы: • «Инициализация ПМ». • «Формирование и хранение криптографических ключей пользователя». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.3.9
FN-1.16	Создание запроса на сертификат транспортного криптографического ключа	Должны быть выполнены методы: • «Инициализация ПМ». • «Формирование и хранение криптографических	Пункт 3.2.3.10

Номер функции	Краткое описание функции	Доступность	Раздел, описывающий требования к реализации метода, соответствующего функции
		ключей пользователя». • «Инициализация работы с хранилищем СКЗИ».	
FN-1.17	Единый метод формирования ключей и создания запроса на СКПЭП	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.3.11
FN-1.18	Единый метод формирования транспортных ключей и создания запроса на сертификат	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.3.12
FN-1.19	Подписание исходящих ЭС	Должны быть выполнены методы: • «Инициализация ПМ». • «Метод формирования криптографических ключей пользователя и создания запроса на СКПЭП пользователя». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.3.13

Номер функции	Краткое описание функции	Доступность	Раздел, описывающий требования к реализации метода, соответствующего функции
FN-1.20	Подписание транзакционного сообщения	Должны быть выполнены методы: • «Инициализация ПМ». • «Метод формирования криптографических ключей пользователя и создания запроса на СКПЭП пользователя». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.3.14
FN-1.21	Зашифрование исходящих ЭС	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.3.15
FN-1.22	Проверка подписи RAW	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.3.16
FN-1.23	Проверка подписи входящих ЭС	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с	Пункт 3.2.3.17

Номер функции	Краткое описание функции	Доступность	Раздел, описывающий требования к реализации метода, соответствующего функции
		хранилищем СКЗИ».	
FN-1.24	Расшифрование входящих ЭС	Должны быть выполнены методы: • «Инициализация ПМ». • «Формирование и хранение криптографических ключей пользователя». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.3.18
FN-1.25	Единый метод проверки ЭС	Должны быть выполнены методы: • «Инициализация ПМ». • «Формирование и хранение криптографических ключей пользователя». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.3.19
FN-1.26	Единый метод формирования ЭС	Должны быть выполнены методы: • «Инициализация ПМ». • «Метод формирования криптографических ключей пользователя и создания запроса	Пункт 3.2.3.20

Номер функции	Краткое описание функции	Доступность	Раздел, описывающий требования к реализации метода, соответствующего функции
		на СКПЭП пользователя». • «Инициализация работы с хранилищем СКЗИ».	
FN-1.27	Удаление данных из хранилища СКЗИ	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.3.21
FN-1.28	Удаление сертификатов ключей пользователя	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.3.22
FN-1.29	Извлечение сертификата ключей пользователя	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.3.23
FN-1.30	Просмотр срока действия сертификата	Должны быть выполнены методы: • «Инициализация ПМ». • «Инициализация работы с	Пункт 3.2.3.24

Номер функции	Краткое описание функции	Доступность	Раздел, описывающий требования к реализации метода, соответствующего функции
		хранилищем СКЗИ».	
FN-1.31	Установка защищенного соединения по протоколу TLS	Должны быть выполнены методы: • «Инициализация ПМ». • «Метод формирования криптографических ключей пользователя и создания запроса на транспортный сертификат». • «Инициализация работы с хранилищем СКЗИ».	Пункт 3.2.4

Приведенный список рекомендуется для реализации, но не является исчерпывающим. Он может быть изменен по усмотрению разработчика ПМ.

2.4 Требования к среде функционирования ПМ

В документе описан ПМ, функционирующий по крайней мере в одной из следующих сред:

- ОС Google Android не ниже версии 8;
- ОС Apple iOS не ниже версии 12;
- ОС Аврора (Aurora) не ниже версии 4.0;
- ОС HarmonyOS не ниже версии 2.0.

Перечень ОС, в которых функционирует ПМ, должен соответствовать эксплуатационной документации на СКЗИ.

Минимальные требования архитектуры процессоров:

- ПМ в составе МП для ОС Google Android и ОС HarmonyOS должен поддерживать архитектуры процессоров x86_64, ARM 32-bit и 64-bit.
- ПМ в составе МП для ОС Apple iOS должен поддерживать архитектуры процессоров на физическом устройстве (ARM 64-bit) и на эмуляторе (x86_64 и ARM 64-bit).
- ПМ в составе МП для ОС Аврора должен поддерживать архитектуры процессоров ARM 64-bit и x86_64.

2.5 Требования к интерфейсам ПМ

2.5.1 Пользовательские интерфейсы

Требования к пользовательским интерфейсам ПМ определяются разработчиком. Требования к программным интерфейсам СКЗИ определены в п. 3.4.4.

2.5.2 Программные интерфейсы

ПМ должен предоставлять набор программных интерфейсов для взаимодействия с МП.

ПМ может представлять собой единый функциональный модуль для интеграции в МП без необходимости подключения дополнительных пакетов и настроек. Допустимо использование транзитивных зависимостей, которые разрешаются в автоматическом режиме.

Для подключения к ПлЦР ПМ должен обеспечивать выполнение требований, предъявляемых в документах [11] в части криптографической защиты информации.

Для подключения к ПлЦР интерфейсы ПМ и МП должны обеспечивать МП выполнение требований документа [33].

Рекомендации по разработке функций ПМ приведены в разделе 3.2.

В ПМ для МП для ОС Apple iOS должны быть определены интерфейсные классы на языке Swift или другом языке программирования для ОС Apple iOS.

В ПМ для МП для ОС Google Android и HarmonyOS должны быть определены интерфейсные классы на языке Java, Kotlin или другом языке программирования для данных ОС.

В ПМ для МП для ОС Аврора должны быть определены интерфейсные классы на языке C++ или другом языке программирования для данной ОС.

3 Функциональные требования

Раздел содержит детальное описание функций ПМ, представленных в п. 2.3 документа, а также рекомендации и требования по их реализации.

3.1 Функциональная схема

На Рисунке 2 представлена схема решения с указанием функциональных взаимодействий между компонентами системы и типов этих взаимодействий.

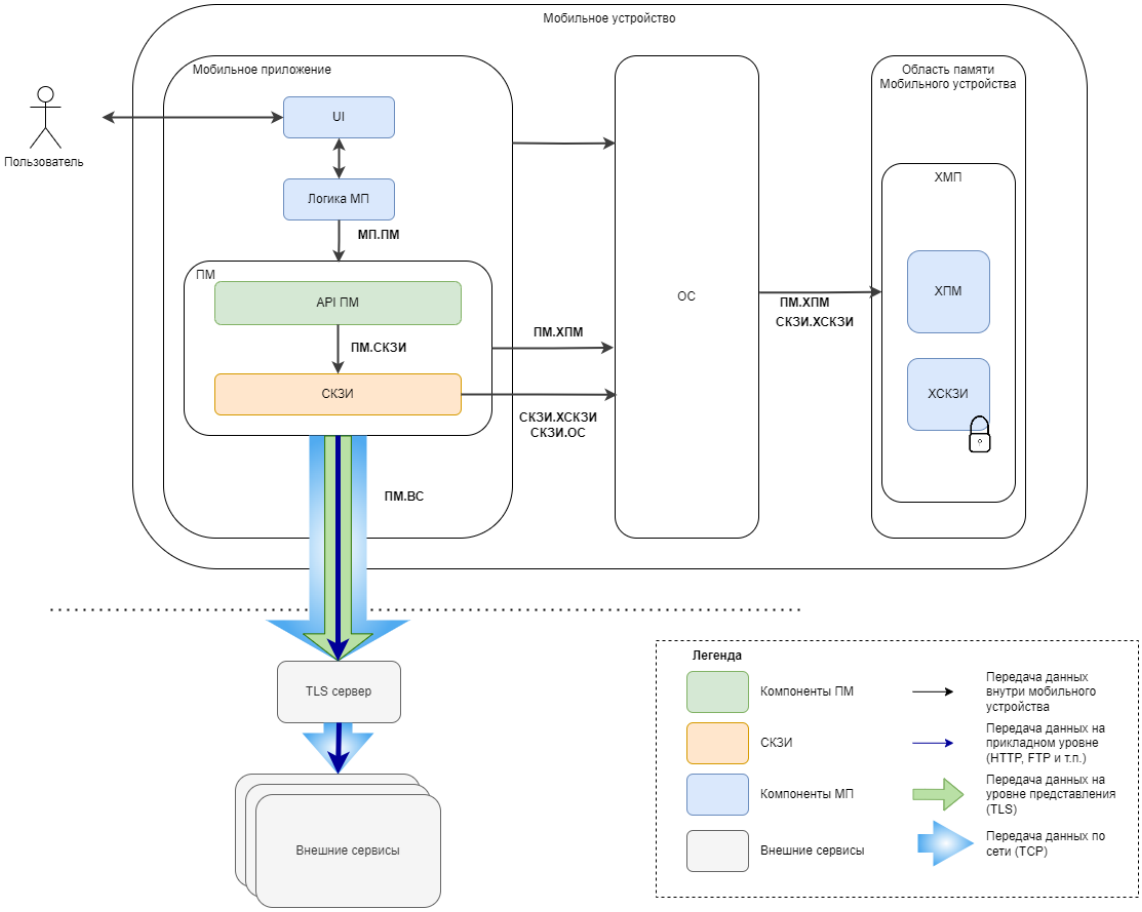

Рисунок 2 — Функциональная схема решения

3.1.1 Правила формирования названия потока

На Рисунке 2 обозначены информационные потоки. Название потока формируется по шаблону: <S>.<D>. Обозначение кодов в шаблоне представлено в Таблице 4.

Сокращения, используемые при формировании шаблона, представлены в Таблице 5.

Таблица 4 — Описание кодов шаблона подписи потоков

Код	Название	Описание
S	Source Источник	Перечень компонентов указан ниже
D	Destination Получатель	Перечень компонентов указан ниже

3.1.2 **Список компонентов**

В Таблице 5 представлено описание компонентов, отображенных на Рисунке 2.

Таблица 5 — Описание компонентов функциональной схемы решения

Компонент	Описание
TLS сервер	Сервер, обеспечивающий безопасную передачу данных посредством протокола TLS.
UI	Пользовательский интерфейс МП, с которым взаимодействует пользователь.
Внешние сервисы	Внешние сервисы, с которыми взаимодействует мобильное приложение, в том числе: сервисы участника ЦР, сервисы финансовых организаций, сервисы ЕСИА, сервисы ЕБС.
Логика МП	Основная функциональность МП, реализующая бизнес-логику, обработку данных и управление взаимодействиями между частями МП.
МП	Мобильное финансовое приложение, используемое при осуществлении дистанционного доступа к финансовым услугам и сервисам, в том числе при выполнении операций с ЦР.
ОС	Операционная система (Android, iOS, Аврора, HarmonyOS).
ПМ	Программный модуль, имеющий в своем составе сертифицированное СКЗИ для осуществления криптографических преобразований.
ППИАРІ ПМ	Набор функций и процедур, предоставляемый ПМ, для взаимодействия с МП и его компонентами.
СКЗИ	Средство криптографической защиты информации.
ХМП	Область памяти, которая выделяется мобильной ОС для обеспечения работы МП.

Компонент	Описание
ХПМ	Внутреннее хранилище ПМ.
ХСКЗИ	Защищенное хранилище СКЗИ.

3.1.3 **Функции по информационным потокам**

В Таблице 6 представлена связь информационных потоков с соответствующими им функциями.

Таблица 6 — Связь потоков с функциями ПМ

Название потока	Функции
МП.ПМ	1. Инициализация ПМ. 2. Инициализация защищенного хранилища. 3. Деинициализация защищенного хранилища. 4. Проверка контроля целостности. 5. Сохранение сертификатов криптографических ключей. 6. Сохранение САС. 7. Формирование и хранение криптографических ключей пользователя. 8. Формирование и хранение транспортных криптографических ключей. 9. Создание запроса на СКПЭП пользователя. 10. Создание запроса на сертификат транспортного криптографического ключа. 11. Единый метод формирования ключей и создания запроса на СКПЭП. 12. Единый метод формирования транспортных ключей и создания запроса на сертификат. 13. Подписание исходящих ЭС. 14. Подписание транзакционного сообщения. 15. Зашифрование исходящих ЭС. 16. Проверка подписи входящих ЭС. 17. Расшифрование входящих ЭС. 18. Проверка входящих ЭС. 19. Формирование исходящих ЭС.

Название потока	Функции
	20. Установка защищенного канала связи. 21. Получение времени, оставшегося до окончания блокировки вызовов криптографических функций. 22. Удаление защищенного хранилища. 23. Удаление сертификата. 24. Извлечение сертификата криптографических ключей. 25. Получение срока действия сертификата. 26. Смена пароля. 27. Получение даты и времени истечения срока действия пароля. 28. Настройка интерфейса работы с БДСЧ. 29. Получение журнала использования ПМ. 30. Регистрация установки СКЗИ.
ПМ.СКЗИ	1. Инициализация СКЗИ. 2. Инициализация защищенного хранилища. 3. Удаление защищенного хранилища. 4. Проверка контроля целостности. 5. Сохранение сертификатов криптографических ключей. 6. Сохранение CAC. 7. Получение CAC. 8. Формирование пары криптографических ключей. 9. Сохранение пары криптографических ключей. 10. Формирование запроса на сертификат криптографических ключей. 11. Подписание ЭС определенного типа. 12. Зашифрование ЭС. 13. Расшифрование ЭС. 14. Проверка ЭП определенного типа. 15. Поиск сертификата по шаблону. 16. Удаление сертификата. 17. Извлечение сертификата. 18. Установка одностороннего и двухстороннего ГОСТ-TLS соединения 19. Получение срока действия сертификата.

Название потока	Функции
	20. Получение времени истечения срока действия пароля. 21. Настройка интерфейса работы с БДСЧ. 22. Смена пароля защищенного хранилища.
ПМ.ХПМ	1. Сохранение журнала использования ПМ. 2. Получение журнала использования ПМ. 3. Сохранение времени блокировки хранилища. 4. Получение времени блокировки защищенного хранилища. 5. Удаление данных из хранилища ПМ.
СКЗИ.ХСКЗИ	1. Функции СКЗИ, для корректной работы которых необходимо взаимодействие с хранилищем СКЗИ.
СКЗИ.ОС	1. Функции СКЗИ, для корректной работы которых необходимо взаимодействие с ОС: a) сбор энтропии; b) запрос ввода пароля от сегмента хранилища; c) запрос смены пароля от сегмента хранилища.
ПМ.ВС	1. Одностороннее и двустороннее ГОСТ-TLS соединение для осуществления взаимодействия с сервисами финансовых организаций. 2. При взаимодействии с ЕСИА одностороннее ГОСТ-TLS-соединение для осуществления функции аутентификации пользователя в соответствии с документом [31]. 3. При взаимодействии с ЕБС одностороннее ГОСТ-TLS-соединение для осуществления функции аутентификации пользователя.

3.2 Описание классов и методов ПМ

ПМ должен обеспечивать корректную обработку аварийных ситуаций, вызванных неверным форматом, размером или недопустимыми значениями входных данных. В случае возникновения ошибок при выполнении функций ПМ должен вернуть негативный результат с кодом и описанием причины. Ошибки не должны приводить к нештатным ситуациям, ПМ должен оставаться в рабочем состоянии. Описание кодов и возможных причин ошибок приведено в разделе 3.2.6.

Необходимость вывода ошибок и отображаемая пользователю информация в интерфейсе МП должны определяться разработчиком МП.

Передача параметров при вызове функций СКЗИ должна быть реализована в соответствии с рекомендациями производителей ОС.

Не рекомендуется использовать пустые значения входных параметров функций.

Все методы должны прерывать выполнение и возвращать код ошибки «Требуется инициализация ПМ», если не был вызван метод инициализации ПМ.
Если вызов метода произошел во время выполнения метода инициализации, ПМ не должен прерывать выполнение метода. Ранее вызванная инициализация должна завершиться, после чего выполняется обработка вызова.

3.2.1 **Класс Core**

Класс Core содержит основные методы ПМ.

3.2.1.1 Метод init (filesWithCheckSums)

Метод инициализации ПМ и СКЗИ.

Входные данные

- filesWithCheckSums — массив пар строк. Массив содержит эталонные значения хэшей, рассчитанные от файлов и ресурсов МП, целостность которых необходимо проверить, и пути к ним. Хэши должны быть рассчитаны в соответствии с документом [30].

Выходные данные

- result — строка, в случае успеха возвращает «ОК».

Требования

Выполнение проверок.

1. В случае повторного вызова метода инициализации вернуть код ошибки «Инициализация ПМ уже выполнена».
2. Проверить:
    a) наличие root-доступа для Google Android, HarmonyOS и Аврора или jailbreak для Apple iOS;
    b) маскировку наличия у пользователя прав суперпользователя (Root Cloaking Detection);
    c) использование устройством модифицированной (неофициальной) прошивки (Custom Firmware Detection);
    d) запуск приложения на эмуляторе устройства (Emulator Detection);

e) использование функций отладчика при запуске приложения (Debug Detection);
f) наличие программных средств, позволяющих изменять поведение приложений или ОС за счёт перехвата вызовов и других способов воздействия на программную среду (Hooks Detection).

3. При наличии хотя бы одного признака небезопасной работы из перечисленных в пункте 2, прервать выполнение функции и вернуть код, соответствующий ошибке «Невозможно вызвать криптографические функции» (см. раздел 3.2.6).
4. Инициализировать СКЗИ для обеспечения корректной работы функций криптографической защиты, инициализировать область памяти в СКЗИ.
5. При получении уведомления от СКЗИ об отсутствии или недостаточном количестве информационной энтропии выполнение метода инициализации должно приостанавливаться, а после сбора достаточного количества энтропии корректного возобновляться.
6. Вызвать метод checkIntegrity (filesWithCheckSums). На вход передать значения, полученные во входных параметрах метода инициализации.
7. Установить в хранилище СКЗИ серверный ГОСТ-TLS-сертификат и связанную с ним цепочку сертификатов, включающую все промежуточные сертификаты, сертификат промежуточного УЦ и сертификат корневого УЦ:
    a) корневые (самоподписанные) сертификаты УЦ БР и УЦ Безопасности, промежуточный сертификат ПУЦ УНЭП должны быть включены в ресурс МП;
    b) в МП должен быть реализован механизм ротации сертификатов в случае окончания срока действия сертификата или при его компрометации.
8. Вернуть «ОК».
9. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.1.2 Метод initCryptContext (storeId, userId)

Метод инициализации контейнера криптографических ключей.

Входные данные[3]

- storeId — строка, уникальный идентификатор сегмента хранилища СКЗИ;
- userId — строка, уникальный идентификатор пользователя МП. Необязательный параметр.

Выходные данные

- cryptContext — структура данных, полученная от СКЗИ и содержащая информацию о состоянии и параметрах работы с криптографическим провайдером.

Требования

Для каждого пользователя может быть создано несколько защищенных хранилищ с доступом по паролю. Идентификаторы хранилищ должны быть уникальными для уникальных пользователей и формироваться на стороне МП (требования к идентификатору хранилища СКЗИ и идентификатору пользователя МП см. в пунктах 3.2.3.2 и 3.3.2).

1. Если идентификатор пользователя не передан, то перейти к шагу 6 и выполнить инициализацию контекста для корневого хранилища.

2. Проверить, был ли ранее сохранен идентификатор пользователя МП во внутреннем хранилище ПМ.

3. Если идентификатор пользователя найден, то перейти к шагу 6.

4. Если идентификатор пользователя не найден, то вызвать метод удаления найденных сегментов хранилища СКЗИ и всей ключевой информации - deleteStore (storeId).

5. Сохранить идентификатор пользователя и идентификатор хранилища СКЗИ во внутреннем хранилище ПМ, если они не были ранее сохранены во внутреннем хранилище ПМ.

6. Вызвать метод инициализации контекста криптографического провайдера в СКЗИ в соответствии с уникальным идентификатором хранилища:

[3] Список входных данных может быть расширен в зависимости от особенностей реализации СКЗИ.

a) если защищенный ключевой контейнер не был создан ранее, необходимо его создать и установить пароль;

b) в ответ СКЗИ должно возвращать объект cryptContext, содержащий структуру данных с информацией о криптопровайдере, для выполнения последующих криптографических операций и управления криптографическими ресурсами.

7. Вернуть полученный объект cryptContext.

8. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.1.3 Метод checkIntegrity (filesWithCheckSums)

Метод проверки, который используется для обнаружения изменений в данных или программном обеспечении, и позволяет определить факт несанкционированного вмешательства.

Входные данные

- filesWithCheckSums — массив, содержащий эталонные значения хэшей, рассчитанных от файлов и ресурсов МП, целостность которых необходимо проверить, и пути к ним (структура элементов массива может определяться разработчиком ПМ в зависимости от особенностей ОС).

Выходные данные

- result — строка. В случае успеха возвращает код «ОК», в случае неуспеха — код “INVALID”.

Требования

1. Рассчитать контрольные значения хэшей для каждого файла из filesWithCheckSums. Для этого:
   a) инициировать вызов в СКЗИ функции вычисления хэшей необходимого файла;
   b) хэши должны рассчитываться в соответствии с документом [30].
2. Сравнить вычисленные хэши и значения контрольных сумм из filesWithCheckSums для проверяемого файла.

3. В случае совпадения всех вычисленных хэшей с эталонными значениями вернуть код «ОК».
4. В случае хотя бы одного не совпадения — вернуть код «INVALID».
5. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ необходимо прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.1.4 Метод getAccessTime (storeId)

Метод используется для получения времени, оставшегося до окончания блокировки вызовов криптографических функций.

Входные данные

- storeId — строка, идентификатор хранилища СКЗИ.

Выходные данные

- accessTime — время в секундах, оставшееся до конца блокировки.

Требования

1. Извлечь из внутреннего хранилища ПМ значение времени окончания блокировки по идентификатору защищенного хранилища СЗКИ.
2. Вычислить время, оставшееся до окончания блокировки вызовов криптографических функций, как дельту между временем окончания блокировки и текущим временем. При расчете дельты привести время к единому часовому поясу по стандарту UTC:
    a) если время окончания блокировки для хранилища не найдено, то вернуть 0;
    b) если в результате вычисления дельты получено отрицательное значение, то вернуть 0.
3. В случае успешного вычисления вернуть время с точностью до секунды.
4. В случае неуспешного выполнения какого-либо из перечисленных выше действий необходимо прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.1.5 Метод getPasswordExpirationTime (storeId, timeZone)

Метод получения времени действия пароля к хранилищу СКЗИ.

Входные данные

- storeId — строка, идентификатор хранилища СКЗИ;
- timeZone — строка, часовой пояс, в котором необходимо вернуть дату и время истечения срока действия пароля.

Выходные данные

- expirationTime — datatime, дата и время срока истечения действия пароля в часовом поясе из timeZone.

Требования

1. Вызвать функцию СКЗИ для получения времени истечения срока действия пароля для полученного идентификатора хранилища СКЗИ.
2. Привести полученное временное значение к часовому поясу, полученному в параметре timeZone.
3. Вернуть дату и время истечения срока действия пароля.
4. В случае неуспешного выполнения какого-либо из перечисленных выше действий необходимо прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.1.6 Метод changePassword (storeId)

Метод изменения пароля к хранилищу СКЗИ.

Входные данные

- storeId — строка, идентификатор хранилища СКЗИ.

Выходные данные

- result — строка, в случае успеха возвращает код «ОК».

Требования

1. Вызвать метод СКЗИ для изменения пароля защищенного хранилища. Интерфейс для смены пароля сегмента хранилища должен вызываться с помощью СКЗИ.
2. После успешного выполнения функции вернуть «ОК».
3. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.1.7 Метод releaseContext (cryptoContext)

Метод деинициализации работы с защищенным хранилищем.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере.

Выходные данные

- result — строка, в случае успеха возвращает код «ОК»

Требования

1. Вызвать функцию СКЗИ для деинициализации cryptContext.
2. Если установлено одностороннее или двустороннее ГОСТ-TLS соединение, то при завершении работы с хранилищем соединение необходимо разрывать.
3. После успешного выполнения функции вернуть «ОК».
4. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ необходимо прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.1.8 Метод getAuditLog (timeBegin, timeEnd)

Метод получения журнала аудита.[4]

Входные данные

- timeBegin — Unix Timestamp, дата начала фиксации событий;
- timeEnd — Unix Timestamp, дата окончания фиксации событий.

Выходные данные

- auditLog — массив структур данных, содержащих следующие элементы:
    - id — строка, уникальный идентификатор действия;
    - createdAt — Unix Timestamp, дата и время действия;
    - functionName — строка, название вызванного метода;
    - result — строка, результат выполнения функции (SUCCESS, FAIL);
    - inputData — строка, список входных данных, может быть пустым;
    - outputData — строка, список выходных данных, может быть пустым;
    - storeId — строка, идентификатор хранилища, может быть пустым;

[4] Журнал формируется за период времени по всем хранилищам в рамках одного мобильного устройства. При вызове функции необходимо дополнительно применять фильтрацию при выводе логов пользователю.

- errorCode — целое число, код ошибки, может быть пустым;
- errorMessage — строка, описание ошибки, может быть пустым.

Требования

1. Получить из внутреннего хранилища ПМ журнал взаимодействия с ПМ, отфильтрованный по полученному временному периоду. Требования к хранению данных описаны в пункте 3.4.

2. Отсортировать извлеченные данные по возрастанию даты и времени действия (createdAt).

3. Вернуть извлеченный отсортированный список.

4. В случае неуспешного выполнения какого-либо из перечисленных выше действий прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.1.9 Метод registerInstallation ()

Метод регистрации установки и учета СКЗИ.

Входные данные

- отсутствуют

Выходные данные

- id — строка, уникальный идентификатор установки СКЗИ;
- date — datatime, дата и время установки СКЗИ в UTC.

Требования

1. Метод должен выполнять действия по регистрации экземпляра СКЗИ с учетом имеющихся механизмов СКЗИ.
2. Разработчик ПМ должен самостоятельно продумать алгоритм регистрации, если используемое СКЗИ не предоставляет механизм поэкземплярного учета.
3. В результате успешного выполнения метода вернуть уникальный идентификатор установки экземпляра СКЗИ и дату и время установки в UTC.
4. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ необходимо

прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.2 **Класс Rng**

3.2.2.1 Метод customizeEntropyScreen (textLabels, backgroundColor, controlColor)

Метод предназначен для адаптации дизайна экрана работы с БДСЧ к стилю МП. Реализация работы БДСЧ и возможность адаптации дизайна экрана работы с БДСЧ должны соответствовать эксплуатационной документации на СКЗИ.

Входные данные[5]

- textLabels — массив строк, содержащий тексты для замены надписей;
- backgroundColor — строка, содержащая шестнадцатеричный код цвета замены для фона;
- controlColor — строка, содержащая шестнадцатеричный код цвета замены для элементов управления.

Выходные данные

- result — строка, в случае успеха возвращает код «ОК».

Требования

1. Вызвать соответствующую функцию СКЗИ для адаптации дизайна экрана работы с БДСЧ. На вход передать полученные параметры.
2. Вернуть «ОК».
3. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3 **Класс Operations**

Класс Operations содержит криптографические методы ПМ.

[5] Список входных данных может быть расширен в зависимости от возможностей СКЗИ по изменению дизайна экрана сбора энтропии.

3.2.3.1 Общие требования к криптографическим методам

Все методы должны прерывать выполнение и возвращать код ошибки в соответствии с описанием в разделе 3.2.6, если:

- отсутствует инициализированное хранилище СКЗИ для полученного cryptContext;
- пароль хранилища неверный;
- целостность файлов или ресурсов МП нарушена.

3.2.3.2 Доступ к криптографическим функциям и защищенному хранилищу в СКЗИ

1. Защищенное хранилище СКЗИ должно быть сегментировано для возможности хранения различных пар криптографических ключей.
2. СКЗИ должно иметь возможность создания отдельного защищенного хранилища с доступом по паролю.
3. Доступ к определенному сегменту хранилища СКЗИ должен выполняться в соответствии с уникальным идентификатором хранилища и паролем.
4. Пароль к хранилищу устанавливается при создании хранилища.
5. В рамках одной сессии пользователя после первой инициализации хранилища по идентификатору и паролю доступ к хранилищу криптографических ключей осуществляется по полученному от СКЗИ объекту cryptContext.
6. При инициализации хранилища необходимо проверять срок действия пароля при помощи функции СКЗИ. Если срок действия пароля истек, то необходимо вернуть код, соответствующий ошибке «Истек срок действия пароля» (см. раздел 3.2.6). Доступ к криптографическим функциям не должен предоставляться до тех пор, пока пароль не будет изменен.

3.2.3.3 Блокировка доступа

Если вызов любого метода СКЗИ для заданного идентификатора хранилища завершается ошибкой, связанной с неверно введенным паролем, ПМ должен:

1. Прервать выполнение метода.

2. Обновить во внутреннем хранилище информацию о том, сколько раз подряд пароль был введен неверно.

3. При достижении 5 неудачных попыток аутентификации необходимо сохранить во внутреннем хранилище время окончания блокировки для данного экземпляра хранилища в UTC, рассчитанное как текущее время + 30 минут. Если ранее время окончания блокировки уже сохранялось, необходимо сохранить новое значение.

4. Вернуть код, соответствующий ошибкам «Истек срок действия пароля» или «Неверный пароль» (см. раздел 3.2.6).

Если пароль введен корректно и выполнен вызов метода СКЗИ, то счетчик неуспешных попыток должен сбрасываться.

Контроль неудачных попыток и блокировка доступа могут быть реализованы на стороне СКЗИ, если СКЗИ предоставляет такую возможность.

Количество неудачных попыток и время блокировки доступа должны настраиваться константами на этапе сборки, храниться в неизменяемых ресурсах и соответствовать правилам использования СКЗИ. Значения параметров могут быть уточнены на этапе разработки ПМ.

Значения по умолчанию:

- количество неудачных попыток — 5;
- время блокировки доступа к криптографическим функциям — 30 минут.

3.2.3.4 Проверка сертификатов и криптографических ключей

При использовании закрытого криптографического ключа СКЗИ должно выполнять проверки срока действия закрытого ключа. Если срок действия ключа истек, необходимо прервать выполнение метода и вернуть соответствующую ошибку.

При использовании сертификатов криптографических ключей необходимо получить доступ к ранее сохраненному САС посредством СКЗИ и проверить

наличие сертификата в списке. Если сертификат недействителен, необходимо прервать выполнение метода и вернуть соответствующую ошибку.

3.2.3.5 Метод saveCertificates (cryptContext, certDate, certStoreType, certFormat)

Метод, сохраняющий сертификаты криптографических ключей.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;
- certData — структура данных, содержащая массивы байт и псевдонимы, под которыми сертификаты сохраняются в хранилище СКЗИ;
- certStoreType — строка, содержащая тип хранилища сертификатов. Перечень возможных значений определяется функциональностью предоставляемой СКЗИ;
- certFormat — строка, содержащая формат сертификатов.

Выходные данные

- result — строка, в случае успеха возвращается «ОК».

Требования

1. Вызвать функцию СКЗИ для сохранения полученных сертификатов в указанное хранилище:
    a) если не удалось сохранить сертификат, прервать выполнение метода и вернуть соответствующую причину ошибки;
    b) при необходимости преобразовать сертификаты в нужный формат.
2. Вернуть «ОК».
3. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.6 Метод saveCrl (crlData, crlFormat)

Метод, сохраняющий САС.

Входные данные

- crlData — массив байт, содержащий САС;
- crlFormat — строка, содержащая формат САС.

Выходные данные

- result — строка, в случае успеха возвращается «ОК».

Требования

1. Вызвать функцию СКЗИ для сохранения САС в хранилище сертификатов СКЗИ:
    a) если не удалось сохранить САС, прервать выполнение метода и вернуть соответствующую причину ошибки;
    b) при необходимости преобразовать САС в нужный формат.
2. Вернуть «ОК».
3. В случае других непредвиденных ошибок прервать выполнение метода и вернуть соответствующую причину ошибки.

3.2.3.7 Метод genKeyPair (cryptContext, lenghtKey)

Метод формирования и сохранения криптографических ключей пользователя.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;
- lenghtKey — целое число, длина закрытого ключа в битах.

Выходные данные

- result — строка, в случае успеха возвращается «ОК».

Требования

1. Вызвать метод СКЗИ для формирования криптографических ключей пользователя:
    a) в соответствии с указанной длиной;
    b) ключи должны формироваться в соответствии с [2],[20].
2. Вызвать метод СКЗИ для сохранения сформированных криптографических ключей пользователя в соответствии с полученным cryptContext:
    a) пара ключей должна сохраняться в соответствующем защищенном хранилище;
    b) закрытый ключ должен быть неэкспортируемым.
3. Если ключи успешно созданы и сохранены, необходимо возвращать код «ОК».
4. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать

выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.8 Метод genTlsKeyPair (cryptContext, lenghtKey)

Метод формирования и сохранения транспортных криптографических ключей.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;
- lenghtKey — целое число, длина закрытого ключа в битах.

Выходные данные

- result — строка, в случае успеха возвращается «ОК».

Требования

1. Вызвать метод СКЗИ для формирования криптографических ключей:
    a) в соответствии с указанной длиной;
    b) ключи должны формироваться в соответствии с документами [2],[20].
2. Вызвать метод СКЗИ для сохранения сформированных криптографических ключей в соответствии с полученным cryptContext:
    a) пара ключей должна сохраняться в соответствующем защищенном хранилище;
    b) закрытый ключ должен быть неэкспортируемым.
3. Если ключи успешно созданы и сохранены, необходимо вернуть код «ОК».
4. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.9 Метод createCertRequest (cryptContext, certRequestParams, isSubstitution)

Метод создания запроса на СКПЭП пользователя.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;

- certRequestParams — структура данных, содержащая атрибуты и поля для включения в СКПЭП. В случае формирования запроса на СКПЭП с целью его использования для операций с ЦР структура certRequsetParams должна соответствовать документу [10];
- isSubstitution — флаг замены существующего сертификата. Значение false, если запрос на сертификат создается первый раз; значение true для запроса на обновление существующего сертификата.

Выходные данные

- request — массив байт, содержащий созданный запрос в формате PKCS#10/PKCS#7.

Требования

1. Вызвать функции СКЗИ для формирования запроса на СКПЭП пользователя МП:
    a) в соответствии с полученными входными данными. Криптографическая пара ключей определяется исходя из полученного cryptContext;
    b) если isSubstitution=true, сформированный запрос подписать старым криптографическим ключом.
2. В случае успеха при первом запросе на сертификат вернуть результат в формате PKCS#10. При обновлении сертификата вернуть результат в формате PKCS#7.
3. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.10 Метод createTlsCertRequest (cryptContext, certRequestParams, isSubstitution)

Метод создания запроса на сертификат транспортного криптографического ключа.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;

- certRequestParams — структура данных, содержащая атрибуты и поля для включения в сертификат транспортного криптографического ключа. В случае формирования запроса на транспортный сертификат с целью его использования для операций с ЦР структура certRequsetParams должна соответствовать корневому УЦ Безопасности участника ПлЦР и документу [10];
- isSubstitution — флаг замены существующего сертификата. Значение false, если запрос на сертификат создается первый раз; значение true для запроса на обновление существующего сертификата.

Выходные данные

- request — массив байт, содержащий созданный запрос в формате PKCS#10/PKCS#7.

Требования

1. Вызвать функции СКЗИ для формирования запроса на сертификат:
    a) если isSubstitution=true, сформированный запрос подписать старым криптографическим ключом.
2. В случае успеха при первом запросе на сертификат вернуть результат в формате PKCS#10. При обновлении сертификата вернуть результат в формате PKCS#7.
3. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.11 Метод genKeyPairAndRequest (cryptContext, certRequestParams, lenghtKey, isSubstitution)

Единый метод формирования ключей и создания запроса на СКПЭП.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;
- certRequestParams — структура данных, содержащая атрибуты и поля для включения в СКПЭП. В случае формирования запроса на СКПЭП с целью

его использования для операций с ЦР структура certRequsetParams должна соответствовать документу [10];

- lenghtKey — целое число, длина закрытого ключа в битах;
- isSubstitution — флаг замены существующего сертификата. Значение false, если запрос на сертификат создается первый раз; значение true для запроса на обновление существующего сертификата.

Выходные данные

- request — массив байт, содержащий созданный запрос в формате PKCS#10/PKCS#7.

Требования

1. Последовательно вызвать методы:
    a) genKeyPair (cryptContext, lenghtKey):
        i. На вход передать контекст хранилища и длину ключа.
    b) createCertRequest (cryptContext, certRequestParams, isSubstituion):
        i. На вход передать контекст хранилища, набор атрибутов для включения в СКПЭП и флаг замены сертификата.
2. Вернуть сформированный запрос на сертификат в формате PKCS#10 при создании запроса на сертификат впервые или в формате PKCS#7 при обновлении сертификата.
3. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций ПМ или СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.12 Метод genTlsKeyPairAndRequest (cryptContext, certRequestParams, lenghtKey, isSubstituion)

Единый метод формирования транспортных ключей и создания запроса на сертификат.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;
- certRequestParams — структура данных, содержащая атрибуты и поля для включения в транспортный сертификат. В случае формирования запроса на

транспортный сертификат с целью его использования для операций с ЦР структура certRequsetParams должна соответствовать корневому УЦ Безопасности участника ПлЦР и документу [10];

- lenghtKey — целое число, длина закрытого ключа в битах;
- isSubstitution — флаг замены существующего сертификата. Значение false, если запрос на сертификат создается первый раз; значение true для запроса на обновление существующего сертификата.

Выходные данные

- request — массив байт, содержащий созданный запрос в формате PKCS#10/PKCS#7.

Требования

1. Последовательно вызвать методы:
    a) genTlsKeyPair (cryptContext, lenghtKey):
        i. На вход передать контекст хранилища и длину ключа.
    b) createTlsCertRequest (cryptContext, certRequestParams, isSubstituion):
        i. На вход передать контекст хранилища, набор атрибутов для включения в СКПЭП и флаг замены сертификата.
2. В случае успеха при первом запросе на сертификат вернуть результат в формате PKCS#10. При обновлении сертификата вернуть результат в формате PKCS#7.
3. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций ПМ или СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.13 Метод createSignature (cryptContext, data, certId, signatureType)

Метод подписания исходящих ЭС.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;
- data — массив байт, содержащий ЭС;

- certId — строка, идентификатор сертификата подписи;
- signatureType — строка, тип операции (возможные типы: CMScert, CMSid).

Выходные данные

- signature — ЭП типа CMScert либо CMSid, закодированная в base64. Сертификат или идентификатор сертификата включен в структуру подписи.

Требования

1. Вызвать функцию СКЗИ по формированию ЭП с передачей массива байтов, содержащего ЭС:
    a) ЭП должна формироваться в соответствии с указанным типом операции и идентификатором сертификата подписи;
    b) алгоритмы, использующиеся при формировании ЭП, должны соответствовать [2],[20].
2. Закодировать ЭП по алгоритму base64. В случае формирования ЭП целью ее использования для операций с ЦР алгоритмы кодирования должны соответствовать документам [11].
3. В случае успешного выполнения вызова метода вернуть сформированную ЭП.
4. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.14 Метод createRawSignature (cryptContext, data, certId)

Метод подписания транзакционного сообщения.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;
- data — массив байт, содержащий ЭС;
- certId — строка, идентификатор сертификата подписи.

Выходные данные

- signature — массив байт, который содержит подпись ЭС сообщения типа RAW, закодированную в base64.

Требования

1. Вызвать функцию СКЗИ по формированию ЭП с передачей массива байтов, содержащего ЭС:
    a) ЭП должна формироваться в соответствии с указанным типом операции и идентификатором сертификата подписи;
    b) алгоритмы, использующиеся при формировании ЭП, должны соответствовать [2],[20].
2. Если функции СКЗИ возвращают ЭП в формате big-endian, то данный массив байт инвертировать в формат little-endian.
3. Закодировать ЭП по алгоритму base64. В случае формирования ЭП целью ее использования для операций с ЦР алгоритмы кодирования должны соответствовать документам [11].
4. В случае успешного выполнения вызова метода вернуть ЭП типа RAW.
5. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.15 Метод encryptData (cryptContext, data, certIds, certFormat, certPattern)

Метод зашифрования исходящих ЭС.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;
- data — массив байт, содержащий ЭС;
- certIds — массив сертификатов получателя в формате X509 (необязательный параметр);
- certFomat — строка, формат сертификатов (необязательный параметр);
- certPattern — шаблон поиска сертификата, содержащий массив пар ключ − значение. В качестве ключа необходимо указать название поля, по которому будет осуществляться поиск, в качестве значения — содержимое поля (необязательный параметр).

Выходные данные:

- encryptedData — строка, содержащая массив байт зашифрованных сообщений.

Требования

1. Во входных данных передать массив и формат сертификатов, или шаблон для поиска сертификатов.
2. Выполнить сжатие полученного массива байт ЭС. В случае использования ЭП для операций с ЦР алгоритмы сжатия должны соответствовать документам [11].
3. Вызвать функцию СКЗИ по зашифрованию ЭС с передачей массива байтов, содержащего ЭС:
    a) зашифрование должно выполняться с использованием полученных сертификатов получателя;
    b) если получен шаблон для поиска сертификата, извлечь сертификат с использованием соответствующей функции СКЗИ по заданному шаблону поиска;
    c) алгоритмы, использующиеся при зашифровании, должны соответствовать [3],[21],[22],[23].
4. Закодировать зашифрованное сжатое ЭС по алгоритму base64. В случае формирования ЭП с целью ее использования для операций с ЦР алгоритмы кодирования должны соответствовать документам [11].
5. В случае успешного выполнения вызова метода вернуть массив байт зашифрованного сообщения.
6. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.16 Метод verifyRawSignature (cryptContext, data, signature, certRawId)

Метод проверки подписи с типом RAW.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;
- data — массив байт, содержащий сообщение;

- signature — массив байт, содержащий ЭП типа RAW для сообщения из data;
- certRawId — строка, содержащая идентификатор сертификата для проверки ЭП типа RAW.

Выходные данные

- result — строка, в случае успеха возвращает код «ОК», в случае неуспеха возвращает код «INVALID».

Требования

1. С помощью функции СКЗИ найти сертификат в хранилище СКЗИ в соответствии с полученным cryptContext и certRawId.
2. Вызвать функцию СКЗИ для проверки ЭП:
    a) проверку необходимо выполнять в соответствии с типом подписи RAW и извлеченным сертификатом;
    b) алгоритмы, использующиеся при вычислении значения ЭП, должны соответствовать [2],[20].
3. В случае успешной проверки ЭП вернуть код «ОК». Если ЭП неверная, вернуть код «INVALID».
4. В случае неуспешного выполнения какого-либо из перечисленных выше действий или если получены ошибки при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.17 Метод verifySignature (cryptContext, data, signature, signatureType)

Метод проверки подписи входящих ЭС.

Входные данные

- cryptContext — массив объектов, содержащий информацию о используемых криптографических провайдерах;
- signature — массив строк, содержащий массивы байт ЭП, закодированные в base64. Может быть передано несколько ЭП;
- data — массив байт, содержащий расшифрованное, декодированное и разжатое ЭС;
- signatureType — массив строк, содержащий типы подписей (возможные значения: CMScert, CMSid).

Выходные данные

- result — строка, в случае успеха возвращает код «ОК», в случае неуспеха возвращает код «INVALID»;
- certIds — массив строк, содержащий идентификаторы сертификатов, с помощью которых была выполнена проверка ЭП. Необязательный параметр.

Требования

1. Декодировать полученные ЭП по алгоритму base64. В случае использования ЭП для операций с ЦР алгоритмы декодирования должны соответствовать документам [11].
2. Если передано несколько ЭП, проверить все полученные ЭП.
3. Для типа CMScert из входящего сообщения извлечь сертификат отправителя, содержащий открытый ключ и данные.
4. Для типа CMSid из входящего сообщения извлечь данные и идентификатор сертификата открытого ключа. Посредством функции СКЗИ выполнить поиск сертификата в хранилище СКЗИ в соответствии с полученным cryptContext.
5. Вызвать функцию СКЗИ для проверки каждой ЭП по отдельности:
    a) проверку выполнить в соответствии с указанным типом подписи и извлеченным открытым ключом;
    b) алгоритмы, использующиеся при вычислении значения ЭП, должны соответствовать [2],[20].
7. В случае успешного выполнения проверки всех ЭП вернуть код «ОК». Если хотя бы одна ЭП неверная, вернуть код «INVALID».
8. Если реализаций функций СКЗИ возвращает id сертификатов проверки ЭП, то вернуть их в параметре certIds.
9. В случае неуспешного выполнения какого-либо из перечисленных выше действий, а также если число ЭП не соответствует количеству переданных типов ЭП или получены ошибки при вызове функций СКЗИ, прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.18 Метод decryptData (cryptContext, data)

Метод расшифрования входящих ЭС.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;
- data — строка, содержащая зашифрованный массив байт.

Выходные данные

- decryptedData — строка, содержащая массив байт расшифрованного ЭС.

Требования

1. Декодировать полученное ЭС по алгоритму base64 в соответствии с [11].
2. Вызвать методы СКЗИ для расшифровки полученного массива байт:
    а) алгоритмы, использующиеся для зашифрования, должны соответствовать [3],[21],[22],[23].
3. Восстановить исходный объем байт, расшифрованный на шаге 2. В случае использования ЭП для операций с ЦР используемые алгоритмы должны соответствовать документам [11].
4. В случае успешного вызова метода вернуть массив байт расшифрованного сообщения.
5. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.19 Метод decryptAndVerifyData (cryptContext, data, signature, signatureType)

Единый метод проверки ЭС.

Входные данные

- cryptContext — массив объектов, содержащий информацию о используемых криптографических провайдерах;
- data — строка, содержащая массив байт зашифрованного, закодированного, сжатого ЭС;
- signature — массив строк, содержащий массивы байт ЭП;
- signatureType — массив строк, содержащий типы подписей (возможные значения: CMScert, CMSid).

Выходные данные:

- result — строка, в случае успеха возвращает «OK», в случае неуспеха возвращает «INVALID»;
- decryptedData — строка, содержащая массив байт расшифрованного ЭС. Возвращается только в случае успешной проверки ЭП.

Требования

1. Последовательно вызвать методы:
    a) decryptData (cryptContext, data):
        i. На вход передать контекст хранилища и массив байт зашифрованного сообщения.
    b) verifySignature (cryptContext, data, signature, signatureType):
        i. На вход передать массив контекстов хранилищ, разжатое ЭС, набор ЭП, типы подписей.
2. Если при выполнении метода verifySignature получен код «INVALID», вернуть код «INVALID».
3. Если при выполнении метода verifySignature получен код «OK», вернуть код «OK» и расшифрованное разжатое ЭС в параметре decryptedData.
4. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.20 Метод signatureAndEncryptData (cryptContext, data, certId, signatureType, certIds, certFormat, certPattern)

Единый метод формирования ЭС.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;
- data — массив байт, содержащий ЭС;
- certId — строка, идентификатор сертификата подписи;
- signatureType — строка, тип подписи (возможные значения: CMScert, CMSid);
- certIds — массив байт, содержащий сертификаты получателя в формате X.509 (необязательный параметр);

- certFormat — строка формат сертификатов (необязательный параметр);
- certPattern — шаблон поиска сертификата получателя, содержащий массив пар ключ — значение. В ключе указать название поля, по которому будет осуществляться поиск, в значении — содержимое поля (необязательный параметр).

Выходные данные

- result — массив байт, содержащий зашифрованное ЭС, закодированное по алгоритму base64.
- signature — массив байт, содержащий сформированную ЭП, закодированную по алгоритму base64.

Требования

1. Необходимо последовательно вызвать методы:
    a) createSignature (cryptContext, data, certId, signatureType):
        i. На вход передать контекст хранилища, массив байт ЭС, тип подписи, идентификатор сертификата подписи.
    a) encryptData (cryptContext, data, certIds, certFormat, certPattern):
        i. На вход передать контекст хранилища, массив байт сжатого ЭС, сертификат получателя и формат сертификата либо шаблон для поиска сертификата.
2. При успешном выполнении методов вернуть:
    - массив байт, содержащий зашифрованное сообщение, закодированное по алгоритму base64;
    - массив байт, содержащий ЭП, закодированную по алгоритму base64.
3. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.21 Метод deleteStore (storeId)

Метод удаления данных сегмента хранилища СКЗИ.

Входные данные

- storeId — строка, идентификатор хранилища СКЗИ.

Выходные данные

- result — строка, в случае успеха возвращает «ОК».

Требования

1. Вызвать функции СКЗИ, отвечающие за гарантированное удаление сегмента хранилища СКЗИ и всех сохраненных в ней данных. Уничтожение ключевой информации должно осуществляться в соответствии с документом [32].
2. После успешного выполнения функций удалить данные из внутреннего хранилища.
3. Вернуть «ОК».
4. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.22 Метод deleteCert (cryptoContext, certPattern)

Метод удаления сертификатов ключей пользователя.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;
- certPattern — шаблон поиска сертификатов, содержащий массив пар ключ − значение.

Выходные данные

- result — строка, в случае успеха возвращает «ОК».

Требования

1. Вызвать функцию СКЗИ для получения сертификатов в соответствии с полученным шаблоном поиска и контекстом хранилища.
2. Вызвать функцию СКЗИ для удаления найденных сертификатов.
3. После успешного выполнения функций вернуть «ОК».
4. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.23 Метод getCert (cryptoContext, certPattern, certFormat)

Метод извлечения сертификатов ключей пользователя.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;
- certPattern — шаблон поиска сертификатов, содержащий массив пар ключ — значение;
- certFormat — формат сертификатов.

Выходные данные

- certs — массив байт, содержащий найденные сертификаты в требуемом формате.

Требования

1. Вызвать функцию СКЗИ для получения сертификатов в соответствии с полученным шаблоном поиска и контекстом хранилища.
2. Вызвать функцию СКЗИ для извлечения найденных сертификатов.
3. После успешного выполнения функций вернуть массив байт, содержащий извлеченные сертификаты, при необходимости преобразованные в нужный формат.
4. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.3.24 Метод getExpirationTimeCert (cryptoContext, certId)

Метод получения срока действия сертификата.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;
- certId — строка, идентификатор сертификата.

Выходные данные

- time — datatime, дата и время истечения срока действия найденного сертификата.

Требования

1. Вызвать функцию СКЗИ для получения времени истечения срока действия сертификата в соответствии с полученным идентификатором и контекстом хранилища.
2. После успешного выполнения функции вернуть полученную дату и время истечения срока действия сертификата.
3. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.4 **Класс GostTlsSocket**

Класс содержит методы для работы с сокетами, предназначенными для создания соединений для передачи данных, защищенных по протоколу ГОСТ TLS.

Реализация методов класса может различаться в зависимости от особенностей среды функционирования ПМ и СКЗИ входящего в состав ПМ.

3.2.4.1 Метод getGostTlsSocket (cryptoContext, clientGostTlsCertId, result)

Метод создания сокета, предназначенного для установки защищенного по протоколу ГОСТ TLS соединения.

Входные данные

- cryptContext — структура данных, содержащая информацию о криптографическом провайдере;
- clientGostTlsCertId — строка, идентификатор клиентского ГОСТ TLS сертификата. Опционально. Передается если необходимо установить двухстороннее ГОСТ TLS соединение;
- result — ссылка на строку для возврата результата исполнения метода.

Выходные данные

- В случае успеха метод возвращает ссылку на экземпляр класса GostTlsSocket.
- В случае ошибки возвращается null и код ошибки в строке result.

Требования

1. Создать экземпляр класса GostTlsSocket. Объект должен содержать:
    - cryptContext;
    - ссылку на хранилище корневых сертификатов;
    - clientGostTlsCertId (опционально, если передан во входных данных).
2. Если при создании объекта возникла ошибка, освободить все ресурсы, задействованные выполнением данного запроса, вернуть null и записать в строку result код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.4.2 Метод GostTlsSocket connect (host, port, waitForConnected)

Метод экземпляра класса GostTlsSocket. Метод используется для открытия защищенного подключения по адресу и порту, указанным в параметрах вызова метода.

Входные данные

- host — строка, host для подключения сокета;
- port — натуральное число, порт для подключения сокета;
- waitForConnected — натуральное число, время ожидания подключения сокета в мс. Передается опционально, значение по умолчанию — 30000 мс.

Выходные данные

- result — строка, в случае успеха возвращает «ОК».

Требования

1. Если для данного сокета существует не закрытое соединение, прервать обработку вызова, вернуть код ошибки «Сокет уже используется».
2. Получить IP-адрес по хосту. В случае ошибки прервать обработку вызова, вернуть код ошибки «Хост неизвестен».
3. При открытии подключения ПМ должен выполнить взаимодействие с сервером по указанным в параметрах вызова адресу и порту по протоколу handshake согласно требованиям документов [4] или [5].
4. Для выполнения проверки статуса отзыва сертификата транспортного (TLS) криптографического ключа, генерации эфемерной ключевой пары и

создания сессионного ключа ПМ должен вызвать соответствующие функции СКЗИ.

5. Если ПМ не поддерживает версию протокола ГОСТ TLS сервера, вернуть код ошибки «Не поддерживаемая версия протокола TLS».
6. Если при получении сессионного ключа возникла ошибка, вернуть код ошибки «Ошибка получения сессионного ключа».
7. В случае превышения времени ожидания подключения сокета значения входящего параметра waitForConnected вернуть код ошибки «Превышен таймаут ожидания подключения».
8. В случае неуспешного выполнения какого-либо из перечисленных выше действий или получения ошибок при вызове функций СКЗИ прервать выполнение метода и вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.4.3 Метод GostTlsSocket close (waitForDisconnected)

Метод экземпляра класса GostTlsSocket. Метод предназначен для закрытия защищенного подключения.

Входные данные

- waitForDisconnected — натуральное число, время ожидания закрытия соединения в мс. Передается опционально, значение по умолчанию — 30000 мс.

Выходные данные

- result — строка, в случае успеха возвращает «ОК».

Требования

1. Выполнить закрытие соединения в соответствии с требованиями документов [4] или [5].
2. В случае превышения времени ожидания закрытия соединения значения входящего параметра waitForDisconnected вернуть код ошибки «Превышен таймаут закрытия соединения».
3. Освободить все задействованные данным соединением ресурсы.

4. В случае возникновения ошибок при выполнении каких-либо из перечисленных выше действий прервать выполнение метода, вернуть код ошибки в соответствии с описанием в разделе 3.2.6.

3.2.4.4 Метод GostTlsSocket write (data, waitForByteWritten)

Метод экземпляра класса GostTlsSocket. Метод предназначен для передачи данных в открытое защищенное по протоколу ГОСТ TLS соединение.

Входные данные

- data — массив байт, передаваемых в сокет;
- waitForByteWritten — натуральное число, время ожидания записи в соединение минимум 1 байта в мс. Передается опционально, значение по умолчанию — 30000 мс.

Выходные данные

- result — строка, в случае успеха возвращает «ОК».

Требования

Метод должен поддерживать режим асинхронной записи.

1. Выполнить передачу массива данных data с использованием методов криптографической защиты, предоставляемых СКЗИ в составе ПМ, в соответствии с требованиями документов [4] или [5].
2. В случае превышения времени записи блока передаваемых данных параметра waitForByteWritten, вернуть код ошибки «Превышен таймаут записи данных».
3. В случае возникновения ошибок при выполнении каких-либо из перечисленных выше действий прервать выполнение метода, вернуть код ошибки в соответствии с описанием в разделе 3.2.6.
4. В случае разрыва соединения вернуть код ошибки «Соединение потеряно».

3.2.4.5 Метод GostTlsSocket read (buf, bufSize, waitForByteRead)

Метод вызывается для экземпляра класса GostTlsSocket и используется для чтения данных из защищенного ГОСТ TLS соединения.

Входные данные

- buf — буфер для данных, получаемых из защищенного ГОСТ TLS соединения.
- bufSize — натуральное число, размер буфера buf в байтах.
- waitForByteRead — натуральное число, время ожидания чтения данных из соединения в мс. Передается опционально, значение по умолчанию — 30000 мс.

Выходные данные

- result — строка, в случае успеха возвращает «ОК».

Требования

Метод должен поддерживать режим асинхронного чтения.

1. Выполнить прием блока данных в массив buf с использованием методов криптографической защиты, предоставляемых СКЗИ в составе ПМ, в соответствии с требованиями документов [4] или [5].
2. В случае превышения времени ожидания данных для чтения параметра waitForByteRead, вернуть код ошибки «Превышен таймаут ожидания данных».
3. В случае возникновения ошибок при выполнении каких-либо из перечисленных выше действий прервать выполнение метода, вернуть код ошибки в соответствии с описанием в разделе 3.2.6.
4. В случае разрыва соединения вернуть код ошибки «Соединение потеряно».

3.2.4.6 Метод GostTlsSocket getStatus ()

Метод вызывается для экземпляра класса GostTlsSocket и используется для получения сведения о его состоянии.

Выходные данные

- result — строка, возможные значения:
  - «Готов к использованию»;
  - «Соединен с <host>:<port>».

Требования

Метод должен вернуть состояние сокета на основании данных ПМ без попытки отправки запроса проверки соединения к серверу.

3.2.4.7 Метод GostTlsSocket getConnectionStatus (waitForConnected)

Метод вызывается для экземпляра класса GostTlsSocket и используется для уточнения статуса соединения сокетов в состоянии «Соединен с <host>:<port>».

Входные данные

- waitForConnected — натуральное число, время ожидания ответа от сервиса в мс. Передается опционально, значение по умолчанию — 30000 мс.

Выходные данные

- result — строка, возможные значения:
  - «Соединен»;
  - «Не соединен»;
  - «Превышен таймаут ожидания ответа от сервера»;
  - «Отсутствует техническая возможность проверки соединения».

Требования

1. В случае если метод вызывается для сокета, находящегося в состоянии «Готов к использованию», вернуть «Не соединен».
2. В случае если на основании данных ПМ сокет находится в состоянии «Соединен» и имеется техническая возможность проверки статуса соединения, метод должен инициировать проверку состояния соединения и вернуть ее результат в строке result. В случае если при проверке состояния соединения превышен лимит ожидания ответа сервера, вернуть «Превышен таймаут ожидания ответа от сервера» в строке result.
3. В случае если на основании данных ПМ сокет находится в состоянии «Соединен» и отсутствует техническая возможность проверки статуса соединения, вернуть «Отсутствует техническая возможность проверки соединения» в строке result.

Сведения, полученные в результате проверки соединения с сервером, носят информативный характер и не влияют на статус сокета. При получении ответа «Не соединен» МП должно вызвать метод GostTlsSocket close (waitForDisconnected) для

освобождения ресурсов, которые были использованы ранее открытым соединением, и (при необходимости) повторного использования сокета.

3.2.4.8 Уничтожение экземпляра класса

Класс должен содержать метод, вызываемый автоматически в момент уничтожения экземпляра класса. Данный метод должен выполнять проверку наличия и инициировать закрытие открытого ГОСТ TLS соединения.

3.2.5 **Сервисные функции**

В ПМ должен быть реализован класс, содержащий сервисные функции. Требования к функциям этого класса приведены в данном разделе.

3.2.5.1 Удаление данных из хранилища ПМ

Метод должен удалять сохраненные данные ПМ на уровне файловой системы ОС в пределах пространства, выделенного для МП.

3.2.5.2 Сбор информации о взаимодействии с ПМ

При внешнем вызове функций ПМ необходимо сохранять соответствующую информацию во внутреннее хранилище. В том числе если результат вызова функции был неуспешным.

Список данных, которые необходимо сохранять:

- id — строка, уникальный идентификатор действия;
- createdAt — Unix Timestamp, дата и время действия;
- functionName — строка, название вызванного метода;
- result — строка, результат выполнения функции (SUCCESS, FAIL);
- inputData — строка, список входных данных, может быть пустым;
- outputData — строка, список выходных данных, может быть пустым;
- storeId — строка, идентификатор хранилища СКЗИ, может быть пустым;
- errorCode — целое число, код ошибки, может быть пустым;
- errorMessage — строка, описание ошибки, может быть пустым

Значение cryptContext, входные и выходные данные для криптографических функций не сохраняются. Описание требований к хранению данных см. в пункте 3.4.

3.2.6 **Список ошибок**

В Таблице 7 приведен минимальный список исключений, при получении которых необходимо прерывать выполнение метода, возвращать код и описание ошибки.

Таблица 7 — Список ошибок

Код ошибки	Описание ошибки
1	Не удалось проверить целостность МП/ПМ
2	Не удалось успешно инициализировать ПМ
3	Не удалось успешно инициализировать СКЗИ
4	Не удалось удалить экземпляр хранилища в СКЗИ
5	Не удалось изменить пароль
6	Не удалось удалить хранилище
7	Не удалось создать хранилище
8	Не удалось выполнить вызов метода СКЗИ
9	Непредвиденная ошибка в работе СКЗИ
10	Истек срок действия сертификата пользователя
11	Не удалось обработать входящие параметры
12	Недопустимые значения входных данных
13	Не удалось сохранить сертификат пользователя
14	Не удалось сохранить CAC
15	Хранилище не найдено

Код ошибки	Описание ошибки
16	Неверный пароль
17	Время действия сессии истекло
18	Требуется инициализация ПМ
19	Хранилище заблокировано
20	Срок действия криптографического ключа истек
21	Сертификат пользователя аннулирован
22	Отсутствует пара криптографических ключей
23	Отсутствует пара транспортных криптографических ключей
24	Не удалось сформировать запрос на сертификат
25	Не удалось сформировать запрос на транспортный сертификат
26	Неверный пароль от хранилища
27	Не удалось сформировать ЭП
28	Не удалось найти сертификат пользователя
29	Не удалось проверить ЭП
30	Не удалось расшифровать ЭС
31	Не удалось зашифровать ЭС
32	Транспортный сертификат аннулирован
33	Не удалось найти сертификат для расшифрования
34	Не удалось удалить сертификат пользователя
35	Серверный сертификат недействителен
36	Истек срок действия транспортного сертификата
37	Не удалось установить соединение

Код ошибки	Описание ошибки
38	Невозможно вызвать криптографические функции
39	Истек срок действия пароля
40	Не удалось найти лог взаимодействия с ПМ за указанный период
41	Не удалось сохранить данные
42	Не удалось сформировать криптографические ключи
43	Не удалось сохранить криптографические ключи
44	Проблема с сетевым подключением
45	Неизвестный сбой
46	Целостность МП/ПМ нарушена
47	Не удалось выполнить сбор энтропии
48	Не удалось получить доступ к хранилищу корневых сертификатов
49	Не удалось найти транспортный сертификат
50	Не удалось сохранить транспортный сертификат
51	Не удалось удалить транспортный сертификат
52	Ошибка при создании сокета
53	Хост не известен
54	Значение параметра port находится вне допустимого диапазона значений
55	Превышен таймаут ожидания подключения
56	Соединение потеряно
57	Превышен таймаут закрытия соединения
58	Превышен таймаут записи данных

Код ошибки	Описание ошибки
59	Превышен таймаут ожидания данных
60	Транспортный сертификат сервера не прошел проверку подлинности
61	Не поддерживаемая версия протокола TLS
62	Ошибка получения сессионного ключа
63	Ошибка подключения к серверу
64	Ошибка записи данных в сокет
65	Ошибка чтения данных из сокета
66	Ошибка зашифрования данных перед записью в сокет
67	Ошибка дешифрования данных, считанных из сокета
68	Ошибка проверки целостности данных
69	Сокет уже используется
70	Инициализация ПМ уже выполнена

3.3 Порядок использования ПМ

3.3.1 Инициализация ПМ

Для инициализации ПМ, СКЗИ и выполнения проверки целостности файлов и ресурсов МП необходимо вызывать метод init (filesWithCheckSums). filesWithCheckSums должен содержать эталонные значения хэшей, рассчитанные от файлов и ресурсов МП, целостность которых необходимо проверить, и пути к ним. Хэши должны быть рассчитаны в соответствии с документом [30].

Рекомендуется вызывать данный метод после каждой перезагрузки приложения однократно до выполнения любых вызовов ПМ.

Список файлов и ресурсов МП, для которых будет выполнятся контроль целостности, определяется разработчиком МП.

3.3.2 **Работа с криптографическими функциями**

Для работы с криптографическими функциями:

1. МП должно вызвать метод init (filesWithCheckSums).
2. МП должно вызвать метод initCryptoContext (storeId, userId):
    a) МП должно передать уникальный идентификатор хранилища СКЗИ и уникальный идентификатор пользователя МП во входных параметрах функции ПМ;
    b) требования к формированию идентификаторов зависят от особенностей реализации СКЗИ и ПМ;
    c) при работе с хранилищем СКЗИ будет запрошена установка/ввод пароля;
    d) для каждого пользователя может быть создано несколько защищенных хранилищ с доступом по паролю. Необходимость использования дополнительного усложненного пароля определяется МП в зависимости от вида выполняемой операции;
    e) идентификаторы хранилищ СКЗИ должны быть уникальными для различных пользователей МП и различных сегментов хранилищ одного пользователя;
    f) идентификатор пользователя МП должен быть уникальным;
    g) МП определяет какой сегмент хранилища СКЗИ необходимо использовать для выполнения криптографической операции, а также формирует и хранит уникальный идентификатор сегмента хранилища;
    h) сегмент хранилища СКЗИ для хранения транспортных сертификатов и сертификатов для проверки ЭП определяется разработчиком исходя из особенностей реализации СКЗИ и особенностей ПМ.
3. В случае использования МП и ПМ новым пользователем криптографические ключи, не принадлежащие ему, должны удаляться с использованием механизмов СКЗИ по гарантированному уничтожению ключей (подробнее см. в разделе 3.2.3.22).
4. После успешного создания защищенного сегмента хранилища МП получит от СКЗИ объект cryptContext, который будет использоваться для вызова криптографических функций в рамках одной сессии пользователя.

5. Объект cryptContext формируется на стороне СКЗИ и содержит информацию о состоянии и параметрах работы с криптографическим провайдером для выполнения операций криптографии и управления криптографическими ресурсами.
6. В случае завершения сессии пользователя или аварийного завершения работы МП или ПМ МП должно вызвать метод releaseContext(cryptoContext) и деинициализировать работу с хранилищем СКЗИ. Для продолжения работы с криптографическими функциями необходимо заново инициализировать хранилище СКЗИ.

3.3.3 **Загрузка CAC**

Возможны следующие взаимодействия с CAC:

1. В ПМ передается пустой CAC:
    a) при вызове функции saveCrl на вход будет передан пустой CAC. СКЗИ должно обновить и обнулить ранее загруженный CAC. В таком случае при проверке наличия в CAC сертификат никогда не будет аннулированным.
2. CAC не передается в ПМ:
    a) функция saveCrl не вызывается. CAC загружен не будет. Проверка наличия сертификатов в CAC в таком случае не выполняется.
3. CAC передается в ПМ:
    a) передать в функции saveCrl CAC, содержащий непустой список аннулированных сертификатов.

3.3.4 **Смена пароля для доступа к хранилищу**

1. Для получения срока истечения действия пароля МП может вызывать функцию getPasswordExpirationTime (storeId, timeZone).
2. Срок действия пароля определяется политикой использования СКЗИ.
3. При приближении времени смены пароля или в случае получения кода ошибки 16 МП должно вызвать метод changePassword (storeId):
    a) в результате вызова метода запрашивается ввод старого и нового пароля. При выполнении требований к паролям устанавливается новый пароль и обновляется его срок действия.

3.3.5 Смена криптографических ключей пользователя

При получении уведомления об истечении срока действия криптографических ключей пользователя МП должно инициировать формирование новых криптографических ключей пользователя и формирование запроса на СКПЭП в соответствии с требованиями из [11].

3.3.6 Сбор ЦО

Сбор информации о системном окружении для последующего использования и формирования ЭС рекомендуется реализовать на стороне МП для снижения рисков и зависимостей в МП.

В зависимости от целей использования рекомендуется собирать ЦО одним из следующих способов:

1. Формировать ЦО в соответствии с требованиями, указанными в стандарте [12];

2. В случае использования ЦО для операций с ЦР формировать ЦО в соответствии с требованиями, описанными ниже.

Требования к сбору ЦО для использования в операциях с ЦР:

1. Собрать информацию о системном окружении. Параметры, включаемые в ЦО, зависят от ОС устройства. Обязательный список параметров зависит от ОС устройства и перечислен в Таблице 8.
2. ЦО формируется следующим образом:

    a) собранная информация об устройстве должна быть объединена в одну строку формата XML, как указано в таблице, с применением функции trim ко всем собранным значениям и названиям параметров. Названия параметров с точностью до символа должны соответствовать указанным в Таблице 8 из этой Спецификации;

b) если какой-либо параметр не удалось получить, необходимо указать его пустое значение в виде <параметр></параметр>.

3. Если для получения параметра требуется дополнительное разрешение пользователя, то МП должно его запрашивать.

Требования к передаче ЦО для операций с ЦР определены в документе [11].

Таблица 8 — Требования к параметрам ЦО

Платформа	**Параметр**	**Описание**	**Формат данных (Длина/Тип данных/Значения)**
ОС Apple iOS	DeviceModel	Производитель и модель мобильного устройства	Длина: 32 символа Тип данных: строка
	DeviceID	Идентификатор устройства	Длина: 15 символов Тип данных: строка
	OSName	Название ОС	Длина: 10 символов Тип данных: строка
	SerialNumber	Серийный номер устройства	Длина: 16 символов Тип данных: строка
	IdentifierForVendor	Буквенно-цифровая строка, которая однозначно идентифицирует устройство для разработчиков приложений	Длина: 16 символов Тип данных: строка
ОС Google Android	DeviceModel	Производитель и модель мобильного устройства	Длина: 32 символа Тип данных: строка
	DeviceID	Идентификатор устройства	Длина: 15 символов Тип данных: строка
	OSName	Название ОС	Длина: 10 символов Тип данных: строка
	SerialNumber	Серийный номер устройства	Длина: 16 символов Тип данных: строка

Платформа	Параметр	Описание	Формат данных (Длина/Тип данных/Значения)
	Build.FINGERPRINT	Строка, которая однозначно идентифицирует сборку	Длина: 100 символов Тип данных: строка
	Build.MANUFACTURER	Производитель сборки	Длина: 16 символов Тип данных: строка
	Build.BOOTLOADER	Номер версии системного загрузчика	Длина: 16 символов Тип данных: строка
ОС Аврора	DeviceModel	Производитель и модель мобильного устройства	Длина: 32 символа Тип данных: строка
	DeviceID	Идентификатор устройства	Длина: 15 символов Тип данных: строка
	OSName	Название ОС	Длина: 10 символов Тип данных: строка
	SerialNumber	Серийный номер устройства	Длина: 16 символов Тип данных: строка
ОС HarmonyOS	DeviceModel	Производитель и модель мобильного устройства	Длина: 32 символа Тип данных: строка
	DeviceID	Идентификатор устройства	Длина: 15 символов Тип данных: строка
	OSName	Название ОС	Длина: 10 символов Тип данных: строка
	SerialNumber	Серийный номер устройства	Длина: 16 символов Тип данных: строка

3.3.7 **Установка ГОСТ TLS соединения**

При установке ГОСТ TLS соединения МП должно:

1. Создать экземпляр класса GostTlsSocket вызовом метода ПМ getGostTlsSocket. На вход метода передать cryptoContext и идентификатор

клиентского ГОСТ TLS сертификата, если требуется установить двухстороннее ГОСТ TLS соединение.

2. Вызвать метод ПМ connect для созданного экземпляра класса GostTlsSocket. В параметрах вызова передать адрес и порт подключения. После успешного выполнения данного метода будет открыто защищенное ГОСТ TLS соединение.

3. МП, используя методы сокета write и read, может передавать и считывать данные из защищенного ГОСТ TLS-соединения.

4. Для проверки статуса соединения сокетов МП может использовать метод getConnectionStatus.

5. После завершения использования соединения оно должно быть закрыто. Для этого МП может вызвать метод GostTlsSocket close (waitForDisconnected). При выполнении данного метода все ресурсы будут освобождены.

6. Контроль необходимости открытия/закрытия соединения должен выполняться на стороне МП. Установка нового TCP-соединения для каждого внешнего вызова не является обязательной. Чтобы создать соединение с другим хостом/портом при открытом соединении, необходимо создать новый экземпляр класса GostTlsSocket.

3.4 Требования к логическому хранению данных

3.4.1 **Требования к составу, структуре и способам организации данных в системе**

В ПМ должно быть реализовано внутреннее хранилище для обеспечения корректной работы его функций.

Требования к хранилищу СКЗИ описаны в пункте 3.4.4.

Внутреннее хранилище должно быть расположено в приватной области данных, недоступной для других приложений.

Во внутреннем хранилище ПМ должна сохраняться следующая информация:

- Блокировка доступа к хранилищу:
  - primary key storeId — строка, идентификатор хранилища СКЗИ;
  - lockEnd — время окончания блокировки доступа в формате Unix Timestamp.
- Учет пользователей
  - primary key id — строка, уникальный идентификатор;
  - userId — строка, идентификатор пользователя МП;
  - storeId — строка, идентификатор хранилища СКЗИ;
  - createdAt — дата и время сохранения информации в формате Unix Timestamp;
  - isDeleted — флаг, отображающий факт удаления ключевой информации пользователя
- Учет неудачных попыток ввода пароля:
  - primary key storeId — строка, идентификатор хранилища СКЗИ;
  - целое число count — целое число, количество неуспешных попыток ввода пароля (подряд).
- Журнал действий:
  - primary key id — строка, уникальный идентификатор действия;
  - createdAt — дата и время действия в формате Unix Timestamp;
  - functionName — строка, название вызванного метода;
  - result — строка, результат выполнения функции (SUCCESS, FAIL);
  - inputData — строка, список входных данных (может быть пустым);
  - outputData — строка, список выходных данных (может быть пустым);
  - storeId — строка, идентификатор хранилища СКЗИ (может быть пустым);
  - errorCode — целое число, код ошибки (может быть пустым);
  - errorMessage — строка, описание ошибки (может быть пустым).

3.4.2 **Схема хранения ключевой информации**

На Рисунке 3 изображена схема хранения ключевой информации при использовании ПМ для подключения к ПлЦР. Описание компонентов схемы приведено в Таблице 9.

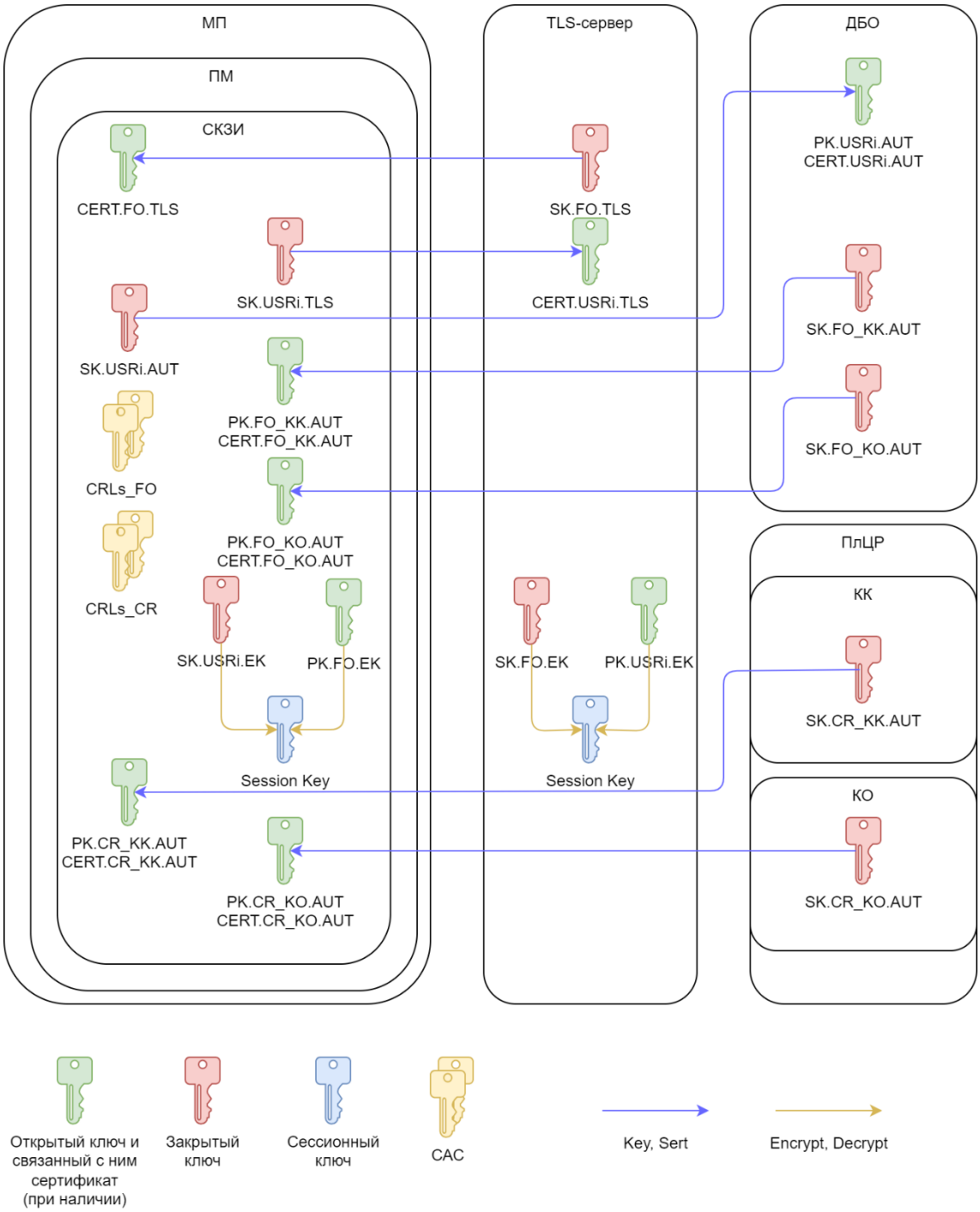

Рисунок 3 — Схема хранения ключевой информации

Таблица 9 — Описание схемы хранения ключевой информации

Ключ/Сертификат	Алгоритм	Длина ключа	Назначение	Жизненный цикл	Хранение и передача
SK.FO.TLS CERT.FO.TLS	ГОСТ Р 34.10-2012/34.10-2018	Закрытый ключ – 256 бит Открытый ключ – 512 бит	Сертификат и соответствующий ему закрытый ключ для аутентификации сервера финансовой организации при установлении двухстороннего TLS-соединения между МП и сервером финансовой организации. Используется при установлении одностороннего TLS-соединения с сервером финансовой организации для передачи запроса на первичную выдачу клиентского ГОСТ-TLS сертификата и при последующем установлении двустороннего TLS-соединения [29].	Серверный сертификат безопасности ГОСТ-TLS выпускается УЦ Безопасности финансовой организации.	Серверный TLS-сертификат и связанная с ним цепочка сертификатов, включающая все промежуточные сертификаты. Корневой (самоподписанный) сертификат УЦ Безопасности, должен быть включен в ресурсы МП. При смене сертификатов требуется обновление МП.
SK.USRi.TLS CERT.USRi.TLS	ГОСТ Р 34.10-2012/34.10-2018	Закрытый ключ – 256 бит	Клиентский сертификат и соответствующий ему закрытый ключ для аутентификации МП при	Ключи для двухстороннего ГОСТ-TLS	Закрытый ключ хранится в сегменте хранилища СКЗИ,

Ключ/Сертификат	Алгоритм	Длина ключа	Назначение	Жизненный цикл	Хранение и передача
		Открытый ключ – 512 бит	установлении двухстороннего TLS-соединения между МП и сервером финансовой организации.	формируются СКЗИ по запросу МП. Первичный запрос на выпуск транспортного сертификата в формате PKCS#10 формируется в ПМ по запросу МП. Для перевыпуска транспортного сертификата создается новая пара транспортных ключей, формируется запрос на выпуск транспортного сертификата в формате PKCS#10. Далее МП формирует сообщение в формате PKCS#7, содержащее данный запрос, подписывает его закрытым транспортным ключом и направляет на сервер финансовой организации.	соответствующем пользователю приложения. Сертификат безопасности ГОСТ-TLS выпускается УЦ Безопасности финансовой организации по запросу от МП.

Ключ/Сертификат	Алгоритм	Длина ключа	Назначение	Жизненный цикл	Хранение и передача
				Транспортный сертификат после выпуска направляется в TLS-криптошлюз.	
SK.FO.EK PK.FO.EK	ГОСТ Р 34.10-2012/34.10-2018	Закрытый ключ – 256 бит Открытый ключ – 512 бит	Эфемерная ключевая пара, предназначенная для обмена сессионными ключами.	Ключи формируются на сервере финансовой организации в процессе установления TLS-соединения.	Открытый эфемерный ключ передается в клиентское приложение в процессе установления двухстороннего TLS-соединения между МП и сервером финансовой организации. Ключи, использованные для получения сессионных ключей, не хранятся в СКЗИ.

Ключ/Сертификат	Алгоритм	Длина ключа	Назначение	Жизненный цикл	Хранение и передача
SK.USRi.EK PK.USRi.EK	ГОСТ Р 34.10-2012/34.10-2018	Закрытый ключ – 256 бит Открытый ключ – 512 бит	Эфемерная ключевая пара, предназначенная для обмена сессионными ключами.	Ключи формируются СКЗИ в процессе установления TLS-соединения.	Открытый эфемерный ключ передается на сервер финансовой организации в процессе установления двухстороннего TLS-соединения между МП и сервером финансовой организации. Ключи, использованные для получения сессионных ключей, не хранятся в СКЗИ.
Session Key	ГОСТ Р 34.12-2015/34.12-2018	Закрытый ключ – 256 бит Открытый ключ – 512 бит	Сессионный ключ. Предназначен для зашифрования/расшифрования данных в SDK и на сервере авторизации.	Ключ вычисляется из пары ключей с использованием алгоритма ECDH-ES при необходимости и не хранится:	Не передается по каналам связи.

Ключ/Сертификат	Алгоритм	Длина ключа	Назначение	Жизненный цикл	Хранение и передача
				В мобильном приложении - SK.USRi.EK + PK.FO.EK На сервере финансовой организации - SK.FO.EK + PK.USRi.EK	
SK.USRi.AUT PK.USRi.AUT CERT.USRi.AUT	ГОСТ Р 34.10-2012/34.10-2018	Закрытый ключ – 256 бит Открытый ключ – 512 бит	Ключевая пара и сертификат пользователя МП (пользователя ПлЦР).	Ключевая пара создается СКЗИ по запросу пользователя МП. Запрос на выпуск сертификата пользователя МП (пользователя ПлЦР) формируется в ПМ по запросу МП. Запрос направляется в ПУЦ УНЭП финансовой организации, который осуществляет его выпуск (см. [28]).	Закрытый ключ хранится в сегменте хранилища СКЗИ, соответствующем пользователю приложения. Сертификат выпускается ПУЦ УНЭП финансовой организации и сохраняется на ее сервере. В случае, если финансовая организация

Ключ/Сертификат	Алгоритм	Длина ключа	Назначение	Жизненный цикл	Хранение и передача
				Для перевыпуска сертификата МП инициирует создание новой ключевой пары и формирование запроса на выпуск сертификата в формате PKCS#10. МП формирует сообщение в формате PKCS#7, содержащее данный запрос, подписывает его закрытым ключом пользователя МП (пользователя ПлЦР) и направляет на сервер финансовой организации.	является участником ПлЦР, данный сертификат также сохраняется на сервере оператора ПлЦР в привязке к пользователю ПлЦР и его счету.
CRLs_CR	ГОСТ Р 34.10-2012/34.10-2018		Списки аннулированных сертификатов БР (CAC БР). Включают в себя список аннулированных сертификатов проверки ЭП.	Списки аннулированных сертификатов загружаются в СКЗИ по запросу МП. В МП должен быть предусмотрен	

Ключ/Сертификат	Алгоритм	Длина ключа	Назначение	Жизненный цикл	Хранение и передача
				механизм обновления в СКЗИ CRLs_CR.	
CRLs_FO	ГОСТ Р 34.10-2012/34.10-2018		Списки аннулированных сертификатов финансовой организации. Включают в себя список аннулированных сертификатов проверки ЭП и список аннулированных ГОСТ-TLS сертификатов.	Списки аннулированных сертификатов загружаются в СКЗИ по запросу МП. В МП должен быть предусмотрен механизм обновления в СКЗИ CRLs_FO.	
Ключи и сертификаты, необходимые для выполнения операций с ЦР					
SK.FO_KK.AUT PK.FO_KK.AUT CERT.FO_KK.AUT	ГОСТ Р 34.10-2012/34.10-2018	Закрытый ключ – 256 бит Открытый ключ – 512 бит	Ключевая пара и сертификат КК участника ПлЦР.	Сертификат выпускается УЦ БР. Доставляется в МП и устанавливается в СКЗИ посредством вызова МП функции ПМ в сценарии "Регистрация сертификата ключа Клиента-ФЛ для доступа к кошельку на ПлЦР" [11].	Закрытый ключ хранится в КК участника ПлЦР. Сертификат КК участника ПлЦР и связанная с ним цепочка сертификатов, включающая все промежуточные сертификаты и сертификат ПУЦ УНЭП, должны

Ключ/Сертификат	Алгоритм	Длина ключа	Назначение	Жизненный цикл	Хранение и передача
				В МП должен быть предусмотрен механизм доставки и установки в СКЗИ сертификата участника ПлЦР при смене сертификата в случае компрометации ключа SK.FO_KK.AUT или по истечении его срока действия.	доставляться в МП и устанавливаться в СКЗИ путем вызова соответствующей функции ПМ. Сертификат УЦ БР должен быть включен в ресурсы МП. При смене сертификата корневого УЦ БР требуется обновление МП.
SK.FO_KO.AUT PK.FO_KO.AUT CERT.FO_KO.AUT	ГОСТ Р 34.10-2012/34.10-2018	Закрытый ключ – 256 бит Открытый ключ – 512 бит	Ключевая пара и сертификат КО участника ПлЦР.	Сертификат выпускается УЦ БР. Доставляется в МП и устанавливается в СКЗИ посредством вызова МП функции ПМ в сценарии "Регистрация сертификата ключа Клиента-ФЛ для	Закрытый ключ хранится в КО участника ПлЦР. Сертификат КО участника ПлЦР и связанная с ним цепочка сертификатов, включающая все промежуточные сертификаты ПУЦ УНЭП, должны

Ключ/Сертификат	Алгоритм	Длина ключа	Назначение	Жизненный цикл	Хранение и передача
				доступа к кошельку на ПлЦР" [11]. В МП должен быть предусмотрен механизм доставки и установки в СКЗИ сертификата участника ПлЦР при смене сертификата в случае компрометации ключа SK.FO_KO.AUT или по истечении его срока действия.	доставляться в МП и устанавливаться в СКЗИ путем вызова соответствующей функции ПМ Сертификат УЦ БР должен быть включен в ресурсы МП. При смене сертификата УЦ БР требуется обновление МП.
SK.CR_KO.AUT PK.CR_KO.AUT CERT.CR_KO.AUT	ГОСТ Р 34.10-2012/34.10-2018	Закрытый ключ – 256 бит Открытый ключ – 512 бит	Ключевая пара и сертификат КО Узла РОРД.	Сертификат выпускается УЦ БР. Доставляется в МП и устанавливается в СКЗИ посредством вызова МП функции ПМ в сценарии "Регистрация сертификата ключа Клиента-ФЛ для доступа к кошельку на ПлЦР" [11].	Закрытый ключ хранится в КО Узла РОРД. Сертификат КО Узла РОРД и связанная с ним цепочка сертификатов, включающая все промежуточные сертификаты, должны доставляться в МП

Ключ/Сертификат	Алгоритм	Длина ключа	Назначение	Жизненный цикл	Хранение и передача
				В МП должен быть предусмотрен механизм доставки и установки в СКЗИ сертификата КО Узла РОРД при смене сертификата в случае компрометации ключа SK.CR_KO.AUT или по истечении его срока действия.	и устанавливаться в СКЗИ путем вызова соответствующей функции ПМ. Сертификат УЦ БР должен быть включен в ресурсы МП. При смене сертификата УЦ БР требуется обновление МП.
SK.CR_КК.AUT PK.CR_КК.AUT CERT.CR_КК.AUT	ГОСТ Р 34.10-2012/34.10-2018	Закрытый ключ – 256 бит Открытый ключ – 512 бит	Ключевая пара и сертификат КК ПлЦР.	Сертификат выпускается УЦ БР. Доставляется в МП и устанавливается в СКЗИ посредством вызова МП функции ПМ в сценарии «Регистрация сертификата ключа Клиента-ФЛ для доступа к кошельку на ПлЦР» [11].	Закрытый ключ хранится в КК ПлЦР. Сертификат КК ПлЦР и связанная с ним цепочка сертификатов, включающая все промежуточные сертификаты, должны доставляться в МП и устанавливаться в СКЗИ путем

Ключ/Сертификат	Алгоритм	Длина ключа	Назначение	Жизненный цикл	Хранение и передача
				В МП должен быть предусмотрен механизм доставки и установки в СКЗИ сертификата КК ПлЦР при смене сертификата в случае компрометации ключа SK.CR_KK.AUT или по истечении его срока действия.	вызова соответствующей функции ПМ. Сертификат УЦ БР должен быть включен в ресурсы МП. При смене сертификата УЦ БР требуется обновление МП.

3.4.3 **Требования к контролю, хранению, обновлению и восстановлению данных**

Необходимо обеспечить бесперебойное функционирование ПМ.

В случае аварийного завершения работы или при возникновении сбоев в аппаратном обеспечении ПМ должен обеспечивать сохранность данных, за исключением случаев повреждения носителей информации.

Специальных требований к сроку хранения данных не предъявляется.

3.4.4 **Требования к программному обеспечению**

3.4.4.1 Общие требования

ПМ должен иметь в своем составе сертифицированное СКЗИ не ниже класса КС1, или же сам являться сертифицированным СКЗИ не ниже класса КС1. Рекомендуется выбирать СКЗИ, разрешенные к использованию на территории Российской Федерации и за ее пределами.

СКЗИ должно быть разрешено для встраивания в ИС.

СКЗИ должно обладать пакетом документации, достаточным для встраивания в ПМ.

Операционная система, для которой предназначено СКЗИ, должна соответствовать операционной системе, в которой будет функционировать МП.

Производитель СКЗИ должен гарантировать работу СКЗИ во всех средах функционирования ПМ.

СКЗИ должно предоставлять программный интерфейс для вызова криптографических функций из ПМ.

СКЗИ не должно возвращать результаты работы методов напрямую пользователю МП.

3.4.4.2 Требования к функциям защиты информации

СКЗИ должно обеспечивать выполнение таких функций защиты информации, как:

- создание ключей ЭП и ключей проверки ЭП в соответствии с [2];
- хранение криптографических ключей, СКПЭП, транспортных криптографических ключей и сертификатов транспортных криптографических ключей в защищенном хранилище в мобильном приложении;
- формирование и проверка ЭП согласно [2], [20];
- зашифрование и расшифрование ЭС. Алгоритмы, используемые при зашифровании: [3], [21], [22], [23];
- установление защищенного соединения между приложением и сервером с использованием протокола одностороннего и двустороннего TLS в соответствии с [4], [5].

СКЗИ должно соответствовать требованиям, предъявляемым к средствам ЭП в документе [35].

СКЗИ должно иметь в составе ПДСЧ, соответствующий требованиям документа [24].

ПДСЧ для выработки инициализирующей последовательности должен использовать БДСЧ.

ПДСЧ должен поддерживать возможность параллельной работы в независимых сессиях.

СКЗИ должен предоставлять функции по адаптации дизайна экрана работы с БДСЧ (при наличии). Реализация работы БДСЧ и возможность адаптации дизайна экрана работы с БДСЧ должны соответствовать эксплуатационной документации на СКЗИ.

В СКЗИ должна быть реализована функция проверки статуса отзыва сертификата транспортного (TLS) криптографического ключа с использованием CAC и/или

OCSP, а также функция проверки статуса отзыва сертификата ЭП с использованием CAC.

Режим проверки статуса отзыва сертификата транспортного (TLS) криптографического ключа с использованием OCSP должен быть настраиваемым в конфигурации СКЗИ.

3.4.4.3 Требования к хранению и доступу к ключевой информации

СКЗИ должно обеспечивать защищенное хранение криптографических ключей пользователя. Доступ к криптографическим операциям СКЗИ должен осуществляться по паролю. Закрытые криптографические ключи должны быть неизвлекаемыми.

СКЗИ должно осуществлять управление паролями пользователей приложения для доступа к криптографическим функциям. В СКЗИ должны быть доступны функции установки и изменения пароля хранилища для сегмента хранилища СКЗИ. Требования к паролю зависят от ограничений, накладываемых СКЗИ.

Пароли пользователей должны быть неизвлекаемыми из СКЗИ. Вызов криптографических функций СКЗИ должен выполняться с указанием идентификатора сегмента хранилища СКЗИ. Интерфейс ввода пароля от хранилища должно предоставлять СКЗИ.

СКЗИ должно осуществлять контроль срока действия пароля и осуществлять его замену по запросу.

СКЗИ должно предоставлять возможность работы в рамках сессии пользователя без дополнительного запроса пароля.

Если указан неверный пароль, то ответ на вызов любого метода СКЗИ для данного пользователя должен содержать соответствующее исключение.

СКЗИ должно поддерживать возможность одновременного хранения на мобильном устройстве криптографических ключей и СКПЭП разных Пользователей МП, транспортных криптографических ключей и сертификатов транспортных криптографических ключей разных Пользователей МП. Данная

возможность должна быть реализована в СКЗИ за счет сегментации хранилища ключевой информации СКЗИ.

В СКЗИ должны быть реализованы функции удаления сегмента хранилища и гарантированного уничтожения ключевой информации по идентификатору хранилища СКЗИ в соответствии с документом [32].

СКЗИ должно предоставлять функции для установки, удаления и получения сертификатов ключей пользователя, CAC, а также извлечения времени истечения действия сертификата.

При сохранении CAC в хранилище СКЗИ должно проверять номер CAC в разрезе его серии, а не сквозную нумерацию. Проверка должна предотвращать загрузку устаревших данных.

СКЗИ должно поддерживать возможность хранения корневых сертификатов в едином хранилище для всех пользователей.

3.4.5 **Требования к форматам ЭП**

СКЗИ должно поддерживать различные форматы ЭП:

- CMScert (PKCS#7, содержит сертификат открытого ключа, соответствующего закрытому ключу, на котором произведено формирование ЭП);
- CMSid (PKCS#7, содержит идентификатор открытого ключа, соответствующего закрытому ключу, на котором произведено формирование ЭП);
- RAW (не содержит сертификата или идентификатора ключа, формат записи последовательности байт подписи — little-endian).

3.4.6 **Требования к встраиванию СКЗИ**

Встраивание СКЗИ в ПМ должно проводится в соответствии с требованиями и рекомендациями, изложенными в эксплуатационной документации на СКЗИ.

При встраивании должен быть проведен контроль возвращаемых значений при обращениях к функциям СКЗИ.

Встраивание не должно влиять на инженерно-криптографические свойства СКЗИ.

МП должно использовать только документированные функции ОС и СКЗИ, использование которых при разработке возможно без построения новых СКЗИ.

Если ПМ не является СКЗИ, то методы СКЗИ не должны быть доступны для вызова из МП.

ПМ должен предоставлять механизм поэкземплярного учета (регистрации) СКЗИ, учитывающий технологию распространения МП через магазины приложений и/или официальный сайт финансовой организации или Банка России. Работоспособность должна быть гарантирована с магазинами приложений Google Play, App Store, RuStore, NashStore, AppGallery.

Действия по регистрации СКЗИ должны осуществляться посредством использования технических средств и механизмов ПМ, в том числе с использованием механизмов СКЗИ при их наличии. При регистрации СКЗИ не должны быть задействованы сторонние сервисы.

4 Нефункциональные требования

4.1 Требования к производительности ПМ

Рекомендуется обеспечить следующие показатели назначения:

- время выполнения каждой криптографической операции ([FN-1.4] — [FN-1.9], [FN-1.12] — [FN-1.16]) должно составлять не более 0,3 секунды;
- время выполнения каждого единого вызова ([FN-1.10], [FN-1.11], [FN-1.17], [FN-1.18]) должно составлять не более 0,5 секунды;
- время инициализации ядра ПМ ([FN-1.1]) на мобильных устройствах, соответствующих требованиям раздела 2.2, должно составлять не более 3 секунд.

Показатели назначения указаны для данных размером до 1 Мб.

4.2 Содержание, объем и организация работ по созданию ПМ

4.2.1 **Общие требования**

Содержание и порядок выполнения работ по созданию ПМ рекомендуется выполнять в соответствии с стандартом [17].

Если ПМ является сертифицированным СКЗИ, то работы по его созданию и вводу эксплуатацию должны быть организованы в соответствии с требованиями документов [9] и [24].

Если ПМ имеет в своем составе сертифицированное СКЗИ, работы по созданию и вводу в эксплуатацию ПМ должны быть организованы в соответствии с его правилами пользования (в соответствии с требованиями к эксплуатации СКЗИ документа [9]). Если в правилах пользования СКЗИ, входящего в состав ПМ, не указано иное, ПМ должен пройти исследования по оценке его влияния на СКЗИ (см. раздел 4.2.3 Спецификации).

Также при организации работ по созданию ПМ рекомендуется учесть требование о внесении ПМ в Реестр (см. раздел 4.2.4 Спецификации).

4.2.2 **Требования к разработчику ПМ**

Разработчик ПМ должен соответствовать всем требованиям, предъявляемым к производителю ПО для внесения его в Реестр [14].

Разработчик ПМ должен иметь лицензию ФСБ России на разработку, производство, распространение шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств, выполнение работ, оказание услуг в области шифрования информации, техническое обслуживание 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) на виды работ из Приложения к Положению [18], в части пунктов 1[6], 2, 3.

4.2.3 **Работы по подготовке и проведению оценки влияния ПМ на выполнение требований, предъявленных к СКЗИ**

Необходимость проведения оценки влияния ПМ на СКЗИ определяется в документе [9] и эксплуатационной документации СКЗИ.

В разделе приведена информация о типовом подходе при проведении оценки влияния ПМ на выполнение требований, предъявленных к СКЗИ.

В эксплуатационной документации на ПМ по результатам оценки влияния среды функционирования на выполнение требований, предъявленных к СКЗИ, должен

[6] В случаях, когда ПМ является СКЗИ.

быть отражен факт использования данной спецификации, если ПМ разработан в соответствии с ней.

Оценка влияния может быть выполнена в один или два этапа. При выполнении оценки влияния в один этап исследование проводится после встраивания ПМ в МП. Оценка в два этапа проводится после встраивания СКЗИ в ПМ и после встраивания ПМ в МП.

Исследования по оценке влияния проводятся в соответствии с документом «Техническое задание на проведение исследований по оценке влияния специального программного обеспечения Программного модуля» (далее — ТЗ).

Разработку ТЗ и проведение исследований ПМ должна выполнять организация, имеющая действующий аттестат аккредитации испытательной лаборатории ФСБ России или допущенная ФСБ России на проведение таких работ (далее - Лаборатория).

Исследования рекомендуется проводить на основании договора между Лабораторией и Разработчиком ПМ.

Для проведения исследований Разработчик ПМ предоставляет материалы в соответствии с требованиями ТЗ. Список материалов для обеспечения исследований может включать:

- исходные тексты исследуемого ПМ, МП (передаются при встраивании ПМ в МП);
- документы разработчика ПМ (МП), описывающие архитектуру ПМ (МП), модульный состав и назначение модулей, среду исполнения;
- описания и комментированные листинги исходных текстов доступа ПМ к функциям встраиваемого СКЗИ. В случае, если ПМ является СКЗИ, передаются исходные тексты доступа МП к функциям СКЗИ;
- спецификации вызовов ПМ (описание аргументов и форматов вызовов);
- описание процедуры компиляции исходного кода ПМ;

- описание внешних вызовов (вызовы операционной среды, аппаратных средств, других программ и т.д.);
- эксплуатационную документацию;
- дистрибутивы ПМ, МП (передаются при встраивании ПМ в МП).

Приведенный список не является исчерпывающим. Лаборатория, проводящая исследования, может запросить дополнительные материалы в соответствии с ТЗ.

Разработчик ПМ должен описать трассы вызовов ПМ — СКЗИ в соответствии с документом [34] для использования данной информации при проведении оценки корректности встраивания СКЗИ в ПМ, а также ПМ в МП.

Для проведения исследований Разработчик предоставляет Лаборатории экземпляры МП со встроенным ПМ. МП предоставляются для каждой ОС, в которой функционирует ПМ.

Для ПМ, функционирующего в ОС iOS, МП предоставляется через магазин приложений App Store.

Для ПМ, функционирующих в ОС Android, ОС HarmonyOS и ОС Аврора, МП предоставляются через магазины приложений Google Play, RuStore, NashStore, AppGallery или в виде rpm-файлов для ОС Аврора или apk-файлов для ОС Android и HarmonyOS с инструкцией по скачиванию и установке на устройство.

По результатам исследований в документацию к ПМ могут быть включены требования по защите с использованием организационных мер.

Экспертиза результатов исследований ПМ осуществляется ФСБ России.

4.2.4 **Работы по внесению ПМ в Реестр**

Для внесения ПМ в Реестр при его проектировании и разработке должны быть учтены требования нормативных документов [14] и [15].

Разработчик должен соответствовать критериям, указанным в разделе 4.2.2 настоящей Спецификации.

ПМ должен сопровождаться комплектом документации в соответствии с разделом 4.3.3 настоящей Спецификации.

Поскольку ПМ имеет в своем составе сертифицированное СКЗИ, для подачи заявки на включение ПМ в Реестр необходимо положительное решение ФСБ России об оценке влияния (если в правилах пользования СКЗИ не указано иное).

Для регистрации ПМ в Реестре должны быть подготовлены и доступны экспертам демонстрационные версии МП со встроенным ПМ. Для ПМ, функционирующего в OS iOS, могут быть представлены исходные тексты проекта и их описание.

4.3 Требования к программной документации

4.3.1 Общие требования к составу технической документации

Разработчик ПМ должен предоставить комплект документации, достаточный для прохождения процедур по оценке влияния ПМ на СКЗИ, а также для включения ПМ в Реестр.

Если ПМ является СКЗИ, то состав и содержание предоставляемой документации должны соответствовать стандартному комплекту документов для СКЗИ в соответствии с [9].

4.3.2 Требования к документации, предоставляемой для прохождения оценки влияния

Требования к составу и содержанию технической документации, предоставляемой в Лабораторию для проведения оценки влияния, перечислены в Таблице 10.

Таблица 10 — Документация для проведения оценки влияния

№	Наименование документа	Требования к содержанию
Эксплуатационная документация		
1	Руководство разработчика	Руководство должно содержать: • описание функциональности ПМ; • требования к среде функционирования ПМ; • требования по встраиванию ПМ в МП:

№	Наименование документа	Требования к содержанию
		○ описание сценариев использования ПМ; ○ требования к разработчику ПМ; • описание интерфейсов ПМ: форматы вызовов, входные и выходные данные.
2	Руководство администратора	Руководство должно содержать: • назначение ПМ; • требования к среде функционирования ПМ; • сведения о встроенном СКЗИ; • правила обращения с ключевыми носителями; • описание конфигурации ПМ; • инструкции по сборке ПМ; • рекомендации по защите от НСД.
Проектная документация		
1	Документация разработчика ПМ	Документация должна содержать: • документ, описывающий структуру ПМ, модульный состав и назначение модулей, среду исполнения; • описания и комментированные листинги исходных текстов доступа ПМ к функциям встраиваемого СКЗИ; • описание процедуры компиляции исходного кода; • спецификацию вызовов: описание аргументов и форматов вызовов.
Программная документация		
1	Исходные тексты ПМ (МП – предоставляются в случае проведения оценки влияния ПМ в составе МП)	Исходные тексты ПМ для каждого языка программирования.

Указанная в Таблице 10 документация предоставляется Лаборатории, проводящей исследования по оценке влияния ПМ на СКЗИ.

Эксперты Лаборатории проводят анализ полноты и корректности эксплуатационной документации на исследуемый ПМ в части, определяющей

порядок использования СКЗИ на соответствие требованиям и рекомендациям, изложенным в эксплуатационной документации на СКЗИ.

Перечень документов, указанных в Таблице 10, не является исчерпывающим. Лаборатория, проводящая исследование, может запросить дополнительную информацию для уточнения данных.

4.3.3 Требования к документации, предоставляемой для включения ПМ в Реестр

Требования к составу и содержанию технической документации, предоставляемой при оформлении заявки на внесение ПМ в Реестр, перечислены в Таблице 11.

Таблица 11 — Документация для включения ПМ в Реестр

№	Наименование документа	Требования к содержанию
1	Руководство разработчика	Руководство должно содержать: • описание функциональности ПМ; • требования к среде функционирования ПМ; • требования по встраиванию ПМ в МП: • описание сценариев использования ПМ; • требования к разработчику ПМ; • описание интерфейсов ПМ: форматы вызовов, входные и выходные данные.
2	Инструкция по установке ПМ	Инструкция по установке полнофункционального или демонстрационного экземпляра МП, содержащего ПМ и предназначенного для экспертной проверки. Инструкция должна содержать: • системные требования к мобильному устройству для экспертной проверки; • для приложения для ОС Android, ОС HarmonyOS и ОС Аврора — доступную

№	Наименование документа	Требования к содержанию
		ссылку для скачивания файла с приложением и инструкцию по его установке; • для приложений для ОС iOS — исходные тексты проекта и его описание; • контакты технических специалистов.
3	Описание функциональных характеристик ПМ	Документ должен содержать описание функциональных характеристик экземпляра ПМ, предоставленного для экспертной проверки.
4	Описание жизненного цикла ПМ	Документ должен содержать описание следующих процессов, обеспечивающих поддержание жизненного цикла ПМ: • проектирование, • разработка, • тестирование, • приобретение, • поставка, • эксплуатация, • документирование, • поддержка версий ПМ и доработка, • устранение сбойных ситуаций. Документ должен содержать полную информацию о персонале организации, осуществляющей доработки и администрирование ПМ, его техническую поддержку, а именно: количество сотрудников и их квалификацию. При описании требований к персоналу необходимо учитывать, что ПМ содержит в своем составе СКЗИ и при его эксплуатации необходимо учитывать требования п. V документа [9]. В документ должна быть включена информация о фактических адресах, имеющих отношение к работе над ПМ, в том числе адрес размещения разработчиков, адрес размещения службы технической поддержки и т. п. Также документ должен содержать регламент службы технической поддержки: описание

№	Наименование документа	Требования к содержанию
		процесса обработки заявок, каналы подачи заявок, режим работы и т. п.
5	Описание технических средств хранения и компиляции исходного кода	Предоставляется в виде письма на бланке с печатью и подписью руководителя. Письмо должно содержать информацию о языках программирования, на которых разработан код ПМ, технических средствах хранения исходного текста ПМ, инструментах компиляции, адресах нахождения исходного кода и средств компиляции. К письму должен прикладываться документ, подтверждающий, что данные находятся по указанному адресу. Например, договор аренды помещения в случае размещения серверов с данными в офисном помещении или договор с облачным сервисом, если для хранения данных используется облако.

Требования к составу и содержанию документации составлены на основе документа [16] и могут изменяться в соответствии с его актуальной версией.

Указанная документация прикладывается к заявлению о включении сведений о ПО в Реестр и согласуется Экспертным советом Министерства цифрового развития, связи и массовых коммуникаций РФ.

4.3.4 Требования к порядку актуализации документации ПМ

В документацию ПМ могут вноситься изменения по следующим причинам:

- устранение обнаруженных ошибок в ПМ и документации ПМ;
- развитие и усовершенствование ПМ;
- инициатива владельца ПМ.

Любые изменения в документе, которые влекут за собой какие-либо изменения в других документах, должны сопровождаться внесением соответствующих изменений в другие документы.

Порядок обновления и необходимость повторной экспертизы ФСБ проектных и/или эксплуатационных документов ПМ должны быть определены в отчетных документах по результатам оценки влияния.

В случае внесения изменений в документы, предоставленные при регистрации ПМ в Реестре, необходимо сформировать Заявление об изменении сведений о программном обеспечении. К заявлению должны быть приложены измененные документы ПМ.

4.4 Порядок контроля и приемки ПМ

4.4.1 Виды, состав, объем и методы испытаний ПМ

Для ПМ рекомендуется проводить следующие виды испытаний:

- предварительные испытания;
- приемочные испытания.

Предварительные и приемочные испытания ПМ должны быть организованы и проведены в соответствии с [19].

Объем и методы предварительных и приемочных испытаний определяются документом «Программа и методика испытаний».

Испытания должны проводиться с помощью демонстрационного или полнофункционального МП, содержащего в составе реализацию ПМ, подлежащую испытаниям. Специальные требования к функциональности демонстрационного приложения в данной спецификации не предъявляются.

Испытания должны проводиться во всех средах функционирования ПМ.

4.4.2 **Общие требования к приемке работ**

Предварительные и приемочные испытания должны проводиться Разработчиком ПМ.

Программы всех этапов испытаний составляются Разработчиком на основании документа «Программа и методика испытаний».

Программы испытаний должны предусматривать следующие виды проверок:

1. Проверка комплектности поставки ПМ и пакета документации.
2. Проверка содержания документации ПМ.
3. Проверка реализации функций ПМ в соответствии с требованиями спецификации.
4. Проверка требований к производительности ПМ.

По результатам этапов испытаний оформляются отчетные документы. К отчетным документам относятся Протоколы и Отчеты о результатах испытаний.

Отчетные документы подписываются членами комиссии и утверждаются председателем комиссии.

Предварительные испытания ПМ проводятся для определения его работоспособности, решения вопросов о возможности проведения приемочных испытаний и мероприятий по передаче ПМ в эксплуатацию. Предварительные испытания проводятся после отладки и тестирования ПМ.

В случае выявления отклонений от требований Спецификации формируется перечень необходимых доработок и рекомендуемые сроки устранения этих отклонений.

ПМ передается на приемочные испытания после устранения отклонений от Спецификации, зафиксированных в протоколе предварительных испытаний. При необходимости документация на ПМ дорабатывается.

Приемочные испытания ПМ проводят в соответствии с документом «Программа и методика приемочных испытаний».

По завершении приемочных испытаний оформляются соответствующие Протоколы, содержащие вывод о соответствии ПМ предъявляемым требованиям, а также критичность и сроки устранения дефектов, выявленных комиссией в ходе приемочных испытаний.

4.5 Требования к порядку встраивания ПМ в МП

4.5.1 Требования к организации работ

Встраивание ПМ в МП должно осуществляться в соответствии с порядком и требованиями, описанными в технической документации на ПМ и в документе [7].

Если ПМ является СКЗИ, то его встраивание в МП должно осуществляться в соответствии с правилами пользования СКЗИ и положениями документа [9].

При планировании и организации работ по встраиванию ПМ в МП необходимо учитывать, что в соответствии с п. 4.1. документа [25], МП должно пройти сертификацию в системе сертификации Федеральной службы по техническому и экспортному контролю или оценку соответствия по требованиям к оценочному уровню доверия (далее — ОУД) не ниже, чем ОУД 4, в соответствии с требованиями национального стандарта Российской Федерации [8], или обеспечить выполнение требований документа [26] в случаях, когда это необходимо.

Работы по встраиванию ПМ в МП должны соответствовать требованиям Приложения 2 документа [29].

Работы по встраиванию ПМ в МП должны включать работы по оценке влияния аппаратных, программно-аппаратных и программных средств сети (системы) конфиденциальной связи, совместно с которыми предполагается штатное функционирование ПМ, на выполнение предъявленных к входящему в его состав СКЗИ требований, если в документе [9] или в эксплуатационной документации на СКЗИ указывается на необходимость их проведения.

4.5.2 **Требования к разработчику МП**

Для встраивания ПМ в МП разработчик должен:

- иметь опыт программирования на языках Kotlin или Java, Swift, C++;
- обладать знаниями о современных стандартах в области криптографии и защиты информации;
- знать требования документов [11], предъявляемые к сертификатам ПлЦР, в случае использования ПМ для выполнения операций с ЦР;
- соответствовать требованиям, предъявляемым к разработчикам в документе [26] для прохождения оценки соответствия ОУД4;
- иметь лицензию ФСБ России на разработку, производство, распространение шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств, выполнение работ, оказание услуг в области шифрования информации, техническое обслуживание 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) на виды работ из Приложения к Положению [18], в части пунктов 1[7], 2, 3.

4.5.3 **Требования к совместимости с МП**

ПМ должен быть совместим с платформой разработки приложения или языком программирования для интеграции в существующую кодовую базу.

ПМ не должен оказывать негативного влияния на производительность приложения, время автономной работы или потребление данных.

[7] В случаях, когда ПМ является СКЗИ

4.5.4 Требования к использованию функций ПМ

При использовании методов ПМ должен выполняться контроль кодов возврата, возникающих при отклонениях от штатного выполнения команд.

Все вызовы методов интерфейса ПМ должны осуществляться вне основного потока приложения.

Вызов функций ПМ для осуществления защиты ЭС в случае выполнения операций с ЦР должен осуществляться в соответствии с требованиями документов, входящих в [11].

Реализация сценариев выпуска ключей и сертификатов должна соответствовать описаниям в документах [28] и [31].

4.5.5 Требования к механизмам доставки транспортных сертификатов и сертификатов проверки ЭП

МП должно содержать встроенный транспортный ГОСТ-TLS сертификат, корневые (самоподписанные) сертификаты УЦ Безопасности и ПУЦ УНЭП для установления соединения с сервисами финансовой организации и сертификат УЦ БР. При изменении данных сертификатов требуется обновление МП. Другие сертификаты должны загружаться в СКЗИ путем вызова соответствующей функции ПМ (см. п. 3.4.2. и 3.2.3.5 настоящей Спецификации).

В МП должны быть предусмотрен механизм обновления транспортного ГОСТ-TLS сертификата финансовой организации, сертификатов проверки ЭП финансовой организации, а также соответствующих цепочек сертификатов, включающих все промежуточные сертификаты. Для финансовой организации являющейся участником ПлЦР это сертификаты проверки ЭП КО и КК участника ПлЦР, КО и КК ПлЦР.

При обновлении сертификатов МП приложение должно вызывать соответствующий метод ПМ для загрузки сертификатов в СКЗИ.

4.6 Поддержка жизненного цикла ПМ

Разработчику ПМ рекомендуется предоставлять следующие услуги по обеспечению сопровождения ПМ:

- консультации по настройке и администрированию;
- консультации для разработчиков МП при встраивании ПМ;
- диагностику неисправностей и исправление найденных ошибок функциональности ПМ;
- устранение выявленных уязвимостей ПМ;
- поддержание программной и эксплуатационной документации ПМ в актуальном состоянии.

При выпуске новой версии ПМ разработчику рекомендуется предоставить обновленный пакет документации и пройти процедуру оценки влияния на СКЗИ (описание см. в п. 4.2.3 настоящей Спецификации) в случае:

- изменения требований к условиям функционирования ПМ, а именно изменения требований к программной и/или аппаратной среде функционирования;
- изменения функциональных требований к ПМ;
- изменения конфигурации или версии программного и аппаратного обеспечения, используемого в СКЗИ;
- выявления уязвимостей или инцидентов, которые могут повлиять на безопасность СКЗИ.

Дополнительные условия, при которых ПМ должен проходить оценку влияния в случае внесения изменений в программный код или документацию ПМ, также могут быть указаны в отчетной документации по результатам оценки влияния.

4.7 Характеристики пользователей и персонала

4.7.1 Требования к пользователям ПМ

Пользователями ПМ являются пользователи МП, совершающие операции с ЦР или другие финансовые операции через дистанционные каналы обслуживания финансовых организаций.

Требования к квалификации пользователей ПМ в настоящей Спецификации не описываются.

4.7.2 Требования к персоналу, обеспечивающему техническую поддержку и модернизацию ПМ

Для сопровождения ПМ рекомендуется создать Службу технической поддержки. При создании Службы технической поддержки необходимо учитывать, что ПМ имеет в своем составе СКЗИ. Любые работы по поддержке и модернизации ПМ должны выполняться с учетом требований правил пользования СКЗИ и положения [9]. Режим и технология работы Службы технической поддержки должны быть определены на стадии проектирования ПМ и описаны организацией-разработчиком ПМ в документе «Описание жизненного цикла ПМ».

Конкретный состав, должностные обязанности, режим и технология работы Службы технической поддержки должны быть определены на стадии проектирования ПМ и описаны в документе «Описание жизненного цикла ПМ» (см. п. 4.3.3 настоящей Спецификации).