ГОСУДАРСТВЕННЫЙ СТАНДАРТ
РЕСПУБЛИКИ БЕЛАРУСЬ

СТБ 34.101.27-2011

Информационные технологии и безопасность

ТРЕБОВАНИЯ БЕЗОПАСНОСТИ К ПРОГРАММНЫМ СРЕДСТВАМ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ

Інфармацыйныя тэхналогіі і бяспека

ПАТРАБАВАННІ БЯСПЕКІ ДА ПРАГРАМНЫХ СРОДКАЎ КРЫПТАГРАФІЧНАЙ АХОВЫ ІНФАРМАЦЫІ

Госстандарт
Минск

УДК 004.4.056.55(083.74)(476) МКС 35.240.40 КП 05

Ключевые слова: технологии информационные, безопасность, криптографическая защита информации, программное средство криптографической защиты информации

Предисловие

Цели, основные принципы, положения по государственному регулированию и управлению в области технического нормирования и стандартизации установлены Законом Республики Беларусь «О техническом нормировании и стандартизации».

1 РАЗРАБОТАН учреждением Белорусского государственного университета «Научно-исследовательский институт прикладных проблем математики и информатики»

ВНЕСЕН Оперативно-аналитическим центром при Президенте Республики Беларусь (ОАЦ)

2 УТВЕРЖДЕН и ВВЕДЕН В ДЕЙСТВИЕ постановлением Госстандарта Республики Беларусь от 25 ноября 2011 г. № 83

3 ВЗАМЕН СТБ П 34.101.27-2007

Издан на русском языке

Содержание

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РЕСПУБЛИКИ БЕЛАРУСЬ

Информационные технологии и безопасность

ТРЕБОВАНИЯ БЕЗОПАСНОСТИ К ПРОГРАММНЫМ СРЕДСТВАМ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ

Інфармацыйныя тэхналогіі і бяспека

ПАТРАБАВАННІ БЯСПЕКІ ДА ПРАГРАМНЫХ СРОДКАЎ КРЫПТАГРАФІЧНАЙ АХОВЫ ІНФАРМАЦЫІ

Information technologies and security

Security requirements for software cryptographic modules

Дата введения 2012-03-01

1 Область применения

Настоящий государственный стандарт устанавливает общие требования безопасности к программным средствам, которые используются для криптографической защиты информации ограниченного распространения (за исключением государственных секретов).

Стандарт предназначен для использования:

– заказчиками (потребителями) при задании требований безопасности к программным средствам криптографической защиты информации;

– разработчиками при создании программных средств криптографической защиты информации;

– экспертами при оценке надежности программных средств криптографической защиты информации.

2 Термины и определения

В настоящем стандарте применяют следующие термины с соответствующими определениями:

2.1 аутентификация: Проверка подлинности идентификатора оператора.

2.2 аутентификационные данные: Информация, которая используется для аутентификации.

2.3 генератор случайных чисел: Аппаратно-программное устройство, которое вырабатывает последовательность непредсказуемых элементов.

2.4 долговременный объект: Объект, который хранится в пределах криптографической границы или передается за ее пределы, операции над которым могут выполняться в нескольких сеансах.

2.5 зашифрование: Преобразование объектов, направленное на обеспечение их конфиденциальности, которое осуществляется с использованием секретного или открытого ключа.

2.6 защита объектов: Контроль целостности и обеспечение конфиденциальности критических объектов, контроль целостности открытых объектов.

2.7 идентификация: Присвоение операторам уникальных идентификаторов.

2.8 имитовставка: Контрольная характеристика объекта, которая определяется с использованием секретного ключа и служит для контроля целостности и подлинности объекта.

2.9 имитозащита: Контроль целостности объектов, который реализуется путем выработки и проверки имитовставок.

2.10 клиентская программа: Программа, которая от лица оператора вызывает сервисы программного средства криптографической защиты информации.

2.11 конфиденциальность: Гарантия того, что объекты доступны для использования только тем сторонам, которым они предназначены.

2.12 криптографическая граница: Точно определенный разработчиком непрерывный физический периметр в среде эксплуатации, который определяет контролируемую границу программного средства криптографической защиты информации.

2.13 криптографический алгоритм: Алгоритм преобразования объектов, направленный на обеспечение их конфиденциальности, контроля целостности или подлинности, в том числе алгоритм управления криптографическими ключами для защиты объектов.

2.14 криптографический ключ: Объект-параметр, используемый вместе с криптографическим алгоритмом для управления операциями зашифрования и расшифрования, вычисления и проверки электронной цифровой подписи, выработки и проверки имитовставки, выработки псевдослучайных данных, выработки совместно используемой конфиденциальной информации.

2.15 криптографический протокол: Точно определенные последовательность действий или набор правил, предусматривающие взаимодействие двух и более сторон с использованием криптографических алгоритмов.

2.16 критические системные компоненты: Находящееся внутри криптографической границы аппаратное и программное обеспечение, которое используется для передачи, обработки и хранения объектов программного средства криптографической защиты информации.

2.17 критический объект: Объект, несанкционированные раскрытие или модификация которого снижают безопасность.

2.18 личный ключ: Криптографический ключ, используемый вместе с криптографическим алгоритмом с открытым ключом, который однозначно связан с конкретным оператором и не является общедоступным.

2.19 неявная копия: Копия объекта, переданная по побочному каналу.

2.20 объект: Элемент, который содержит или получает информацию и над которым выполняются операции.

2.21 оператор: Лицо или клиентская программа, выступающая от имени лица, которые взаимодействуют с программным средством криптографической защиты информации.

2.22 открытый объект: Объект, несанкционированная модификация которого снижает безопасность, а раскрытие — не снижает.

2.23 открытый ключ: Криптографический ключ, используемый вместе с криптографическим алгоритмом с открытым ключом, который строится по личному ключу и может быть сделан общедоступным.

2.24 побочный канал: Нежелательный дополнительный канал передачи информации о входных, промежуточных или выходных данных криптографического алгоритма или протокола, возникающий из-за особенностей его аппаратно-программной реализации.

2.25 подлинность: Гарантия того, что сторона действительно является владельцем (создателем, отправителем) определенного объекта.

2.26 политика управления доступом: Правила, определяющие допустимые операции операторов над сервисами и объектами.

2.27 программное средство криптографической защиты информации; ПСКЗИ: Средство криптографической защиты информации, выполненное целиком программно, без аппаратных компонентов.

2.28 разделение секрета: Разбиение критического объекта на частичные секреты, каждый из которых по отдельности или даже вместе с некоторыми другими частичными секретами не дает информации об исходном объекте.

2.29 расшифрование: Преобразование, обратное зашифрованию, которое определяется с помощью секретного или личного ключа.

2.30 сеанс оператора: Период взаимодействия оператора с программным средством криптографической защиты информации.

2.31 сеансовый объект: Объект, который создается, используется и уничтожается в течение одного сеанса.

2.32 секрет аутентификации: Пароль, PIN-код и другие аутентификационные данные, которые однозначно связаны с конкретным оператором и не являются общедоступными.

2.33 секретный ключ: Криптографический ключ, используемый вместе с криптографическим алгоритмом с секретным ключом, который однозначно связан с конкретным оператором или группой операторов и не является общедоступным.

2.34 сервис: Реализованная в программном средстве криптографической защиты информации и доступная оператору функция.

2.35 синхропосылка: Открытые входные данные криптографического алгоритма или протокола, которые обеспечивают уникальность результатов криптографического преобразования на фиксированном ключе.

2.36 системный сеанс: Непрерывный период работы программного средства криптографической защиты информации.

2.37 среда эксплуатации: Аппаратно-программное обеспечение, организационные процедуры и мероприятия, необходимые для функционирования программного средства криптографической защиты информации.

2.38 средство криптографической защиты информации; СКЗИ: Набор аппаратно-программных компонентов, который реализует криптографические алгоритмы и протоколы, а также возможно дополнительные средства управления ключами, контроля доступа и проверки работоспособности, предназначенные для безопасного управления вызовами криптографических алгоритмов и их входными и выходными данными.

2.39 целостность: Гарантия того, что объект не изменен при хранении или передаче.

2.40 шифрование: Зашифрование или расшифрование.

2.41 хэш-значение: Контрольная характеристика объекта, которая определяется без использования ключа и служит для контроля целостности объекта и для представления объекта в сжатой форме.

2.42 хэширование: Выработка хэш-значений.

2.43 частичный секрет: Критический объект, полученный в результате применения метода разделения секрета.

2.44 электронная цифровая подпись; ЭЦП: Контрольная характеристика объекта, которая определяется с использованием личного ключа, проверяется с использованием

открытого ключа, служит для контроля целостности и подлинности объекта и обеспечивает невозможность отказа от авторства.

3 Оформление требований безопасности

Требования безопасности разбиваются на группы. Группы обозначаются двухбуквенными кодами: КП, РС, УД и др. Требования внутри группы нумеруются последовательно, начиная с единицы.

Требования безопасности задают разбиение ПСКЗИ на два класса. Для каждого требования в круглых скобках перечисляются классы ПСКЗИ, на которые требования распространяются. Возможные варианты: (1), (2) или (1, 2).

В тексте имеются ссылки на требования. Ссылка [Т] означает, что условия требования Т должны быть использованы в месте ссылки. Ссылка {Т} означает, что условия или пояснения в месте ссылки детализируют Т.

Формулировки требований безопасности могут включать указания на необходимость определения списка методов, списка компонентов, списка средств и др. Если не оговорено противное, то результатом определения может быть пустой список.

4 Общие положения

4.1 Назначение

Настоящий стандарт устанавливает требования безопасности к программным средствам криптографической защиты информации. ПСКЗИ представляет собой набор программ и связанных с ними данных, который реализует криптографические алгоритмы и протоколы, а также дополнительно:

– функции управления данными, включая средства генерации, экспорта, импорта и уничтожения ключей;

– механизмы контроля доступа к данным, включая средства идентификации и аутентификации;

– механизмы проверки работоспособности, включая средства тестирования, диагностики и контроля целостности, предназначенные для безопасного управления вызовами криптографических алгоритмов и их входными и выходными данными.

Требования безопасности делятся на функциональные (разделы 5, 6) и гарантийные (раздел 7). Функциональные требования направлены на решение задач безопасности, гарантийные требования поддерживают качество решения данных задач. Функциональные требования обеспечивают противодействие угрозам безопасности, следование определенным правилам безопасности. Гарантийные требования обеспечивают доверие к тому, что программное средство корректно спроектировано и разработано, протестировано в достаточном объеме, правильно установлено и эксплуатируется.

Программы ПСКЗИ выполняются в определенной среде эксплуатации — на персональном компьютере, в мобильном устройстве, на смарт-карте и др. Среда эксплуатации обеспечивает ПСКЗИ процессорными ресурсами, ресурсами памяти, а также предоставляет системные средства управления этими ресурсами. Программное средство зависимо от ресурсов среды и не в состоянии обеспечить безопасность своих объектов самостоятельно. Поэтому функциональные требования реализуются как средствами ПСКЗИ (раздел 5), так и средствами среды (раздел 6).

Требования безопасности задают разбиение ПСКЗИ на два класса. К средствам первого класса выдвигается базовый набор требований, к средствам второго — усиленный. Средства второго класса рекомендуется использовать в тех случаях, когда возможны изменения критических системных компонентов в среде эксплуатации, например, если разрешена установка новых программ, обновление операционной системы, замена оборудования и др. Средства второго класса рекомендуется применять также тогда, когда гипотетический нарушитель имеет возможность наблюдать за побочными каналами, например за временем выработки ЭЦП на смарт-карте.

Методы реализации требований безопасности описываются в функциональной спецификации. Функциональная спецификация может представлять собой отдельный документ либо разделы нескольких документов. Примерное содержание функциональной спецификации представлено в приложении А. Приложение Б содержит примерную спецификацию гипотетического программного средства «Криптодиск».

4.2 Криптографическая поддержка

ПСКЗИ реализует один или несколько криптографических алгоритмов или протоколов. Алгоритмы и протоколы используются как в доступных извне сервисах, так и во внутренних средствах безопасности, обеспечивающих защиту объектов.

Входные данные криптографического алгоритма делятся на служебные и собственно обрабатываемые (содержательные) объекты. К служебным объектам относятся долговременные параметры, которые задают семейство криптографических преобразований для данного алгоритма, ключи, которые определяют выбор одного преобразования из семейства, синхропосылки, которые обеспечивают уникальность результатов преобразования на фиксированном ключе, и др.

Криптографические алгоритмы делятся на алгоритмы с секретным ключом (симметричные), алгоритмы с открытым ключом (асимметричные) и бесключевые алгоритмы. В симметричных алгоритмах для выполнения нескольких связанных операций, например зашифрования и расшифрования, используется один и тот же секретный ключ. В асимметричных алгоритмах используется пара ключей — личный и соответствующий ему открытый. Например, в алгоритмах ЭЦП при выработке подписи используется личный ключ, а при ее проверке — открытый. К бесключевым относятся алгоритмы хэширования, алгоритмы разделения секрета, алгоритмы построения семейства ключей по одному главному ключу. В этих алгоритмах ключи, отвечающие за выбор криптографического преобразования, не используются, хотя обрабатываемые объекты сами могут являться ключами {КП.1}.

При реализации криптографических алгоритмов разрешается сужать множество обрабатываемых объектов. Например, алгоритм шифрования может обрабатывать сообщения не любой, а только определенной длины. Вместе с тем сужение множества ключей не допускается {КП.1}.

Кроме собственно криптографических алгоритмов, их спецификации могут определять методы генерации долговременных параметров и ключей. Методы могут определяться рамочно, без исчерпывающих деталей. Например, при генерации параметров ЭЦП на основе эллиптических кривых используются вспомогательные алгоритмы проверки простоты чисел, расчета порядка группы точек эллиптической кривой, извлечения квадратных корней и др., которые могут быть не определены в спецификации. При реализации

в ПСКЗИ методов генерации параметров проводится уточнение вспомогательных алгоритмов. Уточнения могут касаться также способов генерации случайных чисел, по которым строятся искомые долговременные параметры или ключи {КП.3}.

Криптографические протоколы представляют собой наборы криптографических алгоритмов, которые выполняются в определенной последовательности двумя (или более) сторонами. Для протоколов также справедливы приведенные выше пояснения.

В ПСКЗИ реализуются средства тестирования криптографических алгоритмов и протоколов. Для тестирования могут использоваться: тесты известного ответа, тесты прямого и обратного преобразований, тесты на соответствие между открытым и личным ключами {СТ.3}.

4.3 Операторы

С ПСКЗИ взаимодействуют операторы. Взаимодействие состоит в работе с сервисами. При использовании сервиса оператор готовит и передает входные данные и управляющие параметры сервиса, вызывает сервис, получает и обрабатывает выходные данные.

Некоторые операции могут выполняться не одним, а несколькими сервисами. Например, защита канала связи может быть реализована сервисом выработки общего сеансового ключа и сервисом шифрования данных канала. Определенные последовательности вызовов сервисов могут быть запрещены. Например, запрещено выполнять шифрование до выработки общего сеансового ключа {РС.1}.

Сервисы доступны операторам через устройства ввода/вывода (клавиатуру, дисплей, принтер, порты) и логические интерфейсы (например, наборы функций динамических библиотек). Оператор может взаимодействовать с ПСКЗИ не напрямую, а через клиентские программы (клиентов электронной почты, браузеры, программы управления электронными документами).

ПСКЗИ поддерживает определенные роли (категории) операторов. Имеется роль «Администраторы». Операторы этой роли наделены правами выполнять административные сервисы ПСКЗИ: устанавливать и настраивать ПСКЗИ, управлять правами операторов, вводить мастер-ключи. Рекомендуется определять роль «Пользователи». Пользователи выполняют общие сервисы ПСКЗИ: шифруют почтовые отправления, проверяют ЭЦП электронных документов, генерируют ключи {УД.3}.

Имеется предопределенная роль «Система», которая явно может не определяться. Неявный оператор этой роли (системный оператор) выполняет внутренние сервисы ПСКЗИ: самотестирование при загрузке, контроль доступа, управление состояниями сеансов.

Один и тот же оператор может выполнять сразу несколько ролей {ИА.1}.

4.4 Аутентификация

В среде эксплуатации ПСКЗИ реализуются средства аутентификации для проверки принадлежности оператора к явным ролям и допустимости выполнения оператором сервисов данных ролей. Аутентификация требуется также для проверки прав доступа вызываемых оператором сервисов к открытым и критическим объектам. Полный доступ к таким объектам разрешается, как правило, только владельцам и для проверки полномочности доступа кроме роли, используется также идентификатор оператора. Допускается,

что для выполнения некоторых сервисов или доступа к некоторым объектам требуется дополнительная аутентификация {ИА.3}.

Средства аутентификации могут быть полностью или частично реализованы самим ПСКЗИ.

Методы аутентификации основываются на использовании комбинаций из трех факторов: владение операторами устройствами аутентификации («что я имею»), знание секретов аутентификации («что я знаю»), обладание биометрическими характеристиками («кто я»). Рекомендуется задействовать более одного фактора, например использовать карты доступа вместе с PIN-кодами доступа {ИА.4}.

При аутентификации операторы предъявляют устройства аутентификации (смарт-карты) и вводят аутентификационные данные (пароли, отпечатки пальцев). Средства аутентификации проверяют корректность представленных устройств и сравнивают характеристики аутентификационных данных с контрольными значениями.

При создании и изменении секретов аутентификации предусматривается проверка их качества. Проверка может состоять в контроле длины пароля или контроле включения в пароль как цифровых, так и буквенных символов в различных регистрах. Проверка качества секретов не может основываться на ограничениях в руководствах, т. е. на предположении о том, что оператор обязательно задаст пароль нужного качества {ИА.7}.

4.5 Сеансы

Взаимодействие оператора с ПСКЗИ происходит в форме сеансов. В начале сеанса, как правило, выполняются сервисы идентификации и аутентификации оператора, в конце сеанса — очистка и уничтожение сеансовых объектов. Отдельно выделяется системный сеанс неявного системного оператора, который совпадает с непрерывным периодом работы ПСКЗИ (от загрузки до выгрузки программ).

Системный сеанс ПСКЗИ может находиться в одном из нескольких состояний, которые характеризуются различными правами доступа к сервисам. Имеются состояния, соответствующие сеансам операторов различных ролей. В этих состояниях операторы могут выполнять сервисы в соответствии с политикой управления доступом. Имеется также состояние блокировки, в котором выполнение сервисов запрещается или сильно ограничивается. Примеры других состояний системного сеанса: «самотестирование», «ожидание» (до аутентификации операторов), «временная блокировка» (блокировка на определенный период времени после превышения порога неудачных попыток аутентификации) и др. Могут определяться подчиненные состояния, например внутренние состояния сеансов администратора и пользователя {УД.4}.

Блокировка может состоять в принудительном завершении программ {УД.6}.

Многозадачные операционные системы могут поддерживать одновременное выполнение сеансов для нескольких операторов ПСКЗИ. При этом состояния «сеанс администратора», «сеанс пользователя» могут дублироваться. Считается, что ПСКЗИ одновременно может находиться сразу во всех состояниях, соответствующих сеансам операторов. Такой режим реализуется средствами обеспечения многозадачности операционной системы {УД.4}.

4.6 Объекты

С помощью сервисов операторы выполняют операции над объектами — ключами, открытыми текстами, параметрами криптографических алгоритмов, сертификатами открытых ключей и др.

Объекты делятся на открытые (общедоступные) и критические (ограниченного доступа). Например, открытый ключ подписи является открытым объектом, личный ключ подписи — критическим {УД.2}.

Кроме этого, объекты делятся на сеансовые и долговременные. Сеансовые объекты создаются во время сеансов операторов и уничтожаются при их завершении. Доступ к таким объектам имеет только оператор сеанса. Долговременные объекты существуют как во время, так и вне времени выполнения сеансов, хранятся в пределах криптографической границы или передаются за ее пределы. Доступ к долговременным объектам имеют все операторы с соответствующими правами. Например, личный ключ подписи на магнитном носителе является долговременным объектом, а генерируемое при выработке подписи случайное число — сеансовым {УД.2}.

У объектов есть владельцы. Например, объектами администратора являются журнал аудита (критический объект), настройки политики управления доступом (открытый объект). Пользователи владеют идентификаторами, ключами, документами. Файлы программ и конфигураций относятся к системным объектам, владельцем которых является неявный системный оператор {УД.2}.

При выполнении сервисов ПСКЗИ может создаваться несколько сеансовых копий одного и того же объекта в нескольких переменных программ. Сеансовые копии не обязательно определять как отдельные объекты, достаточно обеспечить очистку сеансовых копий критических объектов {УД.2}.

К объектам ПСКЗИ не относятся секреты аутентификации. При передаче таких секретов от операторов сервисам их защита обеспечивается средствами среды эксплуатации. Однако контрольные значения секретов являются долговременными объектами ПСКЗИ. Кроме этого, в ходе сеансов на основе аутентификационных данных могут вырабатываться сеансовые объекты (например, хэш-значение пароля, используемое в качестве ключа), которые защищаются средствами ПСКЗИ {УД.2}.

Обращения операторов, прошедших аутентификацию, к сервисам и объектам ПСКЗИ регулируются политикой управления доступом. Политика регламентирует набор допустимых операций операторов над сервисами и объектами. Список разрешенных операций определяется идентификатором и ролью оператора, типом объекта, владельцем объекта, состоянием, в котором находится сеанс оператора, и другими условиями. Список операций может включать: выполнение сервисов, создание, удаление, экспорт, импорт, чтение, запись объектов {УД.3}.

4.7 Защита объектов

Сеансовый объект может быть преобразован в долговременный, т. е. может экспортироваться за пределы сеанса. При экспорте сеансовые объекты защищаются, т. е. обеспечивается

- конфиденциальность критических объектов;
- контроль целостности открытых и критических объектов.

Обратно долговременные объекты могут импортироваться из-за пределов сеанса. При импорте проверяется целостность объектов и дополнительно критические объекты преобразуются из защищенной формы в открытую, например расшифровываются.

Для защиты объектов используются следующие методы:

1 *Криптографические методы.* Состоят в применении алгоритмов шифрования (блочного, поточного, с открытым ключом) для обеспечения конфиденциальности, алгоритмов ЭЦП и имитозащиты для контроля целостности {ЗО.3}, {ЗО.4}.

2 *Аппаратные методы.* Состоят в аппаратной защите от несанкционированного чтения и (или) модификации областей памяти, в которых размещаются целевые объекты. Аппаратная защита обеспечивается применением смарт-карт, токенов и других подобных устройств {ЗО.5}.

3 *Методы разделения секрета.* Состоят в разбиении защищаемого критического объекта на частичные секреты, каждый из которых затем защищается по отдельности. Простейшим методом разделения секрета является представление ключа как двоичного слова в виде суммы (поразрядной по модулю 2) нескольких частичных секретов. В этом случае для восстановления исходного ключа требуется располагать всеми частичными секретами. Более сложные пороговые методы позволяют определить исходный ключ при наличии не всех, а только порогового числа частичных секретов, например трех из пяти {ЗО.6}.

4 *Алгоритмические методы.* Состоят в контроле целостности с помощью некриптографических или бесключевых криптографических алгоритмов. К алгоритмическим методам относятся: сверка нескольких копий объекта, проверка контрольных хэш-значений, самоподпись сертификата открытого ключа {ЗО.7}.

5 *Организационные меры.* Состоят в обеспечении конфиденциальности с помощью мероприятий, не относящихся к информационным технологиям.

При выборе методов защиты предпочтение следует отдавать криптографическим методам. Однако для организации криптографической защиты объектов при экспорте и импорте требуется использовать другие объекты-ключи. Если эти объекты также нужно экспортировать или импортировать, то для их защиты должны использоваться новые ключи и т. д. Аппаратные методы и методы разделения секрета позволяют прервать цепочку ключей защиты других ключей.

В алгоритмических методах защиты контрольная характеристика, на основании которой принимается решение о целостности объекта, может не защищаться и храниться вместе с самим объектом. Тогда алгоритмический метод обеспечивает защиту только от случайных сбоев в среде эксплуатации, но не от преднамеренного воздействия. Если же контрольная характеристика защищается, то защита распространяется на контролируемый объект. Пусть, например, ПСКЗИ вырабатывает личный и открытый ключи, сохраняет личный ключ на смарт-карту, а открытый ключ отсылает в удостоверяющий центр для получения сертификата. После получения сертификата ПСКЗИ вычисляет по сохраненному личному ключу открытый и сравнивает его с ключом, размещенным в сертификате. Проверка связи между ключами соответствует алгоритмическому методу защиты. Аппаратная защита личного ключа распространяется в момент импорта на открытый ключ сертификата.

Защита критических сеансовых объектов включает невозможность их определения после завершения сеансов. Для очистки сеансовых объектов в оперативной памяти можно использовать обнуление ячеек памяти, а для очистки объектов в постоянной перепрограм-

мируемой памяти может быть использована многократная перезапись ячеек константами и случайными данными {ЗО.12}.

4.8 Среда эксплуатации

Для учета возможных уязвимостей в среде эксплуатации разработчик определяет криптографическую границу — непрерывный физический периметр, который задает контролируемую границу ПСКЗИ.

В криптографической границе ПСКЗИ выделяются критические системные компоненты — аппаратное и программное обеспечение, которое используется для передачи, обработки и хранения объектов ПСКЗИ. К критическим системным компонентам относятся:

- устройства ввода/вывода;
- устройства обработки и передачи (процессор, физические интерфейсы);
- устройства хранения (жесткий диск);
- службы операционной системы, влияющие на безопасность ПСКЗИ;
- дополнительное оборудование (генераторы случайных чисел).

ПСКЗИ не может обеспечить безопасность критических системных компонентов, обеспечение безопасности возлагается на среду. Перед установкой и использованием ПСКЗИ производится настройка среды. При настройке конфигурируются средства защиты операционной системы, задаются разрешения на установку программ, вводятся ограничения на доступ к системным объектам и др.

В ПСКЗИ предусматриваются средства проверки условий безопасности среды, направленные на контроль состава и правильного функционирования критических компонентов. Проверки могут быть косвенными. Например, успешная загрузка операционной системы может являться основанием для вывода о корректности работы устройств обработки и запоминающих устройств. В свою очередь проверка операционной системы может заключаться в контроле версий ее модулей. В большинстве случаев нельзя провести исчерпывающее тестирование критических системных компонентов. Допускается, что при тестировании проверяется только их наличие и проводится контроль лишь нескольких их основных функций. Для некоторых компонентов при начальном запуске можно провести лишь часть проверок. В таких случаях разрешается выполнить пропущенные тесты позднее. Например, при начальном запуске проверяется наличие устройства чтения смарт-карт, а корректность работы данного устройства проверяется при непосредственном чтении данных с карты {СТ.2}.

При обработке объектов ПСКЗИ в критических системных компонентах могут появляться побочные каналы, по которым передается информация о критических объектах. Например, алгоритм выработки ЭЦП может быть реализован таким образом, что по времени его выполнения нарушитель может сделать вывод о значении некоторых битов личного ключа {КП.4}.

Побочным каналом является канал сохранения неявных копий сеансовых объектов в файле подкачки, регистрах процессора, журнале аудита {РС.4}.

В ПСКЗИ второго класса предусматриваются средства защиты от побочных каналов.

4.9 Генерация случайных чисел

В криптографической границе ПСКЗИ могут находиться генераторы случайных чисел, которые вырабатывают данные для создания секретных и личных ключей, синхропо-

сылок, других критических или уникальных объектов. К генераторам случайных чисел не относятся алгоритмы выработки псевдослучайных чисел, хотя ключи таких алгоритмов могут строиться с помощью генераторов.

Генератор случайных чисел выдает последовательности, каждый следующий элемент которых статистически и вычислительно трудно предсказать по всем предыдущим элементам. Генератор использует один или несколько источников случайности (неопределенности, энтропии) и включает средства обработки данных от источников. Средства обработки могут частично или полностью размещаться в ПСКЗИ {СЧ.1}.

В компьютерных системах используются следующие источники случайности {СЧ.1}:

– физические источники, использующие процессы в физических устройствах (например, шум в радиоэлектронных приборах);

– системные источники, использующие состояния, процессы и события операционной системы (системное время, сетевая активность, прерывания);

– источники, основанные на активности операторов (движения мышью, нажатия клавиш).

Предпочтительным является использование физических источников случайности.

Для источника случайности S проводится оценка энтропии (неопределенности, вариабельности) его выходных последовательностей. Для этого строится вероятностная модель S и в рамках этой модели определяется величина h такая, что основная вероятностная масса выходных последовательностей длины n сосредоточена на множестве мощности 2^{nh}. Величина h называется удельной энтропией на наблюдение. Например, если S выдает случайные независимые символы алфавита A и вероятность появления символа α равняется p_α, то удельная энтропия

$$h = -\sum_{\alpha \in A} p_\alpha \log_2 p_\alpha \quad (0 \cdot \log_2 0 = 0).$$

Кроме h существуют и другие характеристики неопределенности, в частности минимальная удельная энтропия h_{min}, которая характеризует сложность предсказания самой вероятной выходной последовательности S. Для описанного выше источника величина h_{min} определяется как $\min_{\alpha \in A}(-\log_2 p_\alpha)$.

Оценка h (или h_{min}) является сложной задачей, если распределение $\{p_\alpha\}$ известно не полностью, источник S не является стационарным, между выходными символами S имеются зависимости и в других ситуациях. Для оценки h могут применяться статистические методы, основанные на частотах встречаемости в выходных последовательностях m-грамм, а также алгоритмические методы, основанные на коэффициентах обратимого или необратимого сжатия выходных последовательностей {СЧ.2}.

Если удельная энтропия h оценена, то можно сделать вывод о том, что для надежной генерации l-битового секретного ключа требуется использовать не менее l/h наблюдений от источника случайности {СЧ.2}.

Для выявления отказов и сбоев в функционировании физических источников случайности при генерации критических объектов проводится тестирование выходных по-

следовательностей генераторов случайных чисел. Тестирование может быть статистическим {СЧ.5}.

5 Функциональные требования безопасности к программному средству криптографической защиты информации

5.1 Требования по криптографической поддержке (КП)

Требование КП.1 (1, 2). Должны быть определены и корректно реализованы криптографические алгоритмы и протоколы ПСКЗИ. Каждый алгоритм и протокол должен быть однозначно идентифицирован: должен быть указан его тип, дана ссылка на спецификацию, определены режим работы, поддерживаемые длины ключей.

Требование КП.2 (1, 2). Спецификации криптографических алгоритмов и протоколов [КП.1] должны быть приняты в качестве технических нормативно-правовых актов (далее — ТНПА).

Требование КП.3 (1, 2). Если в ПСКЗИ предусмотрены методы генерации долговременных параметров и ключей криптографических алгоритмов и протоколов [КП.1], то эти методы должны быть определены и корректно реализованы. Методы генерации должны соответствовать спецификациям на алгоритмы и протоколы или уточнять данные спецификации.

Требование КП.4 (2). Криптографические алгоритмы и протоколы [КП.1], а также методы генерации их ключей, должны быть реализованы так, чтобы по времени их выполнения нельзя было сделать вывод об используемых или генерируемых личных и секретных ключах.

5.2 Требования по реализации сервисов (РС)

Требование РС.1 (1, 2). Должны быть определены и корректно реализованы сервисы ПСКЗИ. Для каждого сервиса должно быть определено его назначение, входные и выходные данные. Если сервисы выполняются в определенной последовательности, то данная последовательность должна быть определена.

Требование РС.2 (1, 2). В список сервисов должны быть включены {РС.1}:

– сервис вывода номера версии ПСКЗИ;

– сервисы самотестирования [СТ.2], [СТ.3];

– по крайней мере один сервис, который реализует криптографический алгоритм или протокол [КП.1].

Требование РС.3 (1, 2). На любых входных данных сервисы [РС.1] должны нормально завершаться, т. е. возвращать правильные результаты, в том числе признаки некорректных входных данных.

Требование РС.4 (2). Сервисы [РС.1], которые выполняют операции над критическими объектами [УД.2], должны быть реализованы так, чтобы после их завершения в пределах криптографической границы не оставалось неявных копий критических объектов в открытом виде.

5.3 Требования по управлению доступом (УД)

Требование УД.1 (1, 2). Должны быть определены роли операторов ПСКЗИ. Должна быть предусмотрена роль «Администраторы».

Требование УД.2 (1, 2). Должны быть определены объекты ПСКЗИ. Для каждого объекта должно быть задано его назначение, проведена классификация (открытый или критический, сеансовый или долговременный), определен владелец.

Требование УД.3 (1, 2). Должна быть определена и корректно реализована политика управления доступом ПСКЗИ. Политика должна устанавливать набор допустимых операций операторов различных ролей [УД.1] над сервисами [РС.1] и сервисов, выступающих от имени операторов, над объектами [УД.2] и другими сервисами.

Требование УД.4 (1, 2). Должны быть определены состояния системного сеанса ПСКЗИ. Должны быть определены и корректно реализованы правила перехода между состояниями. Должны быть предусмотрены состояния, соответствующие сеансам операторов различных ролей [УД.1], и состояние блокировки.

Требование УД.5 (1, 2). В состояниях, соответствующих сеансам операторов различных ролей, должна действовать политика управления доступом [УД.3] относительно данных ролей. Перед переходом в состояния должна проводиться аутентификация операторов [ИА.3].

Требование УД.6 (1, 2). В состоянии блокировки должно быть запрещено выполнение всех сервисов, кроме сервисов самотестирования [СТ.3], сервисов аутентификации администратора [ИА.3] и вспомогательных сервисов, не связанных с обработкой объектов внутри криптографической границы. При переходе в состояние блокировки должны быть завершены все открытые сеансы операторов. Если системный сеанс ПСКЗИ завершен в состоянии блокировки, то и начаться он должен в этом состоянии.

5.4 Требования по защите объектов (ЗО)

Требование ЗО.1 (1, 2). Должна обеспечиваться конфиденциальность критических объектов [УД.2]. Для этого должны использоваться криптографические методы, аппаратные методы или методы разделения секрета. Для обеспечения конфиденциальности частичных секретов могут дополнительно использоваться организационные меры.

Требование ЗО.2 (1, 2). Должен осуществляться контроль целостности критических и открытых объектов [УД.3]. Для этого должны использоваться криптографические, аппаратные или алгоритмические методы.

Требование ЗО.3 (1, 2). Должны быть определены и корректно реализованы криптографические методы обеспечения конфиденциальности. Методы должны быть основаны на алгоритмах шифрования [КП.1]. Личные и секретные ключи алгоритмов должны быть отнесены к критическим объектам, а открытые ключи и долговременные параметры — к открытым объектам {УД.2}.

Требование ЗО.4 (1, 2). Должны быть определены и корректно реализованы криптографические методы контроля целостности. Методы должны быть основаны на алгоритмах ЭЦП и имитозащиты [КП.1]. Личный ключ ЭЦП и секретный ключ имитозащиты должны быть отнесены к критическим объектам, а открытый ключ ЭЦП и долговременные параметры — к открытым объектам {УД.2}. Длина имитовставки должна выбираться так, чтобы вероятность необнаружения модификации объекта нарушителем, который не знает ключ, не превышала 2^{-32}.

Требование ЗО.5 (1, 2). Должны быть определены и корректно использованы аппаратные методы защиты. Устройства, которые реализуют аппаратные методы, должны соответствовать ТНПА в части физической безопасности.

Требование ЗО.6 (1, 2). Должны быть определены и корректно реализованы методы разделения секрета. При восстановлении критического объекта должно использоваться не менее двух различных частичных секретов. Если для восстановления критического объекта требуется k частичных секретов, то любые $k-1$ частичных секретов не должны давать никакой информации об исходном объекте. Частичные секреты должны быть отнесены к критическим объектам {УД.2}. Владельцы частичных секретов должны быть различны {УД.2}.

Требование ЗО.7 (1, 2). Должны быть определены и корректно реализованы алгоритмические методы контроля целостности. Алгоритмические методы контроля должны гарантировать, что вероятность необнаружения случайной модификации контролируемого объекта не превышает 2^{-32}. Если контролируемый объект не является частичным секретом или системным объектом, то его контрольная характеристика должна быть отнесена к открытым или критическим объектам {УД.2}.

Требование ЗО.8 (1, 2). Должны быть определены и изложены в руководствах {РД.1}, {РД.2} организационные меры по обеспечению конфиденциальности частичных секретов. Меры должны быть направлены на ограничение физического доступа к носителям информации, на которых хранятся секреты, и попаданию порогового числа частичных секретов в руки одного лица.

Требование ЗО.9 (1, 2). При экспорте критических объектов должна устанавливаться их защита [ЗО.1].

Требование ЗО.10 (1, 2). При импорте критических и открытых объектов должен проводиться контроль их целостности [ЗО.2]. При нарушении целостности использование объекта должно быть запрещено.

Требование ЗО.11 (1, 2). Контроль целостности системных объектов [ЗО.2] должен проводиться при самотестировании {СТ.3}.

Требование ЗО.12 (1, 2). Все сенсовые критические объекты [УД.2] должны очищаться до завершения сеансов.

5.5 Требования по самотестированию (СТ)

Требование СТ.1 (1, 2). Должны быть определены критические системные компоненты ПСКЗИ.

Требование СТ.2 (1, 2). При начальном запуске ПСКЗИ должно проверять состав и работоспособность критических системных компонентов [СТ.1].

Требование СТ.3 (1, 2). При начальном запуске, а также по запросу оператора должно проводиться тестирование, обязательно включающее:

- тесты криптографических алгоритмов и протоколов [КП.1];
- контроль целостности всех системных объектов, включая файлы программ [ЗО.11];
- тесты для генераторов случайных чисел [СЧ.5].

Требование СТ.4 (1, 2). Тестовые данные криптографических алгоритмов и протоколов (ключи, открытые тексты, шифртексты) должны быть отнесены к открытым объектам {УД.2}.

Требование СТ.5 (1, 2). При ошибках тестирования системный сеанс ПСКЗИ должен переходить в состояние блокировки {УД.6}.

5.6 Требования по генерации случайных чисел (СЧ)

Требование СЧ.1 (1, 2). Должны быть определены генераторы случайных чисел, которые используются для выработки ключей и других критических объектов [КП.3]. Для каждого генератора должны быть указаны источники случайности и методы обработки данных от источников случайности. Генераторы, которые не являются компонентами ПСКЗИ, должны быть включены в список критических системных компонентов [СТ.1].

Требование СЧ.2 (1, 2). Для каждого генератора случайных чисел [СЧ.1] должна быть проведена оценка энтропии всех его источников случайности. Способ обработки данных от источников случайности должен гарантировать, что собранные данные содержат достаточно неопределенности для надежной генерации критического объекта.

Требование СЧ.3 (1). Если в генераторе случайных чисел [СЧ.1] отсутствуют физические источники случайности, то должно использоваться не менее двух альтернативных разнотипных источников.

Требование СЧ.4 (2). Каждый генератор случайных чисел [СЧ.1] должен обязательно использовать хотя бы один физический источник случайности.

Требование СЧ.5 (1, 2). Должна быть разработана и корректно реализована процедура тестирования выходных последовательностей генератора случайных чисел [СЧ.1], в котором используются физические источники случайности. Процедура должна быть направлена на выявление отказов, сбоев и изменений физических параметров при функционировании физических источников.

Требование СЧ.6 (1, 2). Выходные данные генератора случайных чисел [СЧ.1] должны являться результатом применения криптографических алгоритмов [КП.1] к данным от источников случайности, возможно дополненным обновляемым внутренним состоянием или предыдущими случайными числами. Применяемые криптографические алгоритмы должны обеспечивать сложные зависимости между выходными данными генератора и данными от каждого из источников случайности.

6 Функциональные требования безопасности к среде

6.1 Требования по идентификации и аутентификации (ИА)

Требование ИА.1 (1, 2). Каждому оператору должен быть назначен идентификатор и набор ролей [УД.1].

Требование ИА.2 (1, 2). Для каждого идентификатора оператора [ИА.1] должны быть определены аутентификационные данные. Среди аутентификационных данных должны быть выделены секреты аутентификации. Устройства ввода аутентификационных данных должны быть включены в список критических системных компонентов {СТ.1}.

Требование ИА.3 (1, 2). Должны быть определены и корректно реализованы средства аутентификации для проверки подлинности идентификатора оператора и возможности выполнения оператором сервисов соответствующих ролей [УД.3]. Средства аутентификации, реализуемые ПСКЗИ, должны быть включены в список сервисов {РС.1}.

Требование ИА.4 (2). Должно использоваться не менее двух факторов аутентификации.

Требование ИА.5 (1, 2). Вероятность пройти аутентификацию, не зная секретов аутентификации, не должна превышать 10^{-6}, если предпринимается одна попытка

аутентификации, и не должна превышать 10^{-5}, если предпринимаются попытки в течение 1 мин.

Требование ИА.6 (1, 2). Информация, которая отображается при вводе секретов аутентификации, не должна ослаблять стойкость средств аутентификации.

Требование ИА.7 (1, 2). Должны быть определены и реализованы средства проверки качества секретов аутентификации. Средства должны применяться при каждой установке или смене секрета.

Требование ИА.8 (1, 2). При реализации средств аутентификации в ПСКЗИ контрольные значения аутентификационных данных должны быть отнесены к открытым объектам {УД.2}. Контрольные значения секретов аутентификации должны быть отнесены к критическим объектам {УД.2}. Сеансовые объекты, которые содержат значения секретов аутентификации, также должны быть отнесены к критическим объектам {УД.2}.

6.2 Требования по настройке среды (НС)

Требование НС.1 (1, 2). Должны быть определены настройки среды эксплуатации для безопасной установки ПСКЗИ уполномоченным администратором.

Требование НС.2 (1, 2). Должны быть определены настройки среды эксплуатации для обеспечения целостности долговременных объектов [УД.2] при их хранении внутри криптографической границы. Выбранные настройки должны предотвращать изменение долговременных объектов вне сеансов между операторами и ПСКЗИ.

Требование НС.3 (1, 2). Должны быть определены настройки среды эксплуатации для обеспечения конфиденциальности и целостности сеансовых объектов [УД.2] и аутентификационных данных при их передаче и обработке в критических системных компонентах во время сеансов операторов.

7 Гарантийные требования безопасности

7.1 Требования по проектированию и разработке (ПР)

Требование ПР.1 (1, 2). Должно быть дано описание программ ПСКЗИ и установлено соответствие между функциональной спецификацией и средствами безопасности, реализованными в программах.

Требование ПР.2 (1, 2). В описании программ [ПР.1] должны быть определены все внешние интерфейсы ПСКЗИ.

Требование ПР.3 (2). В описании программ [ПР.1] должны быть определены внутренние компоненты ПСКЗИ и их интерфейсы.

Требование ПР.4 (1, 2). В описании программ [ПР.1] должны быть определены все используемые средства разработки и сборки программ. Должны быть перечислены все конфигурационные файлы, отвечающие за настройку средств разработки и сборки. Конфигурационные файлы должны быть включены в список элементов конфигурации {ЖЦ.2}.

Требование ПР.5 (1, 2). Исходные тексты программ должны быть снабжены комментариями, устанавливающими соответствие с описанием программ {ПР.1}.

Требование ПР.6 (1, 2). Программы должны быть написаны на высокоуровневых языках программирования. Вставки на низкоуровневых языках (языках ассемблера)

допускаются в случаях, критичных для производительности, а также тогда, когда высокоуровневые языки применить нельзя.

7.2 Требования по поддержке жизненного цикла (ЖЦ)

Требование ЖЦ.1 (1, 2). Должна быть определена и реализована система управления конфигурацией для ПСКЗИ. Система должна обеспечивать:

- контроль доступа разработчиков к элементам конфигурации;
- контроль версий элементов конфигурации;
- отслеживание изменений элементов конфигурации;
- сборку программ по исходным текстам [ПР.4].

Требование ЖЦ.2 (1, 2). В перечень элементов конфигурации должны быть включены:

- функциональная спецификация;
- программы;
- описание программ [ПР.1];
- исходные тексты программ;
- документация по управлению конфигурацией [ЖЦ.1];
- документация по поставке ПСКЗИ потребителю [ЖЦ.4];
- документация по устранению недостатков [ЖЦ.6] (только для класса 2);
- руководства [РД.1], [РД.2].

Требование ЖЦ.3 (1, 2). Каждая версия каждого элемента конфигурации должна быть снабжена уникальным идентификатором.

Требование ЖЦ.4 (1, 2). Должна быть определена и реализована система поставки ПСКЗИ потребителю.

Требование ЖЦ.5 (2). Должны быть предусмотрены средства контроля целостности и подлинности инсталляционных программ после их доставки потребителю.

Требование ЖЦ.6 (2). Должна быть определена и реализована система устранения недостатков в программах и документации ПСКЗИ. Система должна обеспечивать:

- регистрацию недостатков;
- определение порядка выявления причин недостатков и исправления недостатков;
- отслеживание статуса недостатков (подтвержден, исправляется, исправлен и др.);
- описание способа устранения недостатков;
- порядок извещения потребителей об устранении недостатков.

7.3 Требования к руководствам (РД)

Требование РД.1 (1, 2). Должно быть разработано руководство администратора. Руководство должно описывать:

- обязанности администратора по настройке среды [НС.1], [НС.2], [НС.3];
- инструкции по установке ПСКЗИ [НС.1];
- доступные администратору сервисы [УД.3];
- обязанности администратора по настройке средств безопасности ПСКЗИ;
- связанные с безопасностью предположения относительно поведения операторов.

Требование РД.2 (1, 2). Для каждой роли [УД.1], отличной от роли «Администраторы», должно быть разработано руководство ее операторов. Руководство должно определять:

– доступные оператору сервисы [УД.3];
– обязанности оператора по обеспечению безопасности ПСКЗИ.

Требование РД.3 (2). Руководства [РД.1], [РД.2] должны описывать типичные ошибки операторов, которые могут привести к снижению безопасности ПСКЗИ. Руководства должны давать рекомендации операторам по избежанию ошибок.

7.4 Требования по программе испытаний (ПИ)

Требование ПИ.1 (1, 2). Должна быть разработана программа испытаний ПСКЗИ разработчиком. Программа должна определять:
– планы тестирования;
– содержание тестов;
– ожидаемые результаты выполнения тестов;
– фактические результаты выполнения тестов.

Требование ПИ.2 (1, 2). Тесты программы испытаний [ПИ.1] должны покрывать все функциональные возможности ПСКЗИ, определенные в функциональной спецификации.

Требование ПИ.3 (2). Тесты программы испытаний [ПИ.1] должны покрывать функциональные возможности всех компонентов ПСКЗИ, определенных в описании программ [ПР.1].

Приложение А
(рекомендуемое)
Содержание функциональной спецификации

1 Описание ПСКЗИ.
   1) Назначение.
   2) Класс (1 или 2).
   3) Основные функциональные возможности.
   4) Криптографическая граница.
   5) Список критических системных компонентов [СТ.1].

2 Криптографическая поддержка.
   1) Список криптографических алгоритмов и протоколов [КП.1].
   2) Методы генерации долговременных параметров и ключей [КП.3].
   3) Средства контроля времени выполнения криптографических алгоритмов и протоколов [КП.4] (только для класса 2).

3 Реализация сервисов.
   1) Список сервисов [РС.1].
   2) Средства защиты от создания неявных копий критических объектов [РС.4] (только для класса 2).

4 Управление доступом.
   1) Список ролей операторов [УД.1].
   2) Список объектов [УД.2].
   3) Описание политики управления доступом [УД.3].
   4) Состояния системного сеанса и правила перехода между состояниями [УД.4].

5 Защита объектов.
   1) Криптографические методы обеспечения конфиденциальности [ЗО.3].
   2) Криптографические методы контроля целостности [ЗО.4].
   3) Аппаратные методы защиты [ЗО.5].
   4) Методы разделения секрета [ЗО.6].
   5) Алгоритмические методы контроля целостности [ЗО.7].
   6) Организационные методы обеспечения конфиденциальности [ЗО.8].
   7) Соответствие «объекты — методы защиты» [ЗО.1], [ЗО.2].
   8) Методы очистки критических сеансовых объектов [ЗО.12].

6 Самотестирование.
   1) Проверка работоспособности критических системных компонентов [СТ.2].
   2) Перечень проверок самотестирования [СТ.3].
   3) Обработка ошибок тестирования [СТ.5].

7 Генерация случайных чисел.
   1) Описание генераторов случайных чисел [СЧ.1].
   2) Оценка энтропии источников случайности [СЧ.2].
   3) Тестирование выходных последовательностей генераторов [СЧ.5].

8 Идентификация и аутентификация.
   1) Методы идентификации операторов [ИА.1].

2) Методы аутентификации операторов [ИА.2], [ИА.3].
3) Проверка качества секретов аутентификации [ИА.7].

9 Настройка среды.
1) Настройка среды для безопасной установки [НС.1].
2) Настройка среды для защиты системных объектов [НС.2].
3) Настройка среды для защиты сеансов [НС.3].

10 Гарантийные меры (могут быть определены в отдельных документах).
1) Описание программ [ПР.1].
2) Внешние интерфейсы программ [ПР.2].
3) Внутренние компоненты и интерфейсы [ПР.3] (только для класса 2).
4) Средства разработки [ПР.4].
5) Языки программирования [ПР.6].
6) Система управления конфигурацией [ЖЦ.1].
7) Система поставки ПСКЗИ потребителю [ЖЦ.4].
8) Методы контроля инсталляционных программ [ЖЦ.5] (только для класса 2).
9) Список руководств [РД.1], [РД.2].
10) Типичные ошибки операторов [РД.3] (только для класса 2).
11) Программа испытаний [ПИ.1].
12) Результаты анализа покрытия тестами [ПИ.2].
13) Результаты анализа глубины тестирования [ПИ.3] (только для класса 2).

Допускается объединять разделы. Если в ПСКЗИ не реализован необязательный механизм безопасности, то соответствующий раздел можно опустить.

Приложение Б
(справочное)
Программное средство «Криптодиск» (примерная спецификация)

Б.1 Описание

Назначение. Программное средство «Криптодиск» предназначено для шифрования файлов на магнитных носителях «на лету». С помощью программного средства пользователи могут создавать частные каталоги, при записи в которые файлы будут зашифровываться, а при чтении — расшифровываться. Сам каталог без обработки программным средством представляет собой обычный файл операционной системы. Пользователи, которым запрещено чтение каталога, не могут раскрыть его содержание даже при доступе к данному файлу.

Класс. «Криптодиск» является средством класса 2.

Основные функциональные возможности. «Криптодиск» реализует:

– управление частными каталогами нескольких пользователей;
– шифрование файлов каталога «на лету»;
– хранение ключей на USB-токенах;
– возможность записи файлов в каталог любым авторизованным пользователем;
– установку разрешений на доступ к отдельным файлам каталога совместно администратору и выбранному доверенному пользователю (на случай утери или блокировки токена владельца).

Криптографическая граница. «Криптодиск» выполняется на персональном компьютере. Криптографической границей является периметр системного блока компьютера.

Критические системные компоненты. В пределах криптографической границы находятся следующие критические системные компоненты:

`CSC.Board` *Вычислительная платформа*

Процессор, материнская плата, устройства хранения, порты ввода/вывода и другие аппаратные компоненты персонального компьютера, необходимые для выполнения программ и хранения данных «Криптодиск». Должны быть доступны два порта для USB-токенов. Должен присутствовать высокоточный таймер.

`CSC.OS` *Операционная система*

Компоненты операционной системы, отвечающие за безопасное выполнение программ «Криптодиск». Должна использоваться операционная система линейки Windows.

Б.2 Криптографическая поддержка

Криптографические алгоритмы. В «Криптодиск» реализованы следующие криптографические алгоритмы:

`Alg.DataWrap` *Алгоритмы одновременного шифрования и имитозащиты*

Алгоритмы одновременного шифрования и имитозащиты, определенные в ТНПА-1. Реализованы алгоритм установки защиты (зашифрование и вычисление имитовставки) и

алгоритм снятия защиты (проверка имитовставки и расшифрование). Алгоритмы предназначены для защиты файлов в частных каталогах.

`Alg.Transport` *Алгоритмы транспорта ключей*

Алгоритмы транспорта ключей, определенные в ТНПА-2. Реализованы алгоритм установки защиты транспортируемого ключа и алгоритм снятия защиты. При установке защиты используется открытый ключ оператора, которому транспортируется ключ, а при снятии защиты используется личный ключ того же оператора. Алгоритмы предназначены для транспорта ключей `Alg.DataWrap`.

`Alg.Sign` *Алгоритмы ЭЦП*

Алгоритмы электронной цифровой подписи, определенные в ТНПА-3. Реализованы алгоритмы выработки и проверки ЭЦП. Используется администратором для заверения открытых ключей `Alg.Transport` и пользователями для проверки заверения.

`Alg.PRNG` *Алгоритм генерации псевдослучайных чисел*

Алгоритм генерации псевдослучайных чисел с секретным параметром, определенный в ТНПА-4. Алгоритм предначен для генерации ключей `Alg.DataWrap` и `Alg.Transport`, других случайных параметров криптографических алгоритмов.

`Alg.Hash` *Алгоритм хэширования*

Алгоритм хэширования, определенный в ТНПА-5. Алгоритм предначен для вычисления контрольных характеристик системных объектов. Является композиционным элементом `Alg.Transport` и `Alg.Sign`. Используется в генераторе случайных чисел.

Генерация ключей и параметров. Алгоритмы `Alg.DataWrap`, `Alg.PRNG` и `Alg.Hash` не имеют долговременных параметров. Долговременные параметры `Alg.Transport` и `Alg.Sign` совпадают. К ним относятся параметры эллиптической кривой и базовая точка на ней. Эти параметры генерируются по алгоритму, определенному в ТНПА-3, фиксируются в программах «Криптодиск» и используются всеми операторами.

Ключ `Alg.PRNG` вырабатывается с помощью генератора случайных чисел (Б.7). После создания ключ записывается на токен вместе с нулевым счетчиком. Дальнейшая работа с `Alg.PRNG` регулируется следующими правилами:

1 При обращении к `Alg.PRNG` ключ и счетчик читаются с токена. По ним вырабатываются псевдослучайные данные требуемого объема.

2 При выработке данных счетчик изменяется. Новое значение счетчика записывается на токен.

3 Если запись прошла успешно, то сформированные псевдослучайные данные используются по назначению. В противном случае данные уничтожаются и обращение к `Alg.PRNG` завершается ошибкой.

По данным правилам вырабатываются секретные ключи и синхропосылки `Alg.DataWrap`, личные ключи и секретные параметры `Alg.Transport` и `Alg.Sign`, частичные секреты и др.

Открытые ключи `Alg.Transport` и `Alg.Sign` вырабатываются по личному ключу в соответствии с алгоритмами, определенными в ТНПА-2, ТНПА-3.

Контроль времени выполнения. Алгоритмы `Alg.DataWrap` и `Alg.PRNG` относятся к классу симметричных. Время их выполнения не зависит от значений ключей.

Алгоритм `Alg.Hash` является бесключевым. При хэшировании объектов, которые могут оказаться критическими, время хэширования не зависит от значения объекта и определяется только размером объекта.

В алгоритмах `Alg.Transport`, `Alg.Sign` определяются кратные $uP = P + P + \ldots + P$, где P — точка эллиптической кривой, которая суммируется u раз. При этом кратность u может являться личным ключом или одноразовым секретным параметром. Для определения кратных точек используется алгоритм «лестница Монтгомери» (Montgomery ladder), время работы которого не зависит от кратности.

Б.3 Реализация сервисов

Сервисы. «Криптодиск» реализует следующие сервисы:

`S.GetVersion` *Номер версии*

Сервис отображает номер версии программ в формате «старший номер, младший номер, номер сборки».

`S.SelfTest` *Самотестирование*

Сервис самотестирования.

`S.InitToken` *Инициализация токена*

На вход сервиса подается идентификатор оператора, которому токен передается во владение. Сервис настраивает файловую систему токена, записывает на токен идентификатор и служебные данные. Если оператор не является администратором, то дополнительно используется уже инициализированный токен администратора, с которого переписывается ключ `R.SignKeyPublic`.

`S.GenUserKeys` *Генерация ключей пользователя*

Сервис запрашивает PIN-код доступа и задает его для токена. После этого для доступа к объектам токена требуется предъявить PIN-код. Сервис генерирует ключи `R.KeyPRNG`, формирует нулевой счетчик `R.CounterPRNG` и записывает эти объекты на токен. Затем генерируются и записываются на токен ключи `R.TransportKeyPrivate` и `R.TransportKeyPublic`. Ключ `R.TransportKeyPublic` дополнительно записывается в каталог администратора вместе с идентификатором владельца токена.

`S.GenAdminKeys` *Генерация ключей администратора*

Сервис выполняет те же действия, что и `S.GenUserKeys`. Дополнительно генерируются и записываются на токен ключи `R.SignKeyPrivate` и `R.SignKeyPublic`. Дополнительно вызывается сервис `S.SignTransportKey`, на вход которого подаются ключ `R.TransportKeyPublic` и идентификатор администратора.

`S.AuthToken` *Аутентификация для доступа к токену*

На вход сервиса подается PIN-код оператора. PIN-код передается управляющей программе токена. После подтверждения PIN-кода оператору разрешается доступ к объектам токена.

`S.SignTransportKey` *Подпись ключа транспорта*

На вход сервиса подается запись, содержащая ключ `R.TransportKeyPublic` и идентификатор его владельца. Используется токен администратора. Сервис читает с токена ключ `R.SignKeyPrivate`, вычисляет на нем ЭЦП записи и добавляет запись вместе с ЭЦП в общедоступный справочник открытых ключей.

`S.WriteFile` *Запись файла*

На вход сервиса подается файл, который пользователю U требуется записать в каталог пользователя H. Используется токен U. Из справочника открытых ключей читается запись, содержащая идентификатор H и его ключ `R.TransportKeyPublic`. ЭЦП этой записи проверяется на ключе `R.SignKeyPublic`, который читается с токена U. Если проверка прошла успешно, то генерируется ключ `R.DataKey` и на нем устанавливается защита входного файла. Ключ `R.DataKey` защищается на ключе `R.TransportKeyPublic` пользователя U и сохраняется вместе с зашифрованным файлом в каталоге.

`S.ReadFile` *Чтение файла*

На вход сервиса подается зашифрованный файл из частного каталога и зашифрованный ключ `R.DataKey` его защиты. Используется токен владельца каталога. С токена читается ключ `R.TransportKeyPrivate` и на нем снимается защита с ключа `R.DataKey`. Затем на ключе `R.DataKey` снимается защита с зашифрованного файла.

`S.ShareFile` *Открытие доступа к файлу*

На вход сервиса подается зашифрованный ключ `R.DataKey`, который хранится вместе с зашифрованным файлом в частном каталоге, и идентификатор доверенного пользователя, которому совместно с администратором предоставляется доступ к файлу. Используется токен владельца каталога. С токена читается ключ `R.TransportKeyPrivate` и на нем снимается защита с ключа `R.DataKey`. Затем ключ `R.DataKey` разбивается на два частичных секрета `R.DataKeyPartial`. Каждый из секретов защищается на ключах `R.TransportKeyPublic` доверенного пользователя и администратора. Ключ `R.TransportKeyPublic` доверенного пользователя читается из общедоступного справочника. Перед использованием проверяется его ЭЦП. Зашифрованные частичные секреты сохраняются вместе с файлом в каталоге.

`S.RecoverFile` *Восстановление файла*

На вход сервиса подается зашифрованный файл из частного каталога и зашифрованные частичные секреты `R.DataKeyPartial` пользователя и администратора, которым совместно открыт доступ к файлу в частном каталоге. Используются токены доверенного пользователя и администратора. С токенов читаются ключи `R.TransportKeyPrivate`. На них снимается защита с `R.DataKeyPartial`. По найденным частичным секретам собирается ключ `R.DataKey`, на котором снимается защита с целевого файла.

Все сервисы возвращают код ошибки или признак успешного завершения.

Сервисы `S.InitToken`, `S.GenAdminKeys`, `S.GenUserKeys`, `S.SignTransportKey` обслуживают управление ключами и выполняются в следующей последовательности:

1 Администратор генерирует свои ключи, вызывая `S.InitToken` и `S.GenAdminKeys`.

2 По запросу пользователя администратор инициализирует для него токен, вызывая `S.InitToken`. Токен передается пользователю.

3 Пользователь генерирует ключи, вызывая `S.GenUserKeys`. Открытый ключ `R.TransportKeyPublic` помещается в каталог администратора.

4 Администратор с определенной периодичностью просматривает свой каталог и подписывает новые записи (идентификатор пользователя, открытый ключ пользователя), вызывая `S.SignTransportKey`. При необходимости перед вызовом сервиса администратор проверяет соответствие между идентификатором и открытым ключом, связываясь с пользователем.

Защита от создания неявных копий критических объектов. Неявные копии критических объектов могут создаваться в следующих областях памяти:

– файл подкачки (`pagefile.sys`);

– файл спящего режима (`hiberfile.sys`);

– отладочные дампы памяти, сохраняемые операционной системой при сбоях в программах.

В «Криптодиск» используется специальные функции выделения оперативной памяти, которые блокируют попадание критических сеансовых объектов в файл подкачки. Дополнительно при настройке среды блокируются спящий режим и средства создания отладочных дампов памяти.

Б.4 Управление доступом

Роли. «Криптодиск» поддерживает роли `Role.Admins` («Администраторы») и `Role.Users` («Пользователи»). Группа `Role.Admins` включает только одного участника.

Объекты. Обрабатываются следующие объекты:

`R.File` *Файл*

Файл частного каталога. Долговременный критический объект. Принадлежит владельцу каталога.

`R.KeyPRNG` *Секретный ключ ГПСЧ*

Секретный ключ алгоритма `Alg.PRNG`. Вырабатывается с помощью генератора случайных чисел. Хранится на токене. Принадлежит владельцу токена.

`R.CounterPRNG` *Счетчик ГПСЧ*

Счетчик алгоритма `Alg.PRNG`. Открытый объект. Первоначально устанавливается в 0, при обращениях к `Alg.PRNG` последовательно увеличивается. Хранится на токене. Принадлежит владельцу токена.

`R.DataKey` *Ключ защиты данных*

Секретный ключ алгоритмов `Alg.DataWrap` для защиты `R.File`. Генерируется с помощью `Alg.PRNG`. Хранится вместе с защищенным файлом. Принадлежит владельцу `R.File`.

`R.DataKeyPartial` *Частичный ключ защиты данных*

Секретный частичный ключ алгоритмов `Alg.DataWrap`. Владельцем является администратор или один из пользователей. Частичный ключ администратора генерируется с помощью `Alg.PRNG`. Частичный ключ пользователя определяется как сумма `R.DataKey` и частичного ключа администратора. Хранится вместе с защищенным файлом.

`R.SignKeyPrivate` *Личный ключ ЭЦП*

Личный ключ алгоритма выработки ЭЦП `Alg.Sign`. Генерируются с помощью `Alg.PRNG`. Хранится на токене администратора. Принадлежит администратору.

`R.SignKeyPublic` *Открытый ключ ЭЦП*

Открытый ключ алгоритма проверки ЭЦП `Alg.Sign`. Генерируется по личному ключу `R.SignKeyPrivate`. Хранится на токене администратора и на токенах пользователей. Принадлежит администратору.

`R.TransportKeyPrivate` *Личный ключ транспорта*

Личный ключ алгоритма `Alg.Transport`. Генерируется с помощью `Alg.PRNG`. Хранится на токене. Принадлежит владельцу токена.

`R.TransportKeyPublic` *Открытый ключ транспорта*

Открытый ключ алгоритма `Alg.Transport`. Генерируется по личному ключу `R.TransportKeyPrivate`. Хранится на токене и в общедоступном справочнике открытых ключей. Принадлежит владельцу токена.

`R.System` *Системные объекты*

Программы, файлы настроек, общедоступный справочник открытых ключей, данные для тестирования криптографических алгоритмов, флаг блокировки. Являются открытыми объектами. Хранятся в пределах криптографической границы.

Политика управления доступом. Политика управления доступом определена в таблице Б.1. В таблице операции над объектами обозначаются следующим образом: `X` — выполнение, `C` — создание, `W` — запись, `R` — чтение. Владельцами объектов, над которыми выполняются операции, в основном являются операторы, вызывающие сервисы. Исключение составляют системные объекты, а также объекты, снабженные надстрочными символами: A — объект администратора, U — объект другого пользователя, H — объект владельца каталога.

Состояния. Главная программа «Криптодиск» реализует службу операционной системы Windows. Выполнение службы начинается при запуске системы и заканчивается при ее завершении. Системный сеанс «Криптодиск» соответствует периоду выполнения службы.

Используются следующие состояния системного сеанса и правила перехода между состояниями:

Таблица Б.1 — Политика управления доступом «Криптодиск»

Операторы и сервисы	Объекты	Операции
Role.Admins,	S.GetVersion	X
Role.Users	S.SelfTest	X
	S.GenUserKeys	X
	S.AuthToken	X
	S.ReadFile	X
	S.RecoverFile	X
Role.Admins	S.InitToken	X
	S.GenAdminKeys	X
	S.SignTransportKey	X
Role.Users	S.WriteFile	X
	S.ShareFile	X
S.GetVersion	R.System (данные о версии)	R
S.SelfTest	R.System (тестовые данные)	R
S.InitToken	R.SignKeyPublic	W
S.GenUserKeys	R.KeyPRNG	CW
	R.CounterPRNG	CW
	R.TransportKeyPrivate	CW
	R.TransportKeyPublic	CW
S.GenAdminKeys	S.GenUserKeys	X
	R.SignKeyPrivate	CW
	R.SignKeyPublic	CW
	S.SignTransportKey	X
S.AuthToken	–	
S.SignTransportKey	R.TransportKeyPublic^{U}	RW
	R.SignKeyPrivate	R
S.WriteFile	R.TransportKeyPublic^{H}	R
	R.SignKeyPublic^{A}	R
	R.DataKey^{H}	CW
	R.File^{H}	W
S.ReadFile	R.File	R
	R.DataKey	R
	R.TransportKeyPrivate	R
S.ShareFile	R.DataKey	R
	R.TransportKeyPrivate	R
	R.DataKeyPartial^{UA}	CW
	R.TransportKeyPublic^{UA}	R
	R.SignKeyPublic^{A}	R
S.RecoverFile	R.DataKeyPartial	R
	R.TransportKeyPrivate	R
	R.DataKey^{H}	C
	R.File^{H}	R

`State.Start` *Загрузка*

Состояние загрузки. При загрузке проверяется флаг блокировки, который хранится в защищенном каталоге (изменение флага разрешено только системе и администраторам). Если флаг установлен, то выполняется переход в состояние `State.Lock`, иначе — самотестирование. Если во время тестирования произошла ошибка, то снова выполняется переход в `State.Lock`. При успешном тестировании разрешается выполнить сервис `S.AuthToken`. При успешной аутентификации выполняется переход в состояние `State.Admin` или `State.User`.

`State.Admin` *Сеанс администратора*

Состояние, которое соответствует сеансу администратора. Разрешено выполнять сервисы администратора. При ошибках и сбоях выполняется переход в состояние `State.Lock`.

`State.User` *Сеанс пользователя*

Состояние, которое соответствует сеансу пользователя. Разрешено выполнять сервисы пользователя. При ошибках и сбоях выполняется переход в состояние `State.Lock`.

`State.Lock` *Блокировка*

Состояние блокировки, из которого ПСКЗИ может быть выведен администратором. При блокировке завершаются все сервисы, закрываются файлы, устанавливается флаг блокировки. Разрешено выполнять сервис аутентификации `S.AuthToken` на роль `Role.Admins`. При успешной аутентификации выполняется переход в состояние `State.Admin`.

Б.5 Защита объектов

Криптографические методы защиты. Для обеспечения конфиденциальности и контроля целостности `R.File` применяется алгоритм `Alg.DataWrap`. Используются ключи `R.DataKey`, свои для каждого `R.File`. Синхропосылки алгоритма выбираются случайно с помощью алгоритма `Alg.PRNG` и сохраняются вместе с защищенным `R.File`.

Для обеспечения конфиденциальности и контроля целостности ключей `R.DataKey` и частичных секретов `R.DataKeyPartial` применяются алгоритмы `Alg.Transport`. При установке защиты используются ключи `R.TransportKeyPublic`, при снятии защиты — ключи `R.TransportKeyPrivate`.

Для контроля целостности ключей `R.TransportKeyPublic`, размещенных в общедоступном справочнике, применяются алгоритмы ЭЦП `Alg.Sign`. При выработке ЭЦП используется личный ключ администратора `R.SignKeyPrivate`. При проверке ЭЦП используется открытый ключ `R.SignKeyPublic`, размещенный на токенах пользователей.

Аппаратные методы защиты. Аппаратные методы защиты реализуются применением USB-токенов, удовлетворяющих ТНПА-6. В защищенной памяти токена размещаются следующие ключи его владельца: `R.CounterPRNG`, `R.KeyPRNG`, `R.TransportKeyPrivate`, `R.TransportKeyPublic`. Дополнительно на токене администратора размещаются его ключи `R.SignKeyPrivate`, `R.SignKeyPublic`, а на токенах пользователей — ключ `R.SignKeyPublic` администратора.

Доступ к ключам регулируется управляющей программой токена. Прежде чем получить доступ, требуется пройти аутентификацию с помощью сервиса `S.AuthToken`.

Методы разделения секрета. Ключ `R.DataKey` разделяется на два частичных секрета `R.DataKeyPartial`, которые передаются администратору и доверенному пользователю. Ключ и частичные секреты являются двоичными строками одинаковой длины. Частичный секрет администратора выбирается случайно с помощью алгоритма `Alg.PRNG`. Второй частичный секрет определяется как сумма (поразрядная по модулю 2) первого с `R.DataKey`.

Алгоритмические методы контроля целостности. Для контроля `R.System` используются контрольные хэш-значения, которые фиксируются в программах «Криптодиск». Хэш-значения вычисляются по алгоритму `Alg.Hash`.

Очистка критических сеансовых объектов. Критические сеансовые объекты размещаются в оперативной памяти компьютера. Очистка состоит в обнулении соответствующих областей памяти. Очистка выполняется после использования объекта или при возникновении исключительной ситуации в ходе выполнения программ.

Б.6 Самотестирование

Работоспособность критических системных компонентов. При инсталляции «Криптодиск» проверяется, что

– используется одна из следующих операционных систем: Windows Vista, Windows 7, Windows Server 2008, Windows Server 2008 R2;

– процессор имеет 64-разрядный регистр-таймер `TSC` (time stamp counter), содержимое которого увеличивается на каждом такте работы;

– частота процессора не ниже 600 МГц.

При каждом запуске «Криптодиск» в состоянии `State.Start` проверяется, что

– спящий режим (hibernation) отключен;

– средства создания отладочных дампов памяти отключены.

При чтении данных с токенов проверяется работоспособность USB-портов.

Самотестирование. В состоянии `State.Start` и по запросу оператора выполняются следующие проверки:

– тесты известного ответа для `Alg.DataWrap`, `Alg.Transport`, `Alg.Sign`, `Alg.PRNG`, `Alg.Hash`;

– тесты прямого и обратного преобразований для `Alg.DataWrap`, `Alg.Transport`, `Alg.Sign`;

– тесты на соответствие между `R.TransportKeyPrivate` и `R.TransportKeyPublic`, между `R.SignKeyPrivate` и `R.SignKeyPublic`.

Перед генерацией личных и секретных ключей проводится тестирование выходных последовательностей генератора случайных чисел.

Ошибки самотестирования. При ошибках самотестирования выполняется переход в состояние `State.Lock`.

Б.7 Генерация случайных чисел

Генератор случайных чисел. Используется генератор случайных чисел с двумя источниками случайности. Первым источником являются временные интервалы между нажатиями оператором клавиш на клавиатуре (клавиатурный источник). Вторым источником является тепловый шум в аналоговых цепях токена (физический источник).

При нажатии клавиш фиксируются значения регистра TSC. Разность между значениями регистра сохраняется, если друг за другом нажаты две различные клавиши и интервал между нажатиями более 50 мс. Всего сохраняется 128 разностей. Из них составляется 1024-битовая строка.

Микроконтроллер токена оцифровывает тепловой шум и формирует 16-битовое случайное слово. По 16 обращениям к микроконтроллеру формируется 256-битовая строка.

Строки, полученные от двух источников случайности, объединяются и хэшируются. Полученное хэш-значение считается выходом генератора.

Оценка энтропии. Были проведены вычислительные эксперименты, направленные на оценку энтропии выходных последовательностей клавиатурного источника. Для этого было сформировано 25 наборов, каждый из которых включал 40 последовательностей, полученных различными пользователями на различных компьютерах.

Было установлено, что ни в одном из наборов наблюдения не повторяются. Данный факт можно объяснить высокой частотой обновления регистра TSC, несоизмеримой с частотой нажатия оператором на клавиши. Пусть $X_{(1)}, X_{(2)}, \ldots, X_{(d)}$ — наблюдения набора X, упорядоченные по возрастанию ($d = 128 \cdot 40$). В силу неповторяемости $X_{(i)}$ величина $h = \log_2(X_{(d)} - X_{(1)})$ является адекватной оценкой удельной энтропии на наблюдение для источника случайности, выдавшего X. Для учета редких длительных пауз между нажатиями на клавиши при расчетах использовалась уточненная оценка $h^* = \log_2(X_{(3d/4)} - X_{(1)})$, заведомо меньшая h.

В проведенных экспериментах величина h^* была не меньше $27,1$. Нижняя граница достигалась для процессоров с минимально допустимой тактовой частотой 600 МГц. Полученные результаты дают основание считать, что энтропия выходной последовательности клавиатурного источника не меньше 256, что достаточно для надежной генерации 256-битового ключа даже при отказе физического источника.

Оценка энтропии физического источника проведена разработчиком микроконтроллера токена. Установлено, что источник выдает случайные равновероятные независимые слова. Поэтому энтропия выходных 256-битовых строк источника оценена максимальным значением — 256.

Тестирование выходных последовательностей. Для проверки работоспособности физического источника случайности используются статистические тесты американского стандарта FIPS PUB 140-2 Security Requirements for Cryptographic Modules. Тестируется двоичная последовательность длины 20000. Тесты имеют следуюший вид:

1 *Тест знаков.* Определяется величина S — число единиц в последовательности. Тест пройден, если $9725 < S < 10275$.

2 *Покер-тест.* Последовательность разбивается на 5000 тетрад. Тетрады интерпретируется как числа от 0 до 15. Определяется статистика $S = 16 \sum_{i=0}^{15} S_i^2 - (5000)^2$, где S_i — количество появлений числа i среди тетрад. Тест пройден, если $10800 < S < 230850$.

3 *Тест серий.* Определяются серии (максимальные последовательности повторяющихся соседних битов) различных длин. Тест пройден, если и для серий из нулей, и для серий из единиц выполняется: $S_1 \in [2315, 2685]$, $S_2 \in [1114, 1386]$, $S_3 \in [527, 723]$, $S_4 \in [240, 384]$, $S_5, S_{6+} \in [103, 209]$. Здесь S_i — количество серий длины $i = 1, 2, \ldots$, $S_{6+} = S_6 + S_7 + \ldots$.

4 *Тест длинных серий.* Тест пройден, если в последовательности отсутствуют серии длины 26 и больше.

Пороги тестов выбраны так, что вероятность ошибки первого рода (ложной тревоги) равняется $0,0001$. Это означает, что даже при нормальной работе микроконтроллера в среднем одна из 10000 его выходных последовательностей будет забракована. Возможность ложной тревоги должна учитываться при анализе причин блокировки «Криптодиск» во время тестирования.

Б.8 Идентификация и аутентификация

Идентификация операторов. Идентификация и аутентификация операторов выполняются средствами операционной системы и токена.

Администратору назначается идентификатор `admin`. Идентификаторы пользователям назначает сам администратор. Операционная система поддерживает уникальность идентификаторов. Идентификаторы пользователей переписываются на их токены в сервисе `S.InitToken`. В сервисе `S.AuthToken` идентификатор оператора сравнивается с идентификатором на токене.

Аутентификация операторов. Для успешной аутентификации требуется предъявить пароль операционной системы, токен и PIN-код доступа к токену. Пароль и PIN-код вводятся в специальном диалоговом окне. Введенные символы маскируются.

Используются сильные пароли Windows, которые состоят не менее чем из 7 символов, обязательно содержат буквы в верхнем регистре, буквы в нижнем регистре, цифры и специальные символы (знаки пунктуации, скобки, знаки арифметических операций). PIN-код состоит из 6 десятичных цифр.

Если при аутентификации операционной системой оператор трижды вводит неверный пароль, то компьютер блокируется на 1 мин. Для этого администратор настраивает политику блокировки учетных записей пользователей операционной системы. Если оператор трижды вводит неверный PIN-код, то токен блокируется навсегда, его содержимое очищается.

Качество секретов аутентификации. Для проверки качества паролей администратор настраивает политику управления паролями операционной системы.

Управляющая программа токена не позволяет задавать PIN-коды, длина которых отлична от 6. Настройка программы не требуется.

Б.9 Настройка среды

Безопасная установка. Перед установкой «Криптодиск» администратор настраивает группы операционной системы. Группа Administrators соответствует роли `Role.Admins`, группа Users — роли `Role.Users`.

Администратор устанавливает защиту от создания неявных копий критических объектов (Б.2), настраивает политику управления паролями и политику блокировки учетных записей пользователей (Б.8). Групповая политика настраивается таким образом, что установка программ, включая «Криптодиск», разрешена только членам группы Administrators. Дополнительно администратор настраивает средства защиты от вредоносных программ.

После этого администратор устанавливает «Криптодиск».

Защита системных объектов. Файлы программ и настроек, общедоступный справочник открытых ключей, флаг блокировки хранятся в специальном каталоге операци-

онной системы. Пользователям (группа Users) запрещается изменять содержимое этого каталога.

Защита сеансов. Защита сеансовых объектов и аутентификационных данных выполняется ядром операционной системы. Настройка не требуется.

Б.10 Гарантийные меры

Проектирование. Разрабатывается документ «Описание программы». Документ описывает функции, реализующие сервисы ПСКЗИ, соответствующие структуры данных и коды ошибок. Описываются состояния программы, форматы хранения объектов. Устанавливается соответствие с функциональной спецификацией.

Криптографические алгоритмы реализует библиотека «Криптоядро». Документ «Описание программы» определяет интерфейсы этой библиотеки.

Разработка. Программы разрабатываются на языке C++ в среде Microsoft Visual Studio 2010. Программы библиотеки «Криптоядро» разрабатываются на языке C.

Управление конфигурацией. Для управления исходными текстами программ используется система Subversion. Система обеспечивает контроль версий модулей программ, управление доступом к исходным текстам для нескольких разработчиков, управление сборкой программ.

Документы снабжаются уникальными идентификаторами по правилам ЕСПД. Идентификатор включает номер версии документа. Номер версии увеличивается при внесении в документ изменений.

Разрабатывается документ «Поддержка жизненного цикла», в котором описываются правила работы с Subversion и правила управления документами.

Поставка. Инсталляционная программа «Криптодиск» размещается в Интернет на сайте разработчика. В инсталляционную программу по технологии Authenticode добавляется ЭЦП разработчика. В руководстве администратора описывается процесс проверки ЭЦП перед установкой «Криптодиск».

Устранение недостатков. Используется система отслеживания ошибок Bugzilla, которая реализует все необходимые функции по устранению недостатков. В документе «Поддержка жизненного цикла» определяются правила работы с Bugzilla.

Руководства. Разрабатываются документы «Руководство администратора» и «Руководство пользователя». Документы содержат раздел «Типичные ошибки».

Программа испытаний. Разрабатывается система тестов «Криптодиск». Планы тестирования ориентированы на проверку цепочек выполнения сервисов. Дополнительно тестируются функции криптографической библиотеки «Криптоядро».

Тесты описываются в документе «Программа и методика испытаний». Документ включает приложения «Анализ покрытия тестами» и «Анализ глубины тестирования».

Поправка к официальной редакции

В каком месте	Напечатано	Должно быть
Приложение Б, подраздел Б.2, 3-й абзац с конца	Алгоритмы `Alg.DataWrap` и `Alg.Sign` относятся к классу симметричных.	Алгоритмы `Alg.DataWrap` и `Alg.PRNG` относятся к классу симметричных.
Приложение Б, подраздел Б.7, 3-й абзац с конца	Определяются серии (последовательности одинаковых символов) различных длин. Пусть S_i — количество серий длины $i = 1, 2, \ldots$, и $S_{6+} = S_6 + S_7 + \ldots$. Тест пройден, если $S_1 \in [2315, 2685]$, $S_2 \in [1114, 1386]$, $S_3 \in [527, 723]$, $S_4 \in [103, 209]$, $S_5, S_{6+} \in [103, 209]$.	Определяются серии (максимальные последовательности повторяющихся соседних битов) различных длин. Тест пройден, если и для серий из нулей, и для серий из единиц выполняется: $S_1 \in [2315, 2685]$, $S_2 \in [1114, 1386]$, $S_3 \in [527, 723]$, $S_4 \in [240, 384]$, $S_5, S_{6+} \in [103, 209]$. Здесь S_i — количество серий длины $i = 1, 2, \ldots$, $S_{6+} = S_6 + S_7 + \ldots$.