Описание функциональных характеристик и информации для установки и эксплуатации

Библиотека подпрограмм операционной системы смарт-карты для однокристальных микроконтроллеров MST-JVM

Оглавление

Введение

Настоящий документ содержит описание функциональных характеристик, а также информацию, необходимую для установки и эксплуатации программы для ЭВМ «Библиотека подпрограмм операционной системы смарт-карты для однокристальных микроконтроллеров MST-JVM» (далее — MST-JVM, библиотека).

Аббревиатуры

- APDU — Application Protocol Data Unit
- API — Application Programming Interface
- ARM — Advanced RISC Machines
- BSP — Basic Secure Platform
- CBC — Cipher Block Chaining
- CVM — Cardholder Verification Method
- DAP — Data Authentication Pattern
- DES — Data Encryption Standard
- ECB — Electronic Codebook
- ECC — Elliptic-curve cryptography
- HAL — Hardware Abstraction Layer
- ISD — Issuer Security Domain
- GP — GlobalPlatform
- JC — Java Card
- JCRE — Java Card Runtime Environment
- JCVM — Java Card Virtual Machine
- KVN — Key Version Number
- MAC — Message authentication code
- NVM — Non-volatile memory
- PKI — Public key infrastructure
- RAM — Random-access memory
- RSA — Rivest–Shamir–Adleman
- SCP — Secure Channel Protocol
- SHA — Secure Hash Algorithm
- SSD — Supplementary Security Domain
- TRNG — True random number generator
- ОС — операционная система
- ЦПУ — центральное процессорное устройство

Нормативные ссылки и спецификации

1. [ISO7816-3] ISO/IEC 7816-3:2006 Identification cards – Integrated circuit cards – Part 3: Cards with contacts – Electrical interface and transmission protocols
2. [ISO7816-4] ISO/IEC 7816-4:2020 Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange
3. [ISO14443-3] ISO/IEC 14443-3:2018 Cards and security devices for personal identification – Contactless proximity objects – Part 3: Initialization and anticollision
4. [ISO14443-4] ISO/IEC 14443-4:2018 Cards and security devices for personal identification – Contactless proximity objects – Part 4: Transmission protocol
5. [GPCS] GlobalPlatform Card Specification V2.2.1
6. [GPAPI] GlobalPlatform Card API (org.globalplatform) v1.6

7. [JCRE] Java Card 3 Platform – Runtime Environment Specification, Classic Edition, V3.0.4
8. [JCVM] Java Card 3 Platform – Virtual Machine Specification, Classic Edition, V3.0.4
9. [JCAPI] Java Card 3 Platform – Application Programming Interface, Classic Edition, V3.0.4

Функциональные характеристики

Цели и назначения

Библиотека MST-JVM является основой для построения компонентно-ориентированной операционной системой (ОС), предназначенной для платёжных и идентификационных приложений. Библиотека относится к категории встроенного программного обеспечения и может использоваться в банковских платёжных смарт-картах. Построенная на основе MST-JVM ОС в свою очередь может служить для реализации карты с платёжной системой, конкретного финансового института. Это достигается путём кастомизации библиотеки, персонализации на её основе ОС.

Полученный на основе MST-JVM программный продукт позволяет использовать смарт-карту в соответствующей платёжной инфраструктуре, системах идентификации, использования в качестве токенов безопасности и доверенного доступа.

Основные функции

Реализованная на базе MST-JVM ОС на конкретных микропроцессорных модулях образуют семейство, которое обладает общими функциональными характеристиками. К таковым относятся:

• поддержка международных стандартов

• ISO/IEC 7816 (в частности, ISO/IEC 7816-3:2006 и ISO/IEC 7816-4:2020);

• ISO/IEC 14443 (в частности, ISO/IEC 14443-3:2018 и ISO/IEC 14443-4:2018);

• поддержка основных отраслевых спецификаций

• GlobalPlatform Card Specification (на текущий момент версии 2.2.1);

• Java Card Classic Platform Specifications (на текущий момент версии 3.0.4).

Краткое описание архитектуры

Библиотека имеет блочную компонентно-ориентированную архитектуру, в которой можно выделить два больших блока:

• базовая защищённая платформа (Basic Secure Platform, BSP);

• пользовательская ОС.

Basic Secure Platform (BSP) / Базовая защищенная платформа

BSP в первую очередь действует как безопасный начальный загрузчик и реализует уровень аппаратной абстракции и так называемый корень доверия (Root-of-Trust, RoT) для всей системы. BSP работает в привилегированном режиме и имеет полный доступ ко всем аппаратным ресурсам и выделенным областям памяти, недоступный пользовательской ОС.

BSP также предоставляет полный набор API-функций, реализованных с использованием специфичных для платформы механизмов повышения привилегий (таких как инструкция Supervisor Call (SVC) для архитектуры ARM). Это позволяет пользовательской ОС иметь абстрагированный и контролируемый доступ к оборудованию, криптографии и другим сервисам, реализованным BSP.

Примечание: BSP должен быть загружен начальным загрузчиком производителя микроконтроллера или с помощью процессов персонализации на кристалле.

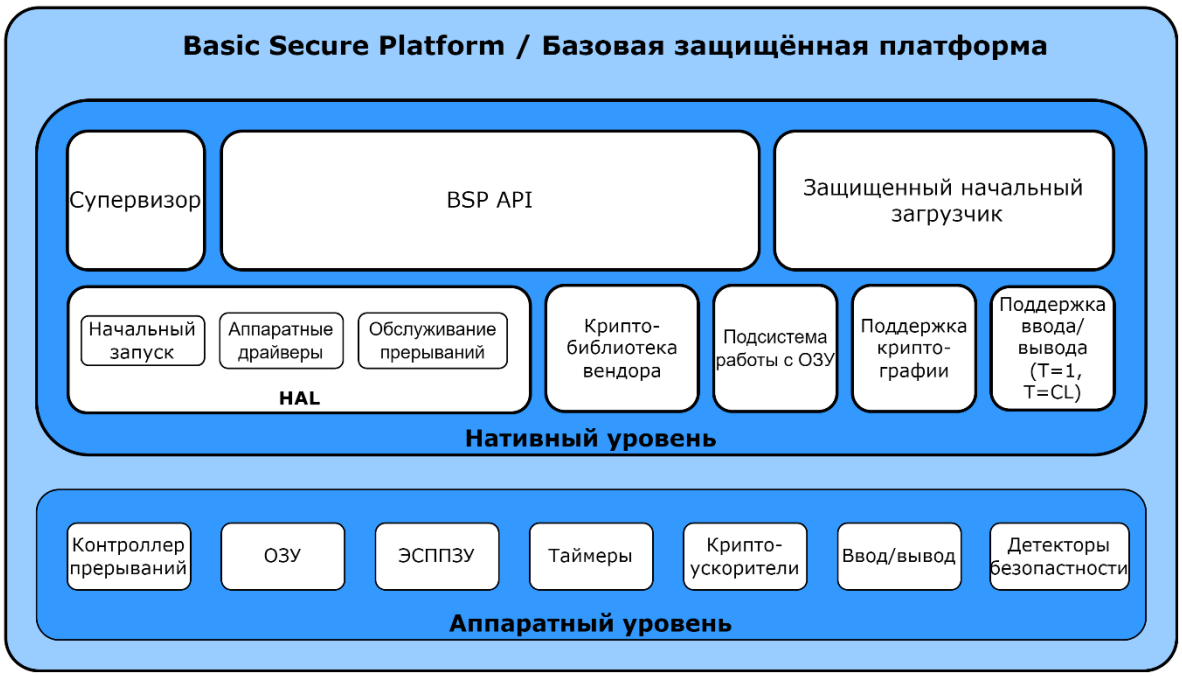

Рис.1. Архитектура блока BSP.

Нативный уровень

Этот уровень в основном содержит переносимый нативный код, написанный на языке программирования C и собранный с помощью набора инструментов, специфичных для целевой платформы. Это единственный уровень, присутствующий в BSP.

Аппаратный уровень

Этот уровень обозначает аппаратные компоненты фактического целевого микроконтроллера (ЦПУ, периферийные устройства, контроллеры и т.п.), а также микропрограммное обеспечение (firmware), при необходимости предоставляемое производителем микроконтроллера. Различные аппаратные платформы имеют различный состав компонентов, и реализация этих компонентов различна (например, может не быть криптоускорителей).

Супервизор

Модуль Супервизора БСП обеспечивает программирование MPU микроконтроллера и обработку исключительных ситуаций, связанных с нарушениями доступа к сегментам памяти и выполняемых операций

1. Должна обеспечиваться защита областей памяти, не предназначенных для ПО Пользователя от чтения, записи и исполнения кода пользователем.
2. Должна обеспечиваться защита аппаратных ресурсов и периферии, в том числе, защита системных регистров от чтения и записи пользователем.
3. Обработка прерываний MPU. Если прерывание возникло по вине пользователя и возможно восстановление, то выдача ППО сообщения о возникшей проблеме и восстановление работоспособности. Если такое восстановление невозможно, зависание БСП и создание соответствующей записи журнала
4. Мониторинг попыток взлома БСП при помощи обработки прерываний от сенсоров атак и ведения журнала атак
5. Поддержка механизмов доверенных областей в ППО. Доверенные области содержат приложения, которые проходят проверку целостности и неизменности кода.

6. Проверка вызовов HAL API и блокировка вызовов к доверенным функциям HAL API из недоверенных областей ППО
7. Блокировка доступа к доверенным функциям при обнаружении модификаций кода доверенных областей.

Защищенный начальный загрузчик

Загрузчик ОС обеспечивает безопасную загрузку пользовательской ОС и управление ее жизненным циклом.

1. Управление ATR, в частности байтом TA1- заявленной скорости ISO7816, поддержка PPS до ETU = 8;
2. Поддержка до 3-х типов ключей (транспортного, административного и ключей загрузки образа);
3. Поддержка 2-х режимов загрузки: открытого и защищенного в режиме шифрования ОС;
4. Возможность персонализации ключей загрузки образов и однократной смены транспортного/административного;
5. Верификация загруженной ОС;
6. Аутентификация с предъявлением транспортного ключа;
7. Очистка, удаление загруженного образа;
8. Возможность отката загрузки в случае возникновения ошибки;

Криптографическая библиотека

Потенциальные заказчики данного программного обеспечения будут разрабатывать криптографические алгоритмы самостоятельно. В текущую реализацию входит базовый набор реализации алгоритмов исключительно в качестве примера встраивания:

1. поддержка алгоритмов 3DES, AES128, AES256 в различных режимах работы CBC, CTR, GCM а также в режиме вычисления имитовставки;
2. поддержка хэш функций SHA-1, SHA-224 ,SHA-256;
3. подсчет контрольных сумм CRC32 , CRC16;
4. реализация функций-заглушек для поддержки следующих операций: RSA up to 4096 bits (фактическое значение зависит от аппаратной платформы), ECDSA up to 512 bits (фактическое значение зависит от аппаратной платформы), HMAC с примером реализации

На основе этого можно встраивать в систему Java Card произвольные алгоритмы.

Пользовательская ОС

Пользовательская ОС представляет собой автономную операционную систему, опирающуюся на BSP API для доступа к аппаратным ресурсам, осуществления ввода/вывода и выполнения криптографических операций.

Именно на уровне блока пользовательской ОС реализована совместимость с отраслевыми спецификациями Java Card и GlobalPlatform.

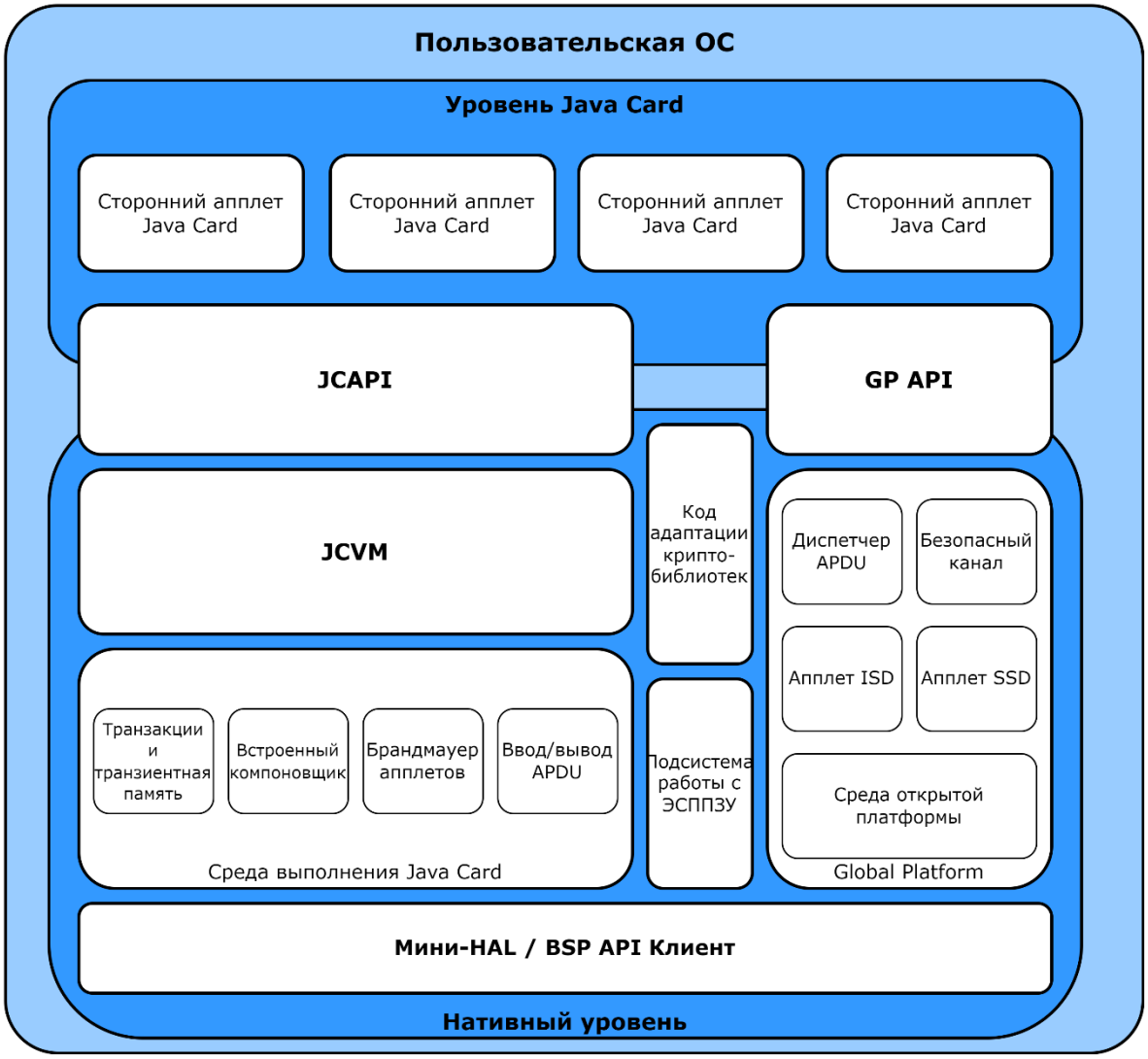

Рис.2. Архитектура блока пользовательской ОС.

Нативный уровень

Этот уровень в основном содержит переносимый нативный код, написанный на языке программирования C и собранный с помощью набора инструментов, специфичных для целевой платформы.

Среда выполнения Java Card

Среда выполнения Java Card обеспечивает абстракцию типов памяти Java Card (временная/постоянная память), атомарные обновления (транзакции), разделение апплетов и контролируемый доступ с помощью брандмауэра, абстракцию APDU-команд для различных протоколов ввода-вывода, работу встроенного компоновщика для загружаемого кода.

Более подробную информацию см. в спецификации JCRE.

JCVM

JCVM обозначает реализацию виртуальной машины Java Card. Она предоставляет собой интерпретатор байт-кода и обеспечивает поддержку для хранения и разбора структур данных, специфичных для Java Card (компоненты CAP-файла, дескрипторы пакетов и методов и т.п.).

Более подробную информацию см. в спецификации JCVM.

JC API (нативный уровень)

Это нативная часть реализации Java Card API. Она содержит нативные функции, необходимые для интерфейса JC API.

Более подробную информацию см. в спецификации JCAPI.

Код адаптации криптобиблиотек

Это обобщённый программный интерфейс криптобиблиотеки, используемый другими нативными компонентами. Он предоставляет базовые функции управления ключами и набор крипто примитивов (шифры, функции свёртки, функции вычисления цифровых подписей и т.п.).

Данный компонент делегирует фактическую обработку криптографических операций и объектов данных BSP через BSP API.

Подсистема работы с ЭСППЗУ

Это реализация «кучи» NVM общего назначения, предоставляющая услуги управления памятью другим нативным компонентам: хранилищу объектов и массивов Java Card, компонентам CAP-файлов, хранилищу ключей GlobalPlatform и тому подобному.

Диспетчер APDU

Этот компонент выполняет отправку APDU-команд в нативные приложения и апплеты Java Card, также выбор и отмену выбора приложений, а также управление логическими каналами. Он также отвечает за обработку системных APDU-команд.

Более подробную информацию см. в спецификациях JCRE и GPCS.

Среда открытой платформы

Это реализация среды исполнения, определённой в архитектуре карты GP. Эта среда управляет реестрами экземпляров файлов загрузки (Load File) и исполняемых модулей (Executable Module), иерархиями доменов безопасности, службами CVM и общим кодом обработки APDU.

Более подробную информацию см. в спецификации GPCS.

Апплет ISD

Этот компонент представляет домен безопасности эмитента и отвечает за выполнение операций по управлению содержимым карты (загрузка и удаление ключей и файлов загрузки, хранение и извлечение данных и т.п.) и управление жизненным циклом карты (см. также Приложение GP ISD).

Более подробную информацию см. в спецификации GPCS.

Апплет SSD

Этот компонент представляет собой дополнительный домен безопасности и отвечает за выполнение делегированных операций по управлению содержимым карты (загрузка и удаление ключей и файлов загрузки, хранение и извлечение данных и т.п.) и управление жизненным циклом локального приложения. Могут быть созданы несколько экземпляров SSD (см. также Приложение GP SSD).

Более подробную информацию см. в спецификации GPCS.

Безопасный канал

Этот компонент обеспечивает поддержку создания и поддержания защищенных каналов для конфиденциального и аутентифицированного доступа к службам управления картой и безопасной персонализации.

Примечание: библиотека поддерживает только SCP02.

Более подробную информацию см. в спецификации GPCS.

GP API (нативный уровень)

Это нативная часть реализации API GlobalPlatform. Она содержит нативные функции, необходимые для интерфейса GP API.

Более подробную информацию см. в спецификации GPAPI.

Мини-HAL / BSP API клиент

Этот уровень обеспечивает необходимую абстракцию целевой платформы для компонентов верхних уровней. Для MST-JVM пользовательской части, использующей BSP API для аппаратной абстракции, этот уровень является тонким и в основном содержит библиотеки и заголовки BSP API пользовательского режима и определения областей памяти, специфичных для чипа.

Уровень JavaCard

Этот уровень содержит переносимый, системный и другой код, написанный с использованием технологии Java Card.

JC API (интерфейс Java Card)

Это фактическая часть Java Card API. Используется облегчённая реализация, поскольку большинство функций делегировано нативному уровню.

Более подробную информацию см. в спецификации JCAPI.

GP API (интерфейс Java Card)

Это фактическая часть GlobalPlatform Card API. Используется облегчённая реализация, поскольку большинство функций делегировано нативному уровню.

Более подробную информацию см. в спецификации GPAPI.

Жизненный цикл

Жизненный цикл основан на GPCS с некоторыми расширениями.

Определены следующие стандартные статусы GP:

- OP_READY,
- INITIALIZED,
- SECURED,
- CARD_LOCKED,
- TERMINATED.

Кроме того, имеется два проприетарных статуса:

- UNINITIALIZED,
- PRE_INIT.

Функциональность JavaCard

Версии Java Card

Поддерживается технология Java Card на основе JCRE, JCVM и JCAPI.

Поддерживаемые функции Java Card

Основные характеристики:

• поддержка 32-разрядного целочисленного типа (наряду с классом JCint из пакета javacardx.framework.util.intx);

• сборка мусора и динамическое управление памятью;

• загрузка файлов в CAP-формате;

• реализация JCVM с усиленной безопасностью.

• Ввод/вывод APDU:

• T=0 (ISO7816-3 и ISO7816-4);

• T=1 (ISO7816-3 и ISO7816-4);

• T=CL (ISO14443-3 и ISO14443-4).

• Криптографические API (javacard.security и javacardx.crypto)

• симметричная криптография (DES, AES):

• шифрование (режимы ECB, CBC);

• имитовставка (MAC).

• асимметричная криптография (RSA, ECC):

• генерация ключа (key generation);

• совместная выработка ключа (key-agreement);

• формирование подписи (signature generation) и проверка подписи (signature verification).

• функции свёртки (hash functions):

• SHA-1;

• SHA-224;

• SHA-256.

• генерация случайных чисел (TRNG).

Функциональность GlobalPlatform

Спецификации GlobalPlatform

Поддерживаются спецификации GlobalPlatform GPCS и GPAPI.

Поддерживаемые функции GlobalPlatform

Домены безопасности:

• см. Приложение GP ISD;

• см. Приложение GP SSD.

Делегированное управление:

• токены на основе PKI (RSA);

• квитанции с использованием симметричной криптографии (DES).

Протоколы безопасного канала:

• SCP02 с параметром "i"='15'.

Проверка шаблона аутентификации данных (DAP Verification):

• DES DAP;

• RSA DAP.

Хэш-сумма блока данных файла загрузки (Load File Data Block Hash):

• на основе функции свёртки SHA-1.

• Приложения глобальных сервисов (Global Services Applications):

• приложение CVM.

Приложения GlobalPlatform

Приложение GP ISD

Домен безопасности эмитента (ISD) — это специальный домен безопасности, который выполняет специфические функции управления картой и неявно создается в процессе инициализации ОС на базе MST-JVM.

На рис.2 приложение ISD обозначено как Апплет ISD.

Одиночный экземпляр

ISD не имеет файла загрузки и исполняемого модуля. Создаётся единственный экземпляр приложения, который идентифицируется следующим AID: A000000003000000.

Параметры по умолчанию

Поддерживается SCP02 с параметром "i"='15'.

Следующий набор SCP02-ключей неявно создаётся и автоматически устанавливается как набор ключей по умолчанию.

• KVN: FF

  • Ключ шифрования сообщения, S-ENC (ID 01):

    404142434445464748494A4B4C4D4E4F

  • Ключ для кода аутентификации сообщения, S-MAC (ID 02):

    404142434445464748494A4B4C4D4E4F

  • Ключ шифрования данных, DEK (ID 03):

    404142434445464748494A4B4C4D4E4F

Приложение GP SSD

Дополнительные домены безопасности (SSD) — это создаваемые на карте дополнительные, необязательные сущности, которые представляют поставщиков приложений, эмитента карты или их агентов. Дополнительные домены безопасности могут быть созданы по мере необходимости, что позволяет формировать сложные иерархии, отражающие требуемую экосистему безопасности.

На рис.2 приложение SSD обозначено как Апплет SSD.

Файлы загрузки и исполняемый модуль

Файл загрузки SSD предоставляется как часть маски Java Card и идентифицируется следующим AID: A0000001515350.

Исполняемый модуль SSD идентифицируется следующим AID: A000000151535041.

Информация для установки и эксплуатации

Установка и эксплуатация MST-JVM имеют свои особенности. Для установки требуется наличие определённого оборудования и программного обеспечения для работы. Эксплуатация конечного продукта на базе библиотеки выполняется опосредованно, в рамках взаимодействия со смарт-картой.

Общее описание

На основе библиотеки, путём её адаптации формируется операционная система для смарт-карт. Установка любой операционной системы для смарт-карт и её эксплуатация сводятся к двум процедурам, обычно называемым процедурой инициализации и процедурой персонализации. Полученный в ходе адаптации MST-JVM продукт не является исключением, хотя наличие компонента BSP накладывает дополнительные требования.

Подготовка

Образ конкретной реализации ОС для конкретного микропроцессорного модуля загружается с помощью BSP и активируется.

Предполагается, что карта находится в статусе UNINITIALIZED.

Примечание: настоятельно рекомендуется использовать случайные данные для системного PIN-кода на каждой карте.

Базовая процедура инициализации

1. Перевести в статус PRE_INIT с помощью APDU-команды OS CHANGE STATE (инициализировать).

2. Персонализировать системный PIN-код (System PIN) с помощью APDU-команды SYSPIN INIT.

3. Заменить набор SCP02-ключей по умолчанию для приложения GP ISD набором случайных ключей для каждой карты:

• с помощью команды GP PUT KEY загрузить данные нового набора ключей;

• с помощью команды GP DELETE удалить набор ключей по умолчанию (с KVN равным FF).

4. Активировать необходимый статус жизненного цикла приложения GP ISD, используя команду GP SET STATUS (например, статус INITIALIZED).

Предупреждение: если был активирован какой-либо статус жизненного цикла, выходящий за рамки OP_READY, убедитесь, что установлены и должным образом поддерживаются соответствующие механизмы безопасности (например, сконфигурирован безопасный канал).

Специфическая процедура инициализации

Данный раздел носит обобщенный вид, так как зависит от требований конкретного приложения. На этом этапе подготавливается необходимая основа для конкретного приложения:

• создаётся иерархия дополнительных доменов безопасности GP, упомянутых в разделе Приложение GP SSD (при необходимости).

• загружаются внешние пакеты Java Card (при необходимости).

• создаются и экстрадируются экземпляры апплетов Java Card (при необходимости).

Процедура персонализации

Данный раздел носит обобщенный вид, так как зависит от требований конкретного приложения. На этом этапе для каждой карты выполняется её персонализация уникальными данными:

• создаются и персонализируются дополнительные наборы ключей (при необходимости):

• для безопасных каналов;

• для DAP;

• для делегированного управления.

• SSD переводятся в соответствующие статусы жизненного цикла (при необходимости);

• персонализируются экземпляры апплетов (напрямую или через связанные с ними домены безопасности).