Приложение SCOneVitalData

Руководство оператора

Версия: 1.0 от 02.02.2023

Количество листов: на 28 листах

Москва, 2023

Общество с ограниченной ответственностью «СКОН ТЕХНОЛОГИИ»
Юридический адрес: 115230, город Москва, Электролитный проезд, д. 1 к. 3, помещ./этаж х/3 ком. 9, 13
ИНН 9726018244, КПП 772601001, р/с 40702810000000256170 в АО "Райффайзенбанк", г. Москва,
к/с 30101810200000000700, БИК 044525700, ОГРН 1227700461790
+7 925 011 30 07 | www.sconetech.ru

АННОТАЦИЯ

В данном программном документе приведено руководство оператора по настройке и использованию приложения SCOneVitalData, представляющего собой файловую систему, обеспечивающую возможность хранения персональных и витальных данных о держателе на микропроцессорной Java-карте, считывание и запись данных, а также изменение данных в постэмиссионный период.

СОДЕРЖАНИЕ

1. ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

AID (Application Identifier)	Идентификатор пакета, приложения и экземпляра приложения на Java-карте, используемый в качестве имени приложения
APDU	Application Protocol Data Unit – тип управляющих команд, используемых для интегральных схем. ГОСТ Р ИСО/МЭК 7816-4-2013 «Карты идентификационные. Карты на интегральных схемах. Часть 4. Организация, защита и команды для обмена»
API	API (программный интерфейс приложения, интерфейс прикладного программирования) (англ. application programming interface, API [эй-пи-ай]) – описание способов (набор классов, процедур, функций, структур или констант), которыми одна программа может взаимодействовать с другой программой.
GlobalPlatform	Спецификация GlobalPlatform Card Specification (GPCS) (далее – GlobalPlatform) определяет безопасные динамические процедуры управления приложениями карты и картой, а также компоненты карты, наборы команд и последовательность их исполнения, механизмы безопасности, используемые в процедурах управления приложениями микропроцессорных карт
Java Card	Версия Java-платформы для устройств с крайне ограниченными вычислительными ресурсами. По сравнению с другими версиями Java изменен байткод, требования к исполняющей платформе, состав стандартных библиотек
МК	Микроконтро́ллер (англ. Micro Controller Unit, MCU) – микросхема, предназначенная для управления электронными устройствами, банковскими картами
ОС	Операционная система смарт-карт (англ. operating system smart cards, OS smart cards) – комплекс взаимосвязанных программ, предназначенных для управления ресурсами МК и организации взаимодействия с пользователем по средствам интерфейсов ввода\вывода.
ПО	Программное обеспечение

2. НАЗНАЧЕНИЕ ПРОГРАММЫ

2.1. Функциональное назначение программы

Приложение SCOneVitalData представляет собой файловую систему, обеспечивающую возможность хранения персональных и витальных данных о держателе на микропроцессорной Java-карте, считывание и запись данных, а также изменение данных в постэмиссионный период.

Приложение SCOneVitalData предоставляет следующие функциональные возможности:

- хранение в виде файлов персональных и витальных данных о держателе таких, как:
  - ФИО;
  - Дата рождения;
  - Пол;
  - Данные об удостоверении личности;
  - Фотография;
  - Контактный телефон;
  - Адрес электронной почты;
  - Адрес места жительства;
  - Группа крови и резус-фактор;
  - Данные о полисе медицинского страхования;
  - Рост, вес;
  - Противопоказания обезболивающих препаратов;
  - Аллергия на препараты;
  - Хронические заболевания;
  - Данные о медицинских осмотрах.
- чтение данных из файлов:
- запись данных в файлы;
- изменение данных в файлах постэмиссионный период;
- взаимная аутентификация с внешним хостом;
- построение защищенного канала с хостом;
- возможность ограничения доступа к каждому файлу на чтение и запись.

2.2. Эксплуатационное назначение программы

Приложение SCOneVitalData – это Java-приложение, работающее на микропроцессорных Java-картах с ОС Java Card, соответствующих спецификациям GlobalPlatform.

2.3. Описание функции по загрузке и персонализации приложения

Для загрузки и персонализации приложения SCOneVitalData необходимо выполнить следующую последовательность команд.

2.3.1. Выбор менеджера карты CardManager/ISD, Select

Синтаксис команды приведен в разделе 11.9 документа Global Platform Card Specification (Version 2.2.1 January 2011).

2.3.2. Посторонние безопасного канала с менеджером карты CardManager/ISD

Синтаксис команд приведен в разделе E.1 документа Global Platform Card Specification (Version 2.2.1 January 2011).

2.3.3. Инсталляция для загрузки пакета апплета Install for load

Синтаксис команды приведен в разделе 11.5, 11.5.2.3.1 документа Global Platform Card Specification (Version 2.2.1 January 2011).

Структура команды

Поле	Значение	Описание
CLA	‘80’	
INS	‘E6’	
P1	‘02’	
P2	‘00’	
P3	‘11’	Длина данных
Data	‘0C53..00’	0C53434F6E654C6F79616C747900000000

Ответные данные - 00

2.3.4. Загрузка пакета апплета Load

Синтаксис команды приведен в разделе 11.6 документа Global Platform Card Specification (Version 2.2.1 January 2011).

2.3.5. Инсталляция для выбора Install for install and make selectable

Синтаксис команды приведен в разделе 11.5, 11.5.2.3., 11.5.2.3.3 документа Global Platform Card Specification (Version 2.2.1 January 2011).

Структура команды

Поле	Значение	Описание
CLA	‘80’	
INS	‘E6’	
P1	‘0C’	
P2	‘00’	
P3	‘33’	Длина данных
Data	‘0C53..00’	0C53434F6E654C6F79616C74790D53434F6E654C6F79616C7479010E53434F6E654C6F79616C74790101010005C903052B0000

Ответные данные - 00

2.3.6. Функции персонализации апплета

Функции по персонализации апплета переставлены в таблице ниже.

Команда	Описание
Initialize Update	GP аутентификация
External Authenticate	GP аутентификация
Put Key	Загрузка ключа приложения
Put Data	Загрузка данных
Set State	Перевод приложения на фазу эксплуатации

2.3.6.1 Построение безопасного канала с апплетом Select

Синтаксис команды приведен в разделе 11.9 документа Global Platform Card Specification (Version 2.2.1 January 2011).

2.3.6.2 Загрузка ключа приложения Put Key

Структура команды

Поле	Значение	Описание
CLA	‘80’	
INS	‘D8’	
P1	‘00’	
P2	‘00’	
P3	‘17’	Длина данных
Data	‘018010xx..xx	xx..xx криптограмма ключа

Ответные данные

Отсутствуют

2.3.6.3 Загрузка данных приложения Put Data

Структура команды

Поле	Значение	Описание
CLA	‘80’	
INS	‘DA’	
P1	‘DF’	
P2	‘11’	
P3	‘0B’	Длина данных
Data	‘00000000000000000017’	Данные

Ответные данные

Отсутствуют

2.3.6.4 Установка состояния приложения SET STATE

Структура команды

Поле	Значение	Описание
CLA	‘80’	
INS	‘F0’	
P1	‘40’	
P2	‘0F’	Персонализировано

Поле	Значение	Описание
P3	‘00’	Длина данных

Ответные данные

Отсутствуют

2.4. Состав функций приложения

Приложение включает в себя следующие функции:

- выбор приложения, Select;
- взаимная аутентификация с приложением, Initialize Update, External Authenticate;
- создание файла, Create File;
- смена ключа приложения, Change Key;
- выбор файла приложения, Select File;
- чтение файла приложения, Read Binary;
- запись файла приложения, Write Binary;
- блокировка и разблокировка приложения, Block, Unblock;
- чтение внутренних данных приложения, Get Data.

2.4.1. Функция выбора приложения Select

Описание команды

Выбор приложения. При выполнении данной команды сбрасываются сессионные переменные. В случае, если приложение заблокировано, будет возвращено SW=6283.

Структура команды

Поле	Значение	Описание
CLA	‘00’	
INS	‘A4’	SELECT
P1	‘04’	
P2	‘00’	
P3	‘XX’	Длина AID приложения
Data	‘XXXX…’	AID приложения

Ответные данные

File Control Information (FCI). FCI представляет собой TLV выражение (составной тэг 6Fh). Кроме стандартных тэгов FCI включает тэг DF11h, значение которого представляет собой номер карты.

2.4.2. Функция взаимной аутентификация с приложением Initialize Update

Описание команды

Инициализация процесса взаимной аутентификации. Для завершения процесса аутентификации необходимо выполнить команду External Authenticate. Максимальное число последовательных незавершенных успешно аутентификаций для одного ключа равно 10.

Структура команды

Поле	Значение	Описание
CLA	‘80’	
INS	‘50’	INITIALIZE UPDATE
P1	‘00’	
P2	‘XX’	Номер ключа для аутентификации
P3	‘08’	Длина случайного числа хоста
Data	‘XXXX…’	Случайное число хоста

Ответные данные

Имя	Длина (байт)	Описание
Key Version	1	Версия ключа
Challenge	8	Случайное число приложения
Host cryptogram	8	Криптограмма приложения

2.4.3. Функция взаимной аутентификация с приложением External Authenticate

Описание команды

Команда завершает процедуру аутентификации. Команде External Authenticate должна предшествовать команда Initialize Update с тем же номером ключа. При этом команды Initialize Update и External Authenticate не обязательно должны непосредственно следовать одна за другой.

Структура команды

Поле	Значение	Описание
CLA	‘80’	
INS	‘82’	EXTERNAL AUTHENTICATE
P1	‘00’	

Поле	Значение	Описание
P2	‘XX’	Номер ключа для аутентификации
P3	‘08’	Длина криптограммы
Data	‘XXXX…’	Криптограмма хоста

Ответные данные

Отсутствуют

2.4.4. Функция создание файла Create File

Описание команды

Создание файла. При создании файла задаются уникальный идентификатор (id) файла, размер файла и условия доступа на чтение и запись данных. Имеется также возможность задать начальные данные файла. Для файлов с доступом только на чтение – это единственная возможность определить содержимое файла.

Блок данных команды имеет следующий вид:

Data block = KK | SSSS | RR | WW | XX.., где

KK – id файла (1 байт)

SSSS – размер файла (2 байта)

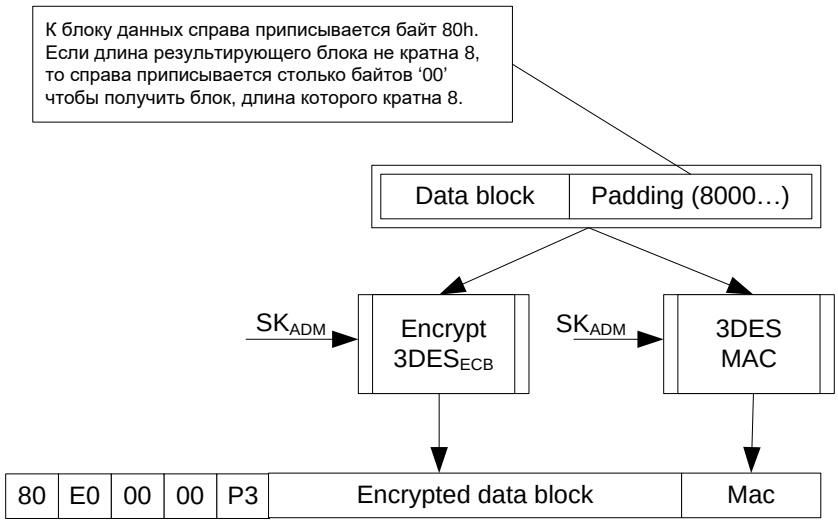
RR – условия доступа на чтение (1 байт)

WW – условия доступа на запись (1 байт)

XX.. – данные файла (от нуля до 235 байт)

Команде должна предшествовать успешная аутентификация на ключе K_{ADM}. Блок данных команды должен быть зашифрован и подписан. Если апплет обнаруживает что подпись команды неверна, он возвращает SW=6982 и закрывает безопасный канал на ключе K_{ADM}. Для возобновления безопасного канала необходимо выполнить повторную аутентификацию на ключе K_{ADM}.

Данные файла должны быть зашифрованы и подписаны. Шифрование и формирование подписи выполняется по следующей схеме:



Структура команды

Поле	Значение	Описание
CLA	‘80’	
INS	‘E0’	CREATE FILE
P1	‘00’	
P2	‘00’	
P3	‘XX’	Длина зашифрованных данных
Data	Cryptogram | Mac	Зашифрованный блок данных и цифровая подпись

Ответные данные

Отсутствуют

2.4.5. Функция смены ключа приложения Change Key

Описание команды

Команда позволяет заменить указанный ключ. Команде должна предшествовать успешная аутентификация на ключе K_{ADM}.

Блок данных команды имеет следующий вид:

Data block = VV | ‘8010’ | EK | ‘03’ | KCV, где

VV – версия ключа (1 байт)

EK – зашифрованное сессионном ключе SK_{ADM} значение ключа (16 байт)

KCV – Key Check Value ключа (3 байт)

Если апплет обнаруживает что подпись команды неверна, он возвращает SW=6982 и закрывает безопасный канал на ключе K_{ADM}. Для возобновления

безопасного канала необходимо выполнить повторную аутентификацию на ключе K_{ADM}.

Схема формирования команды

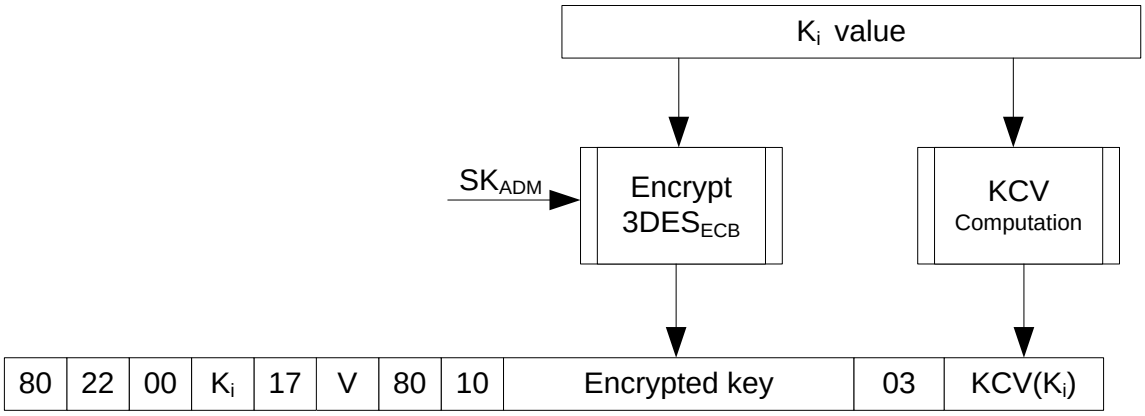
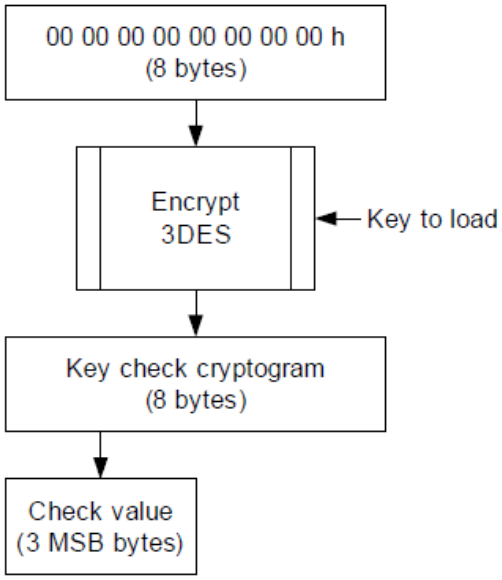


Схема расчета проверочного число ключа (KCV Computation)



Структура команды

Поле	Значение	Описание
CLA	‘80’	
INS	‘22’	CHANGE KEY
P1	‘00’	
P2	‘XX’	Номер ключа заменяемого ключа
P3	‘17’	Длина данных (23 байта)
Data	Data block	См. выше

Ответные данные

Отсутствуют

2.4.6. Функция выбора файла приложения Select File

Описание команды

Выбор файла для дальнейших операций с ним.

Структура команды

Поле	Значение	Описание
CLA	‘80’	
INS	‘A4’	SELECT (FILE)
P1	‘00’	
P2	‘XX’	Идентификатор файла
P3	‘04’	Длина данных

Ответные данные

Имя	Длина (байт)	Описание
Length	2	Длина выбранного файла
Read AC	1	Права доступа к файлу на чтение
Write AC	1	Права доступа к файлу на запись

2.4.7. Функция чтения файла приложения Read Binary

Описание команды

Чтение данных из файла. Для успешного выполнения команды необходимо, чтобы были соблюдены права доступа к файлу на чтение. Существуют два варианта данной команды: без подписи (для чтения данных без аутентификации на ключе) и с подписью (для чтения данных с аутентификацией на ключе). Если апплет обнаруживает что подпись команды неверна, он возвращает SW=6982 и закрывает безопасный канал на ключе, защищающем доступ к файлу. Для возобновления безопасного канала необходимо выполнить повторную аутентификацию.

Структура команды без подписи

Поле	Значение	Описание
CLA	‘80’	
INS	‘B0’	READ BINARY
P1	‘XX’	Старший байт смещения в файле
P2	‘XX’	Младший байт смещения в файле
P3	‘XX’	Длина запрашиваемых данных (L_R)

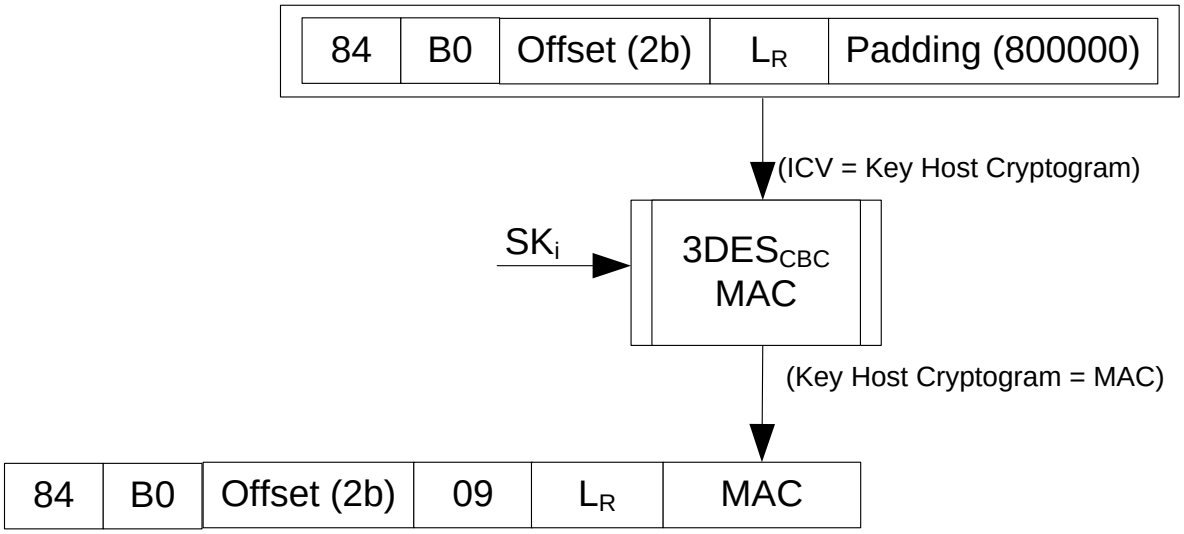
Структура команды с подписью

Подпись команды формируется на сессионном ключе, построенном в результате аутентификации, открывающей доступ к файлу.

Поле	Значение	Описание
CLA	‘84’	
INS	‘B0’	READ BINARY
P1	‘XX’	Старший байт смещения в файле
P2	‘XX’	Младший байт смещения в файле
P3	‘09’	Длина данных
Data	‘XX’+MAC	‘XX’ - Длинна запрашиваемых данных (L_R) MAC – подпись команды

Схема формирования команды с подписью

При формировании подписи используется значение ICV. В процессе аутентификации на ключе K_J ICV получает значение криптограммы хоста в команде External Authenticate. После успешного выполнения каждой команды ReadBinary/WriteBinary значение ICV заменяется на криптограмму хоста, сформированную для данной команды.



Ответные данные

Имя	Длина (байт)	Описание
Data	L_R	Данные файла

2.4.8. Функция запись файла приложения Write Binary

Описание команды

Запись данных в файл. Для осуществления записи необходимо, чтобы были соблюдены права доступа к файлу на запись. Существуют два варианта данной команды: без подписи (для записи данных без аутентификации на ключе) и с подписью (для записи данных с аутентификацией на ключе). Если апплет обнаруживает что подпись команды неверна, он возвращает SW=6982 и закрывает безопасный канал на ключе, защищающем доступ к файлу. Для возобновления безопасного канала необходимо выполнить повторную аутентификацию.

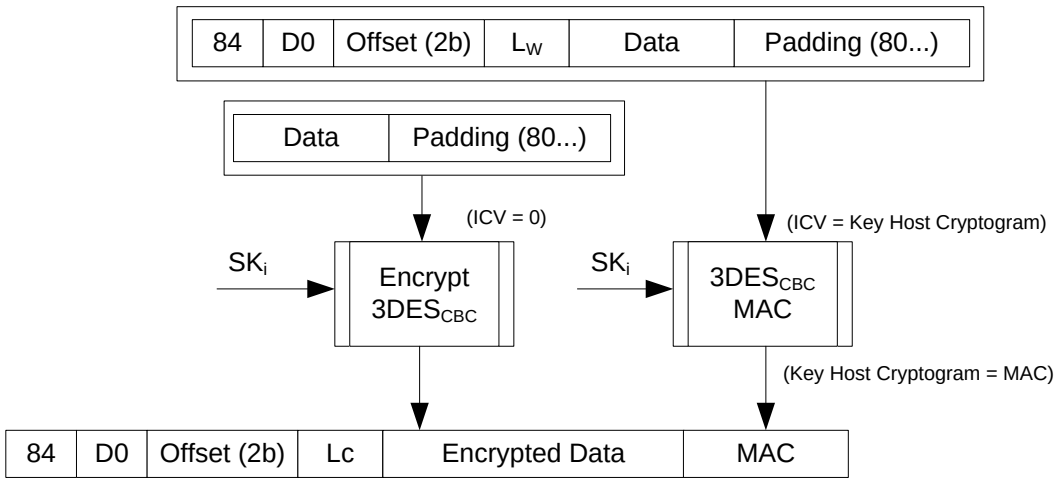
Структура команды без подписи

Поле	Значение	Описание
CLA	‘80’	
INS	‘D0’	WRITE BINARY
P1	‘XX’	Старший байт смещения в файле
P2	‘XX’	Младший байт смещения в файле
P3	‘XX’	Длина записываемых данных (L_W)
Data	‘XXXX…’	Данные, которые необходимо записать в файл

Структура команды с шифрованием

Поле	Значение	Описание
CLA	‘84’	
INS	‘D0’	WRITE BINARY
P1	‘XX’	Старший байт смещения в файле
P2	‘XX’	Младший байт смещения в файле
P3	‘XX’	Длина блока данных = Длина записываемых данных (L_W) + паддинг +длина MAC (8)
Data	ED | MAC	ED - Зашифрованные данные MAC - Подпись команды

Схема формирования команды с подписью и шифрованием



Обратите внимание, что при расчете подписи поле Lc команды должно иметь значение L_W т.е. должно быть равно длине записываемых в файл данных. Кроме того, при формировании подписи используется значение ICV. В процессе аутентификации на ключе K_J ICV получает значение криптограммы хоста в команде External Authenticate. После успешного выполнения каждой команды ReadBinary/WriteBinary значение ICV заменяется на криптограмму хоста, сформированную для данной команды.

Ответные данные

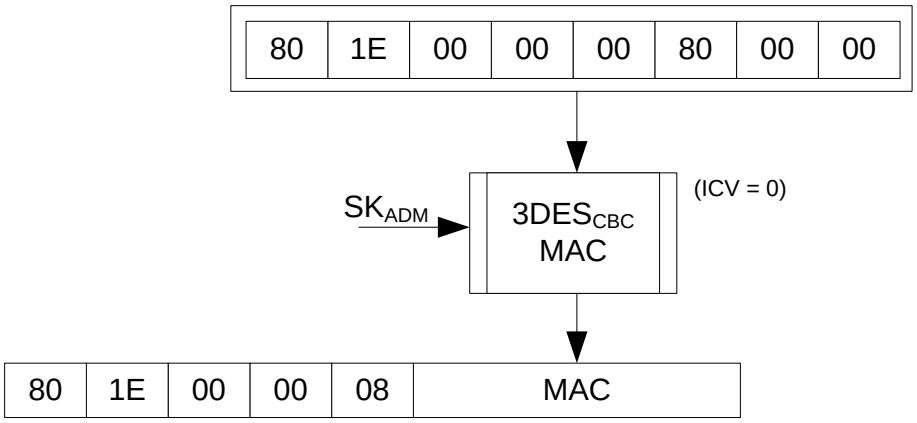
Отсутствуют

2.4.9. Функция блокировки приложения Block

Описание команды

Блокирование приложения. Требуется аутентификация на административном ключе K_{ADM}. Если апплет обнаруживает, что подпись команды неверна, он возвращает SW=6982 и закрывает безопасный канал на ключе K_{ADM}. Для возобновления безопасного канала необходимо выполнить повторную аутентификацию на ключе K_{ADM}.

Схема формирования команды



Структура команды

Поле	Значение	Описание
CLA	‘80’	
INS	‘1E’	BLOCK
P1	‘00’	
P2	‘00’	
P3	‘08’	Длина подписи
Data	‘XXXX…’	MAC

Ответные данные

Отсутствуют

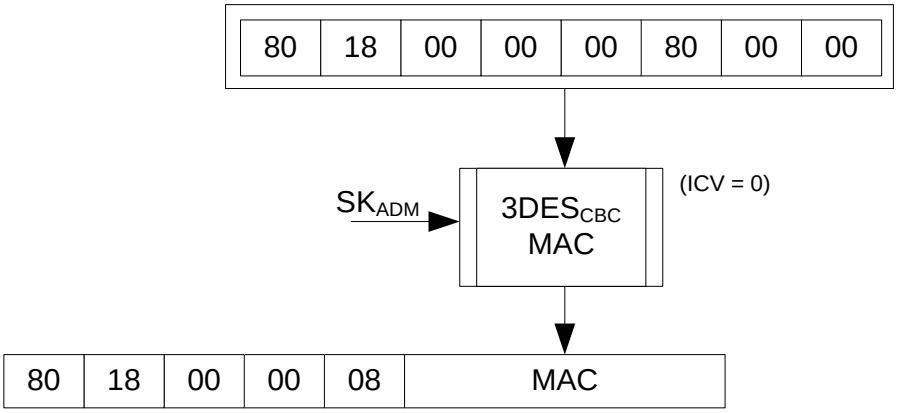
2.4.10. Функция разблокировки приложения Unblock

Описание команды

Снятие блокировки приложения. Требуется аутентификация на административном ключе K_{ADM}. Если апплет обнаруживает что подпись команды неверна, он возвращает SW=6982 и закрывает безопасный канал на

ключе K_{ADM}. Для возобновления безопасного канала необходимо выполнить повторную аутентификацию на ключе K_{ADM}.

Схема формирования команды



Структура команды

Поле	Значение	Описание
CLA	‘80’	
INS	‘18’	UNBLOCK
P1	‘00’	
P2	‘00’	
P3	‘08’	Длина подписи
Data	‘XXXX…’	MAC

Ответные данные

Отсутствуют

2.4.11. Чтение внутренних данных приложения Get Data

Описание команды

Получение параметров служебного приложения. Для получения каждого конкретного параметра приложения необходимо задать его тэг.

Тэг	Длина параметра	Описание
‘DF10’	2	Версия апплета. Первый байт содержит номер версии, второй байт номер подверсии.
‘DF11’	11	номер карты

Тэг	Длина параметра	Описание
‘DF12’	Var	Список файлов приложения. Список имеет следующий формат: Количество файлов (1 байт) | ID файла 1 (1 байт) | ID файла 2 (1 байт) | … | ID файла N (1 байт)
‘DF13’	1	Текущая версия ключа K_{DECR}

Структура команды

Поле	Значение	Описание
CLA	‘80’	
INS	‘CA’	GET DATA
P1	‘XX’	Старший байт тэга
P2	‘XX’	Младший байт тэга
P3	‘00’	

Ответные данные

Имя	Длина (байт)	Описание
Tag	2	Тэг
Length	1	Длина значения
Value	Var	Значение

3. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Приложение SCOneVitalData представляет собой бинарный файл формата «cap», скомпилированный под ОС Java Card вер. 2.2.1 и выше., предназначенный для загрузки во встроенную память микропроцессорной Java-карты с ОС Java Card вер. 2.2.1 и выше.

Загрузка ПО во встроенную память микропроцессорной Java-карты осуществляется с помощью APDU-команд согласно протоколу по ГОСТ Р ИСО/МЭК 7816, либо ГОСТ Р ИСО/МЭК 14443.

3.1. Минимальный состав программно-аппаратных средств

Программно-аппаратные требования смарт карт представлены в таблице 1.

Таблица 1 – Программные, аппаратные требования смарт карт

Наименование требования	Значение
Программные требования	
Java Card SDK	2.2.1
GlobalPlatform Card Specification	v2.1.1 – v1.0
Аппаратные требования	
Семейство микроконтроллеров	Java-карты с ОС Java Card 2.2, GlobalPlatform 2.1
Криптографический сопроцессор 3DES	Доступ из Java API
Безопасность	В В составе операционных систем (ОС) смарт-карт
Объем ПЗУ \Flash	12 KB
Объем ОЗУ\RAM	6 KB

3.2. Требования к составу периферийных устройств

Устройства чтения/записи Java Card, например, PC\SC кард-ридеры.

3.3. Требования к персоналу (оператору)

Пользователь должен обладать практическими навыками владения персонального ПК, иметь знания в области смарт-карт, обладать практическими навыками системного программирования для МК семейства Java Card v 2.2 и выше.

4. ПРИМЕР РАБОТЫ С ПРИЛОЖЕНИЕМ

4.1. Загрузка приложения SCOneVitalData

Ниже представлен пример загрузки пакета приложения SCOneVitalData на МК с ОС Java Card, соответствующих требованиям раздела 3.1:

МК с ОС Java Card	APDU интерфейс	Управлявшее прикладное ПО и устройство работы со смарт-картами	Комментарий
	← Reset		
	ATR →		
	Построение безопасного канала, SCP 02, с менеджером карты		
	← SELECT AID ISD/CM		Выбор менеджера карты по AID
	FCI ISD/CM + SW 9000 →		
	← INITIALIZE UPDATE		Взаимная аутентификация, протокол
	DATA + SW 9000 →		
	←EXTERNAL AUTHENTICATE		
	SW 9000 →		
	Загрузка пакета		
	←INSTALL FOR LOAD AID 53434F6E65566974616C44617461		Загрузка пакета AID 53434F6E6556697461 6C44617461
	00 + SW 9000 →		
	← LOAD DATA [80E80000EExx]		
	SW 9000 →		
	← LOAD DATA [80E80001EExx]		
	SW 9000 →		
	…		
	← LOAD DATA [80E80023EExx]		
	SW 9000 →		
	← 80E88024070D050708080814		Последняя команды загрузки данных пакета

МК с ОС Java Card	APDU интерфейс	Управлявшее прикладное ПО и устройство работы со смарт-картами	Комментарий
	00 + SW 9000 →		

4.2. Персонализация экземпляра приложения SCOneVitalData

Ниже представлен пример персонализации (запись данных) экземпляра приложения SCOneVitalData, загруженного на МК с ОС Java Card, соответствующих требованиям раздела 3.1:

SCOneVitalData	APDU интерфейс	Управлявшее прикладное ПО и устройство работы со смарт-картами	Комментарий
	Персонализация экземпляра апплета		
	← Reset		
	ATR →		
	Построение безопасного канала с менеджером карты		
	← SELECT AID ISD/CM		Выбор менеджера карты по AID
	FCI ISD/CM + SW 9000 →		
	← INITIALIZE UPDATE		Взаимная аутентификация, протокол
	DATA + SW 9000 →		
	←EXTERNAL AUTHENTICATE		
	SW 9000 →		
	Инсталляция экземпляра апплета		
	←INSTALL FOR INSTALL and MAKE SELECTABLE 80E60C00390E53434F6E6556697461 6C446174610F53434F6E655669 74616C4461746101 1053434F6E655 66974616C4461746101010005C9 03052B0000		Инсталляция экземпляра апплета AID 53434F6E6556697461 6C446174610101
	00 + SW 9000 →		

SCOneVitalData	APDU интерфейс	Управлявшее прикладное ПО и устройство работы со смарт-картами	Комментарий
	Персонализация экземпляра апплета		
	← SELECT AID 53434F6E65566974616C44617461010 1		Выбор экземпляра апплета AID 53434F6E6556697461 6C446174610101
	FCI + SW 9000 →		
	← INITIALIZE UPDATE		
	DATA + SW 9000 →		
	←EXTERNAL AUTHENTICATE		
	SW 9000 →		
	← PUT DATA DF 11 [80DADF110B0000000000000000000 00017]		Установка тега DF11, серийный номер
	SW 9000 →		
	← PUT KEY 00 [80D8000017018010xx..xx]		Установка ключа K_FILE 0
	SW 9000 →		
	← PUT KEY 10 [80D8001017018010xx..xx]		Установка ключа K_OTP
	SW 9000 →		
	← PUT KEY 11 [80D8000117018010xx..xx]		Установка ключа K_AC
	SW 9000 →		
	← PUT KEY 12 [80D8001217018010xx..xx]		Установка ключа K_DATA
	SW 9000 →		
	← PUT KEY 01 [80D8000117018010xx..xx]		Установка ключа K_FILE 1
	SW 9000 →		
	← PUT KEY 02 [80D8000217018010xx..xx]		Установка ключа K_FILE 2
	SW 9000 →		
	Установка статуса		

SCOneVitalData	APDU интерфейс	Управлявшее прикладное ПО и устройство работы со смарт-картами	Комментарий
	← 80F0400F00		Установка статуса персонализировано
	SW 9000 →		
	Загрузка пользовательских данных		
	← SELECT AID 53434F6E65566974616C446174610101		Выбор экземпляра апплета AID 53434F6E65566974616C446174610101
	FCI + SW 9000 →		
	← INITIALIZE UPDATE		
	DATA + SW 9000 →		
	← EXTERNAL AUTHENTICATE [8082000008xx..xx]		
	SW 9000 →		
	← CREATE FILE DF11..DF16 [80E0000020xx..xx]		Запись содержимого файла DF11..DF16
	SW 9000 →		
	← SELECT FILE ID 0001 [80A4000104]		Выбор файла ID 0001 - DF 11..DF16
	DATA+SW 9000 →		Данные атрибутов файла
	← INITIALIZE UPDATE		
	DATA + SW 9000 →		
	← EXTERNAL AUTHENTICATE [8082000008xx..xx]		
	← WRITE BINARY [84D000E7F0xx..xx]		Запись данных в файл DF11..DF16
	SW 9000 →		
	…		
	← WRITE BINARY [84D0090688 xx..xx]		

SCOneVitalData	APDU интерфейс	Управлявшее прикладное ПО и устройство работы со смарт-картами	Комментарий
	SW 9000 →		
	← INITIALIZE UPDATE		
	DATA + SW 9000 →		
	← EXTERNAL AUTHENTICATE [8082000008xx..xx]		
	SW 9000 →		
	← CREATE FILE DF21..DF24 [80E0000070xx..xx]		Запись содержимого файлов DF21..DF24
	SW 9000 →		
	← CREATE FILE DF31..DF40 [80E0000060 xx..xx]		Запись содержимого файлов DF31..DF40
	SW 9000 →		

4.3. Чтение данных из экземпляра приложения SCOneVitalData

Ниже представлен пример чтения данных из экземпляра приложения SCOneVitalData, загруженного на МК с ОС Java Card, соответствующих требованиям раздела 3.1:

SCOneVitalData	APDU интерфейс	Управлявшее прикладное ПО и устройство работы со смарт-картами	Комментарий
	Чтение данных приложения		
	← Reset		
	ATR →		
	← SELECT AID 53434F6E65566974616C446174610101		экземпляра апплета AID 53434F6E65566974616C446174610101
	FCI + SW 9000 →		
	← GET DATA [80CADF1100]		Запрос тега DF11 серийный номер
	DF110B00000000000000000000017 + SW 9000 →		

SCOneVitalData	APDU интерфейс	Управлявшее прикладное ПО и устройство работы со смарт-картами	Комментарий
	← SELECT FILE ID 0001 80A4000104		Выбор файла ID 0001 , DF11..DF16
	DATA+SW 9000 →		Данные атрибутов файла
	← INITIALIZE UPDATE		Построение безопасного канала
	DATA + SW 9000 →		
	← EXTERNAL AUTHENTICATE [8082000008xx..xx]		
	SW 9000 →		
	← READ BINARY [84B0000009 10 xx..xx]		Получение данных файла
	DF11820959xx..xx+ SW 9000 →		
	…		
	← READ BINARY [84B0090609 xx..xx]		Получение остатка данных файла
	← SELECT FILE ID 0002 [80A4000204]		Выбор файла ID 0002 - DF 21..DF24
	DATA+SW 9000 →		Данные атрибутов файла
	← INITIALIZE UPDATE		Построение безопасного канала
	DATA + SW 9000 →		
	← EXTERNAL AUTHENTICATE [8082000008xx..xx]		
	SW 9000 →		
	← READ BINARY [84B0000009 60 xx..xx]		Чтение данных файла, сдвиг x00
	DATA [DF210Cxx]+SW 9000 →		
	← SELECT FILE ID 0003 [80A4000304]		Выбор файла ID 0002 - DF 31..DF40

SCOneVitalData	APDU интерфейс	Управлявшее прикладное ПО и устройство работы со смарт-картами	Комментарий
	DATA+SW 9000 →		Данные атрибутов файла
	← INITIALIZE UPDATE		Построение безопасного канала
	DATA + SW 9000 →		
	← EXTERNAL AUTHENTICATE [8082000008xx..xx]		
	SW 9000 →		
	← READ BINARY [84B0000009 4C xx..xx]		Чтение данных файла, сдвиг x00
	DATA [DF310Fxx]+SW 9000 →		