



Приложение SCOneLoyalty

Руководство оператора

Версия: 1.0 от 17.11.2022

Количество листов: на 30 листах

Москва, 2022

Общество с ограниченной ответственностью «СКОН ТЕХНОЛОГИИ»
Юридический адрес: 115230, город Москва, Электролитный проезд, д. 1 к. 3, помещ./этаж х/З 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 SCLoyalty, представляющего собой файловую систему, обеспечивающую возможность хранения данных о программах лояльности на микропроцессорной Java-карте, считывание и запись данных, а также изменение данных в постэмиссионный период.

СОДЕРЖАНИЕ

1. ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	4
2. НАЗНАЧЕНИЕ ПРОГРАММЫ.....	5
2.1. Функциональное назначение программы	5
2.2. Эксплуатационное назначение программы	5
2.3. Описание функции по загрузке и персонализации приложения	5
2.3.1. Выбор менеджера карты CardManager/ISD, Select	6
2.3.2. Посторонние безопасного канала с менеджером карты CardManager/ISD	6
2.3.3. Инсталляция для загрузки пакета апплета Install for load.....	6
2.3.4. Загрузка пакета апплета Load.....	6
2.3.5. Инсталляция для выбора Install for install and make selectable	6
2.3.6. Функции персонализации апплета	7
2.3.6.1 Построение безопасного канала с апплетом Select	7
2.3.6.2 Загрузка ключа приложения Put Key	7
2.3.6.3 Загрузка данных приложения Put Data	8
2.3.6.4 Установка состояния приложения Set State	8
2.4. Состав функций приложения	9
2.4.1. Функция выбора приложения Select	9
2.4.2. Функция взаимной аутентификация с приложением Initialize Update	10
2.4.3. Функция взаимной аутентификация с приложением External Authenticate	10
2.4.4. Функция генерация пароля приложения Internal Authentica	11
2.4.5. Функция генерация криптограммы приложения Generate AC	11
2.4.6. Функция дешифрования Decrypt	12
2.4.7. Функция создание файла Create File	13
2.4.8. Функция смены ключа приложения Change Key	14
2.4.9. Функция выбора файла приложения Select File.....	16
2.4.10. Функция чтения файла приложения Read Binary	16
2.4.11. Функция запись файла приложения Write Binary	18
2.4.12. Функция блокировки приложения Block.....	20
2.4.13. Функция разблокировки приложения Unblock	20
2.4.14. Чтение внутренних данных приложения Get Data.....	21
3. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ.....	23
3.1. Минимальный состав программно-аппаратных средств	23
3.2. Требования к составу периферийных устройств	23
3.3. Требования к персоналу (оператору).....	23
4. ПРИМЕР РАБОТЫ С ПРИЛОЖЕНИЕМ.....	24
4.1. Загрузка приложения SCOneLoyalty.....	24
4.2. Персонализация экземпляра приложения SCOneLoyalty	25
4.3. Чтение данных из экземпляра приложения SCOneLoyalty	28

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	Микроконтроллер (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 SCOneLoyalty представляет собой файловую систему, обеспечивающую возможность хранения данных о программах лояльности на микропроцессорной Java-карте, считывание и запись данных, а также изменение данных в постэмиссионный период.

Приложение SCOneLoyalty предоставляет следующие функциональные возможности:

- хранение в виде файлов данных о программах лояльности таких, как:
 - идентификатор держателя в программе лояльности;
 - данные штрих-кода или QR-кода;
 - идентификатор торгово-сервисного предприятия(ий);
 - размер скидки / процент скидки / величина кэшбэка / накопленные бонусные баллы / купон;
 - номенклатура аукционных товаров;
 - срок действия программы лояльности;
 - расчетный период;
 - дополнительные условия, характерные для программы лояльности.
- 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 SCOneLoyalty – 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 SCOneLoyalty 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 социального номера
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'	

Поле	Значение	Описание
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	'000000000 0000000000 017'	Номер карты

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
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;
- генерация пароля приложения, Internal Authenticate;
- генерация криптограммы приложения, Generate AC;
- дешифрование, Decrypt;
- 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

Поле	Значение	Описание
P1	'00'	
P2	'XX'	Номер ключа для аутентификации
P3	'08'	Длина криптограммы
Data	'XXXX...'	Криптограмма хоста

Ответные данные

Отсутствуют

2.4.4. Функция генерация пароля приложения Internal Authentica

Описание команды

Генерация одноразового пароля.

Структура команды

Поле	Значение	Описание
CLA	'80'	
INS	'88'	INTERNAL AUTHENTICATE
P1	'00'	
P2	'00'	
P3	'08'	Длина блока данных
Data	'XXXX...'	Случайное число 8 байт (Challenge)

Ответные данные

Имя	Длина (байт)	Описание
Key Version	1	Версия ключа
Host cryptogram	8	Криптограмма

2.4.5. Функция генерация криптограммы приложения Generate AC

Описание команды

Генерация криптограммы приложения.

Структура команды

Поле	Значение	Описание
CLA	'80'	
INS	'AE'	GENERATE AC
P1	'00'	
P2	'00'	
P3	'XX'	Длина блока данных
Data	'XXXX...'	Данные для расчета криптограммы приложения

Ответные данные

Имя	Длина (байт)	Описание
Key Version	1	Версия ключа
Cryptogram number	2	Номер криптограммы
Host cryptogram	8	Криптограмма

2.4.6. Функция дешифрования Decrypt

Описание команды

Команда позволяет расшифровывать данные. Одной командой можно расшифровать до 248 байт данных. Если размер данных превышает 248 байт, то можно построить цепочку команд. Если посылается одна команда, то параметр P2 должен быть равен INIT | FINAL. Если посылается цепочка из двух команд, то в первой команде P2 = INIT, а во второй P2 = FINAL. Цепочка из 3-х и более команд: в первой команде P2 = INIT, во второй и последующих (кроме последней) P2 = CONTINUE, P2 последней команды должен быть равен FINAL. Флаг FINAL обрезает паддинг '80...' расшифрованных данных.

Структура команды

Поле	Значение	Описание
CLA	'80'	
INS	'2A'	DECRYPT
P1	'XX'	Механизм шифрования ECB = '00', CBC = '01'
P2	'00'	INIT = '01', CONTINUE = '02', FINAL = '04'

Поле	Значение	Описание
P3	‘XX’	Длина блока данных (должна быть кратна 8)
Data	‘XXXX...’	Зашифрованные данные

Ответные данные

Имя	Длина (байт)	Описание
Data		Расшифрованные данные. Длина может отличаться от длины зашифрованных данных в случае наличия паддинга в конце финального блока.

2.4.7. 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)

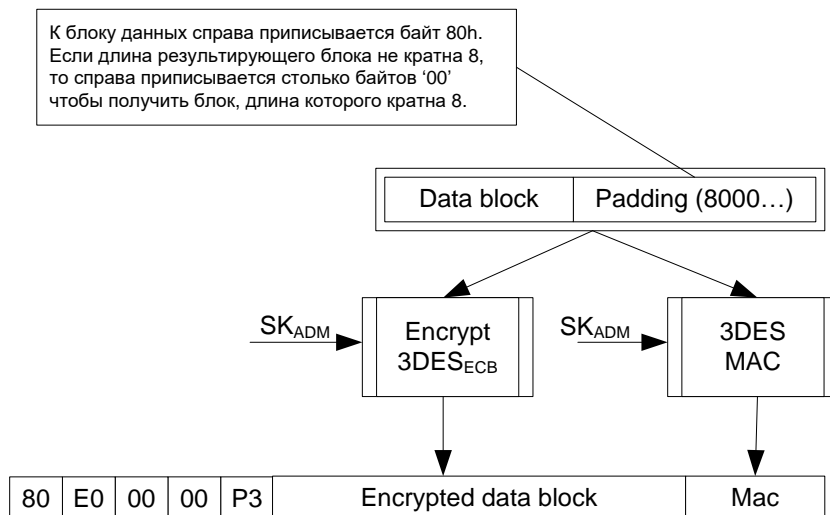
SSSS – размер файла (2 байта)

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

2.4.8. 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

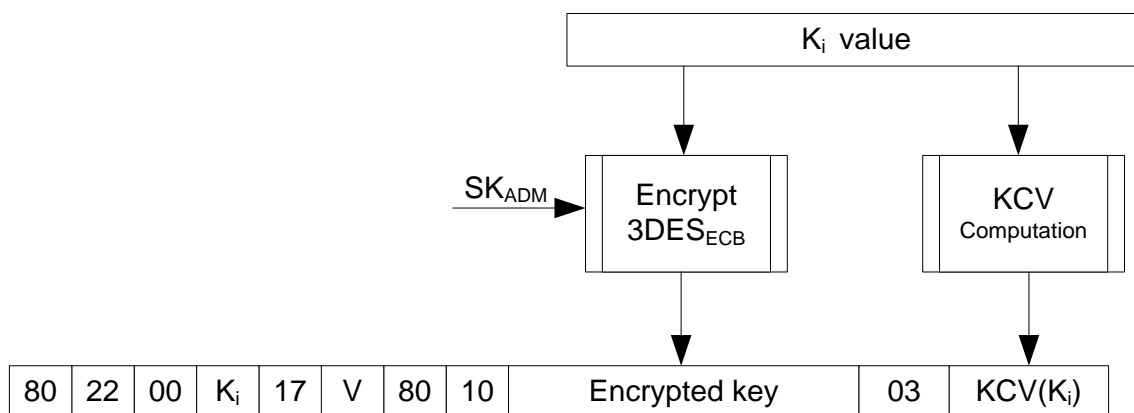
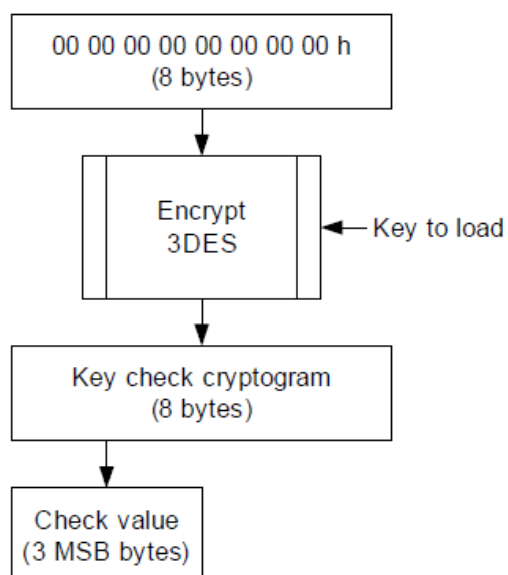


Схема расчета проверочного число ключа (KCV Computation)



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

2.4.9. 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

2.4.10. 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)

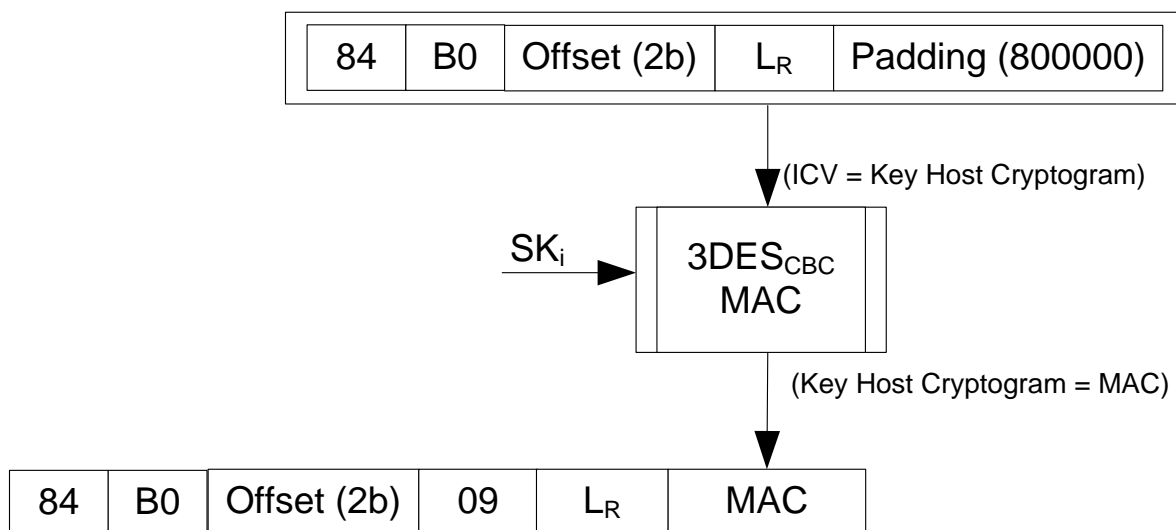
Структура команды с подписью

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' - Длина 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

2.4.11. 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.

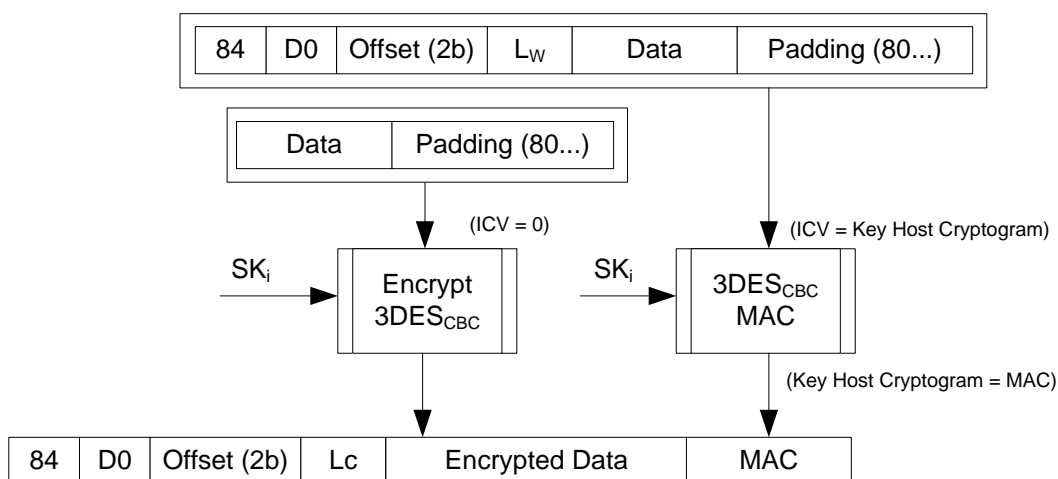
Структура команды без подписи

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 L_c 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

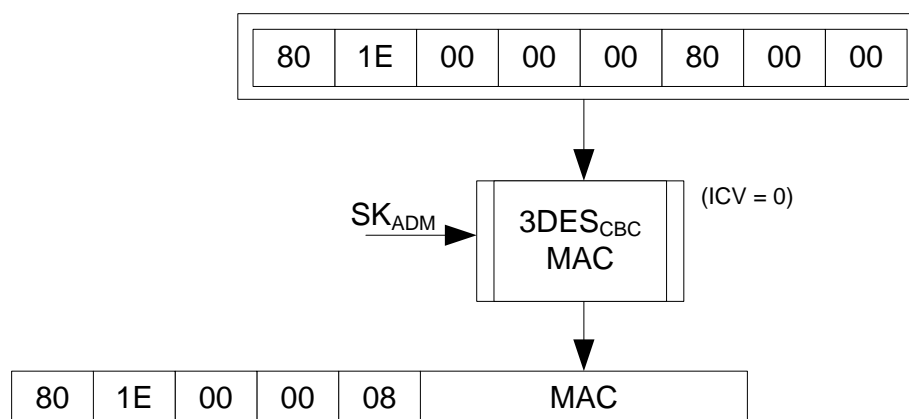
Отсутствуют

2.4.12. 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

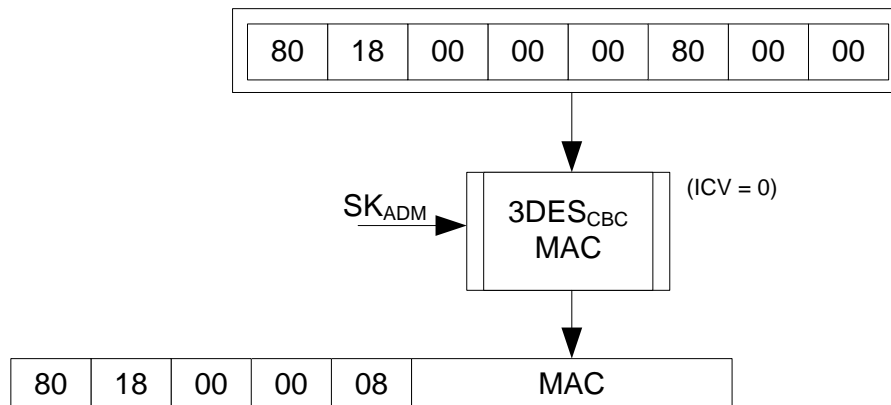
2.4.13. Функция разблокировки приложения Unblock

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

2.4.14. 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 SCONeLoyalty представляет собой бинарный файл формата «сар», 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
<i>Программные требования</i>	
Java Card SDK	2.2.1
GlobalPlatform Card Specification	v2.1.1 – v1.0
<i>Аппаратные требования</i>	
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 КВ
Объем ОЗУ\RAM	6 КВ

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 SCONeLoyalty

Ниже представлен пример загрузки пакета приложения SCONeLoyalty 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		
	←INSATLL FOR LOAD AID 53434F6E654C6F79616C7479		Загрузка пакета AID 53434F6E654C6F7961 6C7479
	00 + SW 9000 →		
	← LOAD DATA [80E80000EExx]		
	SW 9000 →		
	← LOAD DATA [80E80001EExx]		
	SW 9000 →		
	...		
	← LOAD DATA [80E80023EExx]		
	SW 9000 →		

МК с ОС Java Card	APDU интерфейс	Управлявшее прикладное ПО и устройство работы со смарт-картами	Комментарий
	← 80E88024070D050708080814		Последняя команды загрузки данных пакета
	00 + SW 9000 →		

4.2. Персонализация экземпляра приложения SCLoyalty

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

SCLoyalty	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		

SCOneLoyalty	APDU интерфейс	Управлявшее прикладное ПО и устройство работы со смарт-картами	Комментарий
	<p>←INSATLL FOR INSATLL and MAKE SELECTABLE</p> <p>80E60C00330C53434F6E654C6F79616C74790D53434F6E654C6F79616C7479010E53434F6E654C6F79616C74790101010005C903052B0000</p>		<p>Инсталляция экземпляра апплета AID</p> <p>53434F6E654C6F79616C74790101</p>
	<p>00 + SW 9000 →</p>		
	<p>Персонализация экземпляра апплета</p>		
	<p>← SELECT AID</p> <p>53434F6E654C6F79616C74790101</p>		<p>Выбор экземпляра апплета AID</p> <p>53434F6E654C6F79616C74790101</p>
	<p>FCI + SW 9000 →</p>		
	<p>← INITIALIZE UPDATE</p>		
	<p>DATA + SW 9000 →</p>		
	<p>←EXTERNAL AUTHENTICATE</p>		
	<p>SW 9000 →</p>		
	<p>← PUT DATA DF 11</p> <p>[80DADF110B00000000000000000000000017]</p>		<p>Установка тега DF11, серийный номер</p>
	<p>SW 9000 →</p>		
	<p>← PUT KEY 00</p> <p>[80D8000017018010xx..xx]</p>		<p>Установка ключа K_FILE 0</p>
	<p>SW 9000 →</p>		
	<p>← PUT KEY 10</p> <p>[80D8001017018010xx..xx]</p>		<p>Установка ключа K_OTP</p>
	<p>SW 9000 →</p>		
	<p>← PUT KEY 11</p> <p>[80D8000117018010xx..xx]</p>		<p>Установка ключа K_AC</p>
	<p>SW 9000 →</p>		
	<p>← PUT KEY 12</p> <p>[80D8001217018010xx..xx]</p>		<p>Установка ключа K_DATA</p>
	<p>SW 9000 →</p>		

SCOneLoyalty	APDU интерфейс	Управлявшее приложение ПО и устройство работы со смарт-картами	Комментарий
	← 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		
	← 80F0400F00		Установка статуса персонализировано
	SW 9000 →		
	Загрузка пользовательских данных		
	← SELECT AID 53434F6E654C6F79616C74790101		экземпляра апплета AID 53434F6E654C6F7961 6C74790101
	FCI + SW 9000 →		
	← INITIALIZE UPDATE		
	DATA + SW 9000 →		
	← EXTERNAL AUTHENTICATE [8082000008xx..xx]		
	SW 9000 →		
	← CREATE FILE DF11 [80E0000020xx..xx]		Запись содержимого файла DF11
	SW 9000 →		
	← CREATE FILE DF21 [80E00000F8xx..xx]		Запись содержимого файла DF21
	SW 9000 →		
	← SELECT FILE ID 0002 [80A4000204]		Выбор файла ID 0002 - DF 21
	DATA+SW 9000 →		Данные атрибутов файла

SCOneLoyalty	APDU интерфейс	Управлявшее прикладное ПО и устройство работы со смарт-картами	Комментарий
	← INITIALIZE UPDATE		Построение безопасного канала
	DATA + SW 9000 →		
	← EXTERNAL AUTHENTICATE [8082000008xx..xx]		
	SW 9000 →		
	← 84D000E7F0		
	← WRITE BINARY [84D000E7Fxx..xx]		Перезапись данных в файле DF 21
	DATA + SW 9000 →		
	← WRITE BINARY [84D001CExx..xx]		Перезапись данных в файле DF 21
	DATA + SW 9000 →		

4.3. Чтение данных из экземпляра приложения SCLoyalty

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

SCOneLoyalty	APDU интерфейс	Управлявшее прикладное ПО и устройство работы со смарт-картами	Комментарий
	Чтение данных приложения		
	← Reset		
	ATR →		
	← SELECT AID 53434F6E654C6F79616C74790101		экземпляра апплета AID 53434F6E654C6F79616C74790101
	FCI + SW 9000 →		
	← GET DATA [80CADF1100]		Запрос тега DF11 серийный номер

SCLoyalty	APDU интерфейс	Управлявшее прикладное ПО и устройство работы со смарт-картами	Комментарий
	DF110B000000000000000000000017 + SW 9000 →		
	← SELECT FILE ID 0001 80A4000104		Запись содержимого файла ID 0001 , DF11
	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
	DF110D3935323031323334353637 3838 + SW 9000 →		
	← SELECT FILE ID 0002 [80A4000204]		Выбор файла ID 0002 - DF 21
	DATA+SW 9000 →		Данные атрибутов файла
	← INITIALIZE UPDATE		Построение безопасного канала
	DATA + SW 9000 →		
	← EXTERNAL AUTHENTICATE [8082000008xx..xx]		
	SW 9000 →		
	← READ BINARY [84B0000009 E7 xx..xx]		Чтение данных файла, сдвиг x00
	DATA+SW 9000 →		
	← READ BINARY [84B0E70009 E7 xx..xx]		Чтение данных файл, сдвиг xE7
	DATA+SW 9000 →		
	← READ BINARY [84B001C009 98 xx..xx]		Чтение данных файл, сдвиг x1CE

SCLoyalty	APDU интерфейс	Управлявшее прикладное ПО и устройство работы со смарт-картами	Комментарий
	DATA+SW 9000 →		