ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 7816-4— 2013

Карты идентификационные

КАРТЫ НА ИНТЕГРАЛЬНЫХ СХЕМАХ

Часть 4

Организация, защита и команды для обмена

ISO/IEC 7816-4:2005
Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange (IDT)

Издание официальное

Москва
Стандартинформ
2014

Предисловие

1 ПОДГОТОВЛЕН Федеральным государственным унитарным предприятием «Всероссийский научно-исследовательский институт стандартизации и сертификации в машиностроении» (ВНИИНМАШ) и Техническим комитетом по стандартизации ТК 22 «Информационные технологии» на основе собственного аутентичного перевода на русский язык стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 22 «Информационные технологии»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 22 ноября 2013 г. № 1630-ст

4 Настоящий стандарт идентичен международному стандарту ИСО/МЭК 7816-4:2005 «Карты идентификационные. Карты на интегральных схемах. Часть 4. Организация, защита и команды для обмена» (ISO/IEC 7816-4:2005 «Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange»), включая изменение А1:2008, которое выделено в тексте одинарной линией.

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им национальные стандарты Российской Федерации, сведения о которых приведены в дополнительном приложении ДА

5 ВЗАМЕН ГОСТ Р ИСО МЭК 7816-4—2004

6 Некоторые положения международного стандарта, указанного в пункте 4, могут являться объектом патентных прав. Международная организация по стандартизации (ИСО) и Международная электротехническая комиссия (МЭК) не несут ответственности за идентификацию подобных патентных прав

Правила применения настоящего стандарта установлены в ГОСТ 1.0—2012 (раздел 8). Информация об изменениях к настоящему стандарту публикуется в годовом (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячно издаваемом информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте национального органа российской Федерации по стандартизации в сети Интернет (gost.ru).

Содержание

Введение

Настоящий стандарт — один из серии стандартов, описывающих параметры карт на интегральных схемах и их применение для обмена информацией.

Данные карты представляют собой идентификационные карты, предназначенные для обмена информацией между внешним устройством и интегральной схемой карты. В ходе обмена карта поставляет информацию (результаты вычислений, хранимые данные) и (или) изменяет свое содержимое (память данных, память событий).

Пять стандартов из серии ИСО/МЭК 7816 относятся к картам с гальваническими контактами, а три из них определяют электрический интерфейс:

ИСО/МЭК 7816-1 — определяет физические характеристики карт с контактами;

ИСО/МЭК 7816-2 — определяет размеры и расположение контактов;

ИСО/МЭК 7816-3 — определяет электрический интерфейс и протоколы передачи для асинхронных карт;

ИСО/МЭК 7816-10 — определяет электрический интерфейс и ответ на восстановление для синхронных карт;

ИСО/МЭК 7816-12 — определяет электрический интерфейс и рабочие процедуры для USB карт.

Все остальные стандарты не зависят от технологии физического интерфейса. Они применяются к картам, имеющим доступ при помощи контактов и/или радиочастоты:

ИСО/МЭК 7816-4 — определяет организацию, защиту и команды для обмена информацией;

ИСО/МЭК 7816-5 — определяет регистрацию провайдеров прикладных программ;

ИСО/МЭК 7816-6 — определяет элементы данных для межотраслевого обмена;

ИСО/МЭК 7816-7 — определяет команды языка структурированных запросов для карты;

ИСО/МЭК 7816-8 — определяет команды, обеспечивающие операции по защите информации;

ИСО/МЭК 7816-9 — определяет команды для управления картами;

ИСО/МЭК 7816-11 — определяет верификацию личности биометрическими методами;

ИСО/МЭК 7816-13 — определяет команды для управления приложением в мульти-прикладной среде.

ИСО/МЭК 7816-15 — определяет приложение с криптографической информацией.

ИСО/МЭК 10536 [13] определяет доступ при помощи поверхностного действия. ИСО/МЭК 14443 [15] и ИСО/МЭК 15693 [17] определяют радиочастотный доступ. Такие карты известны также как бесконтактные карты.

ИСО и МЭК обращают внимание на заявление о том, что соответствие настоящему стандарту может включать использование следующих патентов и иностранных эквивалентов:

JPN 2033906 Portable electronic device;

JPN 2557838 Integrated circuit card;

JPN 2537199 Integrated circuit card;

JPN 2856393 Portable electronic device;

JPN 2137026 Portable electronic device;

JPN 2831660 Portable electronic device;

DE 198 55 596 Portable microprocessor-assisted data carrier that can be used with or without contacts.

ИСО и МЭК не занимают никакой позиции относительно наличия, действительности и области применения данных патентных прав.

Обладатели данных патентных прав заверили ИСО и МЭК, что они готовы вести переговоры с претендентами со всего мира о предоставлении лицензии на разумных и не недискриминационных условиях, включая сроки. В связи с этим, утверждение владельцев данных патентных прав зарегистрировано в ИСО и МЭК. Информацию можно получить у следующих компаний:

Контакты	Данные патента
Toshiba Corporation Intellectual Property Division 1-1, Shibaura 1-Chome Minato-ku, Tokyo 105-8001, Japan	JPN 2033906 (дата приоритета: 1986-02-18; дата опубликования: 1996-03-19), FRA 8614996, KOR 44664 JPN 2557838 (дата приоритета: 1986-02-18; дата опубликования: 1996-09-05), FRA 8700343, GER 3700504, KOR 42243, USA 4841131 JPN 2537199 (дата приоритета: 1986-06-20; дата опубликования: 1996-07-08), FRA 8708646, FRA 8717770, USA 4833595, USA 4901276 JPN 2856393 (дата приоритета: 1987-02-17; дата опубликования: 1998-11-27), FRA 8801887, KOR 43929, USA 4847803 JPN 2137026 (дата приоритета: 1987-02-20; дата опубликования: 1998-06-26), JPN 3054119, FRA 8802046, KOR 44393, USA 4891506 JPN 2831660 (дата приоритета: 1988-08-26; дата опубликования: 1998-09-25), FRA 8911249, KOR 106290, USA 4988855
Orga Kartensysteme Gmbh Am Hoppenhof 33 D-33104 Paderborn Germany	DE 198 55 596 (дата приоритета: 1998-12-02; дата опубликования: 2000-06-29) Applications pending in Europe, Russia, Japan, China, USA, Brazil, Australia

Международный стандарт ИСО/МЭК 7816-4 подготовлен подкомитетом № 17 «Карты и идентификация личности» совместного технического комитета № 1 ИСО/МЭК «Информационные технологии» (ISO/IEC JTC 1/SC 17).

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Карты идентификационные

КАРТЫ НА ИНТЕГРАЛЬНЫХ СХЕМАХ

Часть 4

Организация, защита и команды для обмена

Identification cards. Integrated circuit cards. Part 4.
Organization, security and commands for interchange

Дата введения — 2015—01—01

1 Область применения

Настоящий стандарт устанавливает:

- содержание пар команда-ответ, передаваемых между устройством сопряжения и картой;
- средства извлечения элементов данных и информационных объектов из карты;
- структуры и содержание байтов предыстории для описания рабочих характеристик карт;
- структуры для приложений и данных в карте, прослеживаемые на стыке между картой и устройством сопряжения при обработке команд;
- методы доступа к файлам и данным, содержащимся в карте;
- архитектуру безопасности, определяющую права доступа к файлам и данным, содержащимся в карте;
- средства и механизмы для идентификации и способов адресации приложений, содержащихся в карте;
- методы безопасного обмена сообщениями;
- методы доступа к алгоритмам, обрабатываемым картой (исключая описание самих алгоритмов);

Стандарт не распространяется на реализацию обмена данными внутри карты или внешнего устройства.

Настоящий стандарт не зависит от технологии физического сопряжения. Он применяется к картам, имеющим доступ при помощи одного или нескольких следующих методов: контактов, поверхностного действия и радио частоты.

2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие стандарты. Для датированных ссылок следует использовать только указанное издание, для недатированных ссылок следует использовать последнее издание указанного документа, включая все поправки:

ИСО/МЭК 7816-3 карты идентификационные. Карты на интегральных схемах. Часть 3. Карты с контактами. Электрический интерфейс и протоколы передачи (ISO/IEC 7816-3, Identification cards — Integrated circuit cards — Part 3: Cards with contacts — Electrical interface and transmission protocols»).

ИСО/МЭК 7816-6 карты идентификационные. Карты на интегральных схемах. Часть 6. Межотраслевые элементы данных для обмена информацией (ISO/IEC 7816-6, Identification cards — Integrated circuit cards — Part 6: Interindustry data elements for interchange).

ИСО/МЭК 8825-1:2002[1)] Информационная технология. Правила кодирования АСН.1. Часть 1. Спецификация базовых (BER), канонических (CER) и отличительных (DER) правил кодирования (ISO/IEC 8825-1:2002, Information technology — ASN.1 encoding rules. Part 1: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER))

[1)] Заменен на ИСО/МЭК 8825-1:2008.

Издание официальное

3 Термины и определения

В настоящем стандарте используют следующие определения:

3.1 **правило доступа** (access rule): Элемент данных, который содержит метод доступа, связанный с действием, и условия секретности, которым необходимо соответствовать перед началом действия.

3.2 **файл Ответа-на-Восстановление** (Answer-to-Reset file): Дополнительный EF, который указывает рабочие характеристики карты.

3.3 **приложение** (application): Структуры, элементы данных и программные модули, необходимые для выполнения определенных функций.

3.4 **DF приложения** (application DF): Структура, принимающая приложение в карте.

3.5 **идентификатор приложения** (application identifier): Элемент данных (до шестнадцати байт), который идентифицирует приложение.

3.6 **метка приложения** (application label): Элемент данных для использования в человеко-машинном интерфейсе.

3.7 **провайдер приложения** (application provider): Организация, предоставляющая компоненты, которые составляют приложение в карте.

3.8 **шаблон приложения** (application template): Множество информационных объектов, относящихся к приложению и включающих один информационный объект «идентификатор приложения».

3.9 **ассиметричный криптографический метод** (asymmetric cryptographic technique): Метод криптографии, который использует две взаимосвязанные операции: открытую операцию, определенную открытым числом или открытым ключом, и приватную операцию, определенную приватным числом или приватным ключом (две операции обладают таким свойством, что при наличии открытой операции вычислительно невозможно определить приватную операцию).

3.10 **сертификат** (certificate): Цифровая подпись, связывающая конкретного человека или объект с его соответствующим открытым ключом (организация, выдающая сертификат, также действует в качестве органа, распределяющего теги для элементов данных в сертификате).

3.11 **пара команда-ответ** (command-response pair): Набор из двух сообщений на стыке сопряжения: командного APDU и следующего за ним ответного APDU в противоположном направлении.

3.12 **элемент данных** (data element): Смысловой элемент информации, прослеживаемый на стыке между картой и устройством сопряжения, для которого определены наименование, описание логического содержания, формат и кодирование.

3.13 **информационный объект** (data object): Информация, прослеживаемая на стыке между картой и устройством сопряжения, состоящая из сцепления обязательного поля тега, обязательного поля длины и условного поля значения.

3.14 **единица данных** (data unit): Наименьший набор битов, на который можно дать однозначную ссылку в пределах EF, поддерживающего единицу данных.

3.15 **назначенный файл** (dedicated file): Структура, содержащая контрольную информацию файла и, возможно, свободную память для распределения.

3.16 **имя DF** (DF name): Элемент данных (до шестнадцати битов), который уникальным образом идентифицирует DF в карте.

3.17 **цифровая подпись** (digital signature): Присоединенные данные или криптографически преобразованные данные, позволяющие получателю данных подтвердить источник и целостность данных и защитить их от подделки, например получателем.

3.18 **справочный файл** (directory file): Дополнительный EF, содержащий список приложений, поддерживаемых картой, и дополнительно связанные элементы данных.

3.19 **элементарный файл** (elementary file): Набор единиц данных или записей, или информационных объектов, которые совместно используют один и тот же идентификатор файла и один(ни) и тот(те) же атрибут(ы) секретности.

3.20 **файл** (file): Структура для приложения и/или для данных, имеющихся в карте, прослеживаемая на стыке между картой и устройством сопряжения при обработке команд.

3.21 **идентификатор файла** (file identifier): Элемент данных (двухбайтовый), используемый для обращения к файлу.

3.22 **список заголовков** (header list): Сцепление пар полей тегов и полей длины без разграничения.

3.23 **идентификационная карта** (identification card): Карта, которая содержит данные о ее держателе и эмитенте и может содержать сведения, необходимые в качестве входных данных для применения карты в соответствии с ее назначением и выполнения основанных на них транзакций.

[ИСО/МЭК 7810 [2]]

3.24 **внутренний элементарный файл** (internal elementary file): EF для хранения данных, интерпретируемых картой.

3.25 **ключ** (key): Последовательность символов управления криптографической операцией (например, операциями шифрования и дешифрования, приватной или открытой операциями в динамической аутентификации, получением подписи, верификацией подписи).

3.26 **главный файл** (master file): Уникальный DF, представляющий собой корень файловой структуры в карте, использующей иерархию файлов DF.

3.27 **смещение** (offset): Число последовательных обращений к единице данных в EF, который поддерживает эту единицу данных, или байт при записи.

3.28 **родительский файл** (parent file): DF, непосредственно предшествующий заданному файлу в пределах иерархии файлов DF.

3.29 **пароль** (password): Данные, которые могут быть затребованы приложением от пользователя карты для аутентификации.

3.30 **путь** (path): Сцепление идентификаторов файлов без разграничения.

3.31 **приватный ключ** (private key): Ключ субъекта пары ассиметричных ключей, который может использовать только этот субъект.

[ИСО/МЭК 9798-1 [8]]

3.32 **провайдер** (provider): Орган, имеющий или получивший право на создание DF в карте.

3.33 **открытый ключ** (public key): Ключ субъекта пары ассиметричных ключей, который может быть сделан общедоступным.

[ИСО/МЭК 9798-1 [8]]

3.34 **запись** (record): Строка байтов, к которой карта обращается и которую карта обрабатывает в пределах EF, поддерживающего запись.

3.35 **идентификатор записи** (record identifier): Числовое значение, используемое для обращения к одной или более записей в пределах EF, поддерживающего запись.

3.36 **номер записи** (record number): Порядковый номер, который однозначно идентифицирует каждую запись в пределах EF, поддерживающего запись.

3.37 **зарегистрированный идентификатор провайдера приложения** (registered application provider identifier): Элемент данных (из пяти байтов), который однозначно идентифицирует провайдер приложения.

3.38 **секретный ключ** (secret key): Ключ, используемый в методах симметричной криптографии при помощи установки определенных объектов.

[ИСО/МЭК 11770-3 [14]]

3.39 **безопасный обмен сообщениями** (secure messaging): Совокупность методов для криптографической защиты [частей] пары команда-ответ.

3.40 **атрибут секретности** (security attribute): Условие использования объектов, имеющихся в карте, включающих записанные данные и функции обработки данных, выраженные как элемент данных, содержащий одно или несколько правил доступа.

3.41 **безопасная среда** (security environment): Совокупность компонентов, необходимых для приложения в карте, для обеспечения безопасного обмена сообщениями или для операций по защите.

3.42 **ассиметричный криптографический метод** (asymmetric cryptographic technique): Криптографический метод, который использует один и тот же секретный ключ, как для операций отправителя, так и для операций получателя (без секретного ключа невозможно путем вычислений определить обе операции).

3.43 **список тегов** (tag list): Сцепление полей тегов без разграничения.

3.44 **шаблон** (template): Множество информационных объектов BER-TLV для формирования поля значения составного информационного объекта BER-TLV.

3.45 **рабочий элементарный файл** (working elementary file): EF для хранения данных, не интерпретируемых картой.

4 Сокращения и обозначения

В настоящем стандарте применены следующие сокращения.

AID — идентификатор приложения (application identifier);

APDU — блок данных прикладного протокола (application protocol data unit));

ARR — указатель правила доступа (access rule reference);

ASN.1 — абстрактно-синтаксическая нотация версии 1 (abstract syntax notation one) (см. ИСО/МЭК 8825-1).

AT — шаблон управляющих ссылок аутентификации (control reference template for authentication);

ATR — Ответ-на-Восстановление (Answer-to-Reset);

BER — базовые правила кодирования абстрактно-синтаксической нотации версии 1 (АСН.1) (basic encoding rules of ASN.1) (см. ИСО/МЭК 8825-1);

CCT — шаблон управляющих ссылок «криптографическая контрольная сумма» (control reference template for cryptographic checksum);

CLA — байт класса (class byte);

CRT — шаблон управляющих ссылок (control reference template);

CT — шаблон управляющих ссылок конфиденциальности (control reference template for confidentiality);

DF — назначенный файл (dedicated file);

DIR — директория (directory);

DST — шаблон управляющих ссылок цифровой подписи (control reference template for digital signature);

EF — элементарный файл (elementary file);

EF.ARR — файл указателя правил доступа (access rule reference file);

EF.ATR — файл Ответа-на-Восстановление (Answer-to-Reset file);

EF.DIR — справочный файл (directory file);

FCI — контрольная информация файла (file control information);

FCP — контрольный параметр файла (file control parameter);

FMD — данные управления файлом (file management data);

HT — шаблон управляющих ссылок хэш-кода (control reference template for hash-code);

INS — командный байт (instruction byte);

KAT —шаблон управляющих ссылок для согласования ключей (control reference template for key agreement);

L_c field — поле L_c (поле длины для кодирования числа N_c);

LCS byte — байт состояния жизненного цикла (life cycle status byte).

Le field — поле L_e (поле длины для кодирования числа N_e);

MF — главный файл (master file);

N_c — число байтов в поле данных команды;

N_e — максимальное число байтов, ожидаемое в поле данных ответа;

N_r — число байтов в поле данных ответа;

PIX — проприетарное расширение идентификатора приложения (proprietary application identifier extension);

P1-P2 — байты параметров (parameter bytes (указаны для ясности, тире не существенно));

RFU — зарезервировано для использования в будущем (reserved for future use);

RID — зарегистрированный идентификатор провайдера приложения (registered application provider identifier);

SC — условие секретности (security condition);

SCQL — язык структурированных запросов для карты (structured card query language);

SE — безопасная среда (security environment);

SEID byte — байт идентификатора безопасной среды (security environment identifier byte);

SM — безопасный обмен сообщениями (secure messaging);

SW1-SW2 — байты состояния (status bytes (указаны для ясности, тире не существенно)),

(SW1-SW2) — значение сцепления байтов SW1 и SW2 (первый байт является старшим значащим байтом);

TLV — тег, длина, значение (tag, length, value);

{T-L-V} — информационный объект (указан для ясности, дефис и фигурные скобки не существенны);

'XX' — обозначение, использующее шестнадцатеричные цифры от '0' до '9' и от 'A' до 'F', равно XX по основанию 16.

5 Организация обмена информацией

Для организации обмена информацией в настоящем разделе определены следующие основные компоненты:

1) Пары команда-ответ;
2) Информационные объекты;
3) Структуры для приложений и данных;
4) Архитектура безопасности.

5.1 Пары команда-ответ

В таблице 1 приведена пара команда-ответ, а именно командный APDU, за которым следует ответный APDU в противоположном направлении (см. ИСО/МЭК 7816-3). Чередования пары команды-ответа на стыке сопряжения не должно быть, т. е. ответный APDU должен быть получен до инициализации другой пары команды-ответа.

Таблица 1 — Пара команда-ответ

Поле	Число байтов	Число байтов
Заголовок команды	Байт класса CLA	1
	Командный байт INS	1
	Байты параметров P1-P2	2
Поле L_c	Отсутствует для кодирования $N_c = 0$,	0, 1 или 3
	присутствует для кодирования $N_c > 0$	
Поле данных команды	Отсутствует, если $N_c = 0$, присутствует как строка байтов N_c, если $N_c > 0$	N_c
Поле L_e	Отсутствует для кодирования $N_e = 0$, присутствует для кодирования $N_e > 0$	0, 1, 2 или 3
Поле данных ответа	Отсутствует, если $N_r = 0$, присутствует как строка байтов N_r, если $N_r > 0$	N_r (не более чем N_e)
Завершитель ответа	Байты состояний SW1-SW2	2

В любой паре команда-ответ, включающей в себя и поле L_e и поле L_c (см. ИСО/МЭК 7816-3), короткое и расширенное поля длины не должны складываться: либо оба поля длины короткие, либо оба поля длины расширенные.

Если в карте явно заявлена возможность обрабатывать «расширенные поля L_c и L_e» (см. таблицу 88, третья таблица программных функций) в байтах предыстории (см. 8.1.1) или EF.ATR (см. 8.2.1.1), то карта обрабатывает короткое и расширенное поле длины. В противном случае (значение по умолчанию), карта работает только с короткими полями длины.

N_c обозначает число байтов в поле данных команды. Поле L_c кодирует N_c.

- Если поле L_c отсутствует, то N_c равно нулю;
- короткое поле L_c состоит из одного байта, которое отлично от '00' и принимает значения:
- от '01' до 'FF'. Этот байт кодирует N_c от одного до 255;
- Расширенное поле L_e состоит из трех байтов: одного байта, установленного на '00', за которым следуют два байта, которые отличны от '0000', и принимают значения:
- от '0001' до 'FFFF'. Эти два байта кодируют N_c от одного до 65535.

N_e обозначает максимальное число байтов, ожидаемое в поле данных ответа. Поле L_e кодирует N_e.

Если поле L_e отсутствует, то N_e равно нулю;

короткое поле L_e состоит из одного байта с произвольным значением:

от '01' до 'FF'. Этот байт кодирует N_e от одного до 255;

если этот байт установлен на '00', то N_e равно 256.

- Расширенное поле L_e состоит либо из трех байтов (один байт установлен на '00', за которым следуют два байта с произвольным значением), если поле L_e отсутствует, либо из двух байтов (с произвольным значением), если расширенное поле L_e присутствует:

- от '0001' до 'FFFF'. Эти два байта кодируют N_e от одного до 65535;

- если два байта установлены на '0000', то N_e равно 65536.

N_r обозначает число байтов в поле данных ответа. N_r должно быть меньше или равно N_e. Поэтому в любой паре команда-ответ отсутствие поля L_e является обычным способом получения поля данных без ответа. Если поле L_e содержит только установленные на '00' байты, то значение N_e максимально, т. е. в пределах 256 для короткого поля L_e или в пределах 65356 для расширенного поля L_e, при этом все имеющиеся байты должны быть возвращены.

Если операция прерывается, то карта может стать невосприимчивой. Однако если возникает ответный APDU, то поле данных ответа должно отсутствовать и SW1-SW2 должны показывать ошибку.

P1-P2 указывают элементы управления и опции для обработки команд. Байт параметров, установленный на '00' обычно не обеспечивает дальнейшее уточнение. Других общих правил для кодирования байтов параметров не существует.

Ниже установлены общие правила для кодирования байта класса CLA (см. 5.1.1), командного байта INS (см. 5.1.2) и байтов состояний (см. 5.1.3). В этих байтах биты, зарезервированные для использования в будущем, должны быть установлены на 0, пока не будет определено иначе.

5.1.1Байт класса

CLA указывает класс команды. С учетом требований, определенных в ИСО/МЭК 7816-3, значение 'FF' является недействительным. Бит 8 в CLA проводит различие между межотраслевым классом и проприетарным классом.

Бит 8, установленный на 0, указывает межотраслевой класс. Ниже определены значения 000x xxxx и 01xx xxxx. Значения 001x xxxx зарезервированы для использования в будущем ИСО/МЭК СТК 1/ПК 17.

В таблице 2 значения 000x xxxx определены как первые межотраслевые значения:

- Биты 8, 7 и 6 установлены на 000;

- Бит 5 управляет сцеплением команд (см. 5.1.1.1);

- Биты 4 и 3 указывают безопасный обмен сообщениями (см. 6):

- Биты 2 и 1 кодируют номер логического канала от нуля до трех (см. 5.1.1.2).

Таблица 2 — Первые межотраслевые значения CLA

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0	0	0	x	—	—	—	—	**Управление сцеплением команд** (см. 5.1.1.1)
0	0	0	0	—	—	—	—	- Команда является последней или единственной командой в цепочке;
0	0	0	1	—	—	—	—	- Команда не является последней командой в цепочке
0	0	0	—	x	x	—	—	**Индикация безопасного обмена сообщениями**:
0	0	0	—	0	0	—	—	- Нет SM или SM не указан;
0	0	0	—	0	1	—	—	- Проприетарный формат SM
0	0	0	—	1	0	—	—	- SM в соответствии с 6, заголовок команды не обрабатывается в соответствии 6.2.3.1
0	0	0	—	1	1	—	—	- SM в соответствии с 6, заголовок команды аутентифицирован в соответствии с 6.2.3.1
0	0	0	—	—	—	x	x	**Номер логического канала от нуля до трех** (см. 5.1.1.2)

В таблице 3 значения 01xx xxxx определены как последующие межотраслевые значения:

- Биты 8 и 7 установлены на 01;

- Бит 6 указывает безопасный обмен сообщениями (см. 6);

- Бит 5 управляет сцеплением команд (см. 5.1.1.1);

- Биты с 4 по 1 кодируют число от нуля до пятнадцати; данное число плюс четыре — это номер логического канала от четырех до девятнадцати (см. 5.1.1.2).

Таблица 3 — Последующие межотраслевые значения CLA

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0 0 0	1 1 1	x 0 1	— — —	— — —	— — —	— — —	— — —	**Индикация безопасного обмена сообщениями**: - Нет SM или SM не указан; - SM в соответствии с 6, заголовок команды не обрабатывается в соответствии 6.2.3.1
0 0 0	1 1 1	— — —	x 0 1	— — —	— — —	— — —	— — —	**Управление сцеплением команд** (см. 5.1.1.1) - Команда является последней или единственной командой в цепочке; - Команда не является последней командой в цепочке
0	1	—	—	x	x	x	x	**Номер логического канала от четырех до девятнадцати** (см. 5.1.1.2)

Бит 8, установленный на 1, указывает проприетарный класс, за исключением значения 'FF', которое является недействительным. Контекст приложения определяет остальные биты.

5.1.1.1 Управление сцеплением команд

Данный раздел определяет механизм, при помощи которого в межотраслевом классе последовательные пары команда-ответ могут быть соединены в цепочку. Данный механизм может быть использован при выполнении многоступенчатой операции, например, передачи слишком длинной для одной команды строки данных.

Если карта поддерживает этот механизм, то это должно быть указано (см. таблицу 88, третья таблица программных функций) в байтах предыстории (см. 8.1.1) или в EF.ATR (см. 8.2.1.1).

Настоящий стандарт определяет поведение карты только в тех случаях, когда цепочка, один раз инициированная, прерывается до инициализации пары команда-ответ, не являющейся частью цепочки. В другом случае поведение карты стандартом не определено.

Для сцепления в межотраслевом классе должен использоваться бит 5 в CLA, пока остальные семь бит остаются постоянными.

- Если бит 5 установлен на 0, то команда является последней или единственной командой в цепи;
- Если бит 5 установлен на 1, то команда не является последней командой в цепочке.

В ответе на команду, которая не является последней командой в цепочке, SW1-SW2, установленные на '9000', означают, что операция к настоящему времени завершена; предупредительная индикация запрещена (см. 5.1.3), кроме того, могут происходит следующие специфические состояния ошибки:

- Если SW1-SW2 установлены на '6883', то ожидается последняя команда в цепочке;
- Если SW1-SW2 установлены на '6884', то сцепление команд не поддерживается.

5.1.1.2 Логические каналы

В настоящем разделе определен механизм, посредством которого в межотраслевом классе пара команда-ответ может обратиться к логическим каналам.

Если карта поддерживает механизм, то она должна указывать максимальное число доступных каналов (см. таблицу 88, третья таблица программных функций) в байтах предыстории (см. 8.1.1) или в EF.ATR (см. 8.2.1.1).

Если указанное картой число доступных каналов ? четыре или меньше, то следует применять таблицу 2.

Если указанное картой число доступных каналов ? пять или более, то следует применять еще таблицу 3.

Для обращения к логическим каналам в межотраслевом классе следует применять следующие правила:

- CLA кодирует номер канала пары команда-ответ,
- основной канал должен быть постоянно доступен, т. е. он не может быть закрыт. Ему присваивается нулевой номер;
- карты, не поддерживающие данный механизм (по умолчанию), должны использовать только основной канал;
- любой другой канал может быть открыт при завершении либо команды SELECT (см. 7.1.1), при этом CLA кодирует номер еще не использованного канала, либо команды MANAGE CHANNEL с функцией открытия (см. 7.1.2);

- любой другой канал может быть закрыт при завершении команды MANAGE CHANNEL с функцией закрытия. После закрытия канал становится доступным для повторного использования;
- только один канал должен быть активен единовременно. Использование логических каналов не снимает запрет чередования пар команда-ответ на стыке интерфейса, т. е. ответный APDU должен быт получен до инициализации другой пары команда-ответ (см. 5.1);
- для одной и той же структуры можно открыть более одного канала (см. 5.3), т. е. для DF, и возможно для DF приложения, а также возможно и для EF, если это явно не исключено байтом описателя файла (см. таблицу 14).

Каждый логический канал имеет свое состояние защиты (см. 5.4). Метод распределения состояний защиты выходит за рамки настоящего стандарта.

5.1.2 Командный байт

INS указывает команду для обработки. Учитывая положения ИСО/МЭК 7816-3, значения '6X' и '9X' являются недействительными.

В таблице 4 приведены все команды, указанные в ИСО/МЭК 7816 на момент публикации.

В таблице 4.1 приведены наименования команд в алфавитном порядке.

В таблице 4.2 приведены коды INS числовом порядке.

Таблица 4.1 — Команды в алфавитном порядке

Наименование команды	INS	См.
ACTIVATE FILE	'44'	Раздел 9
APPEND RECORD	'E2'	7.3.7
CHANGE REFERENCE DATA	'24'	7.5.7
CREATE FILE	'E0'	Раздел 9
DEACTIVATE FILE	'04'	Раздел 9
DELETE FILE	'E4'	Раздел 9
DISABLE VERIFICATION REQUIREMENT	'26'	7.5.9
ENABLE VERIFICATION REQUIREMENT	'28'	7.5.8
ENVELOPE	'C2', 'C3'	7.6.2
ERASE BINARY	'0E','0F'	7.2.7
ERASE RECORD(S)	'0C'	7.3.8
EXTERNAL (/MUTUAL) AUTHENTICATE	'82'	7.5.4
GENERAL AUTHENTICATE	'86', '87'	7.5.5
GENERATE ASYMMETRIC KEY PAIR	'46'	Раздел 8
GET CHALLENGE	'84'	7.5.3
GET DATA	'CA', 'CB'	7.4.2
GET RESPONSE	'C0'	7.6.1
INTERNAL AUTHENTICATE	'88'	7.5.2
MANAGE CHANNEL	'70'	7.1.2
MANAGE SECURITY ENVIRONMENT	'22'	7.5.11

Таблица 4.2 — Команды в числовом порядке

INS	Наименование команды	См.
'04'	DEACTIVATE FILE	Раздел 9
'0C'	ERASE RECORD(S)	7.3.8
'0E', '0F'	ERASE BINARY	7.2.7
'10'	PERFORM SCQL OPERATION	Раздел 7
'12'	PERFORM TRANSACTION OPERATION	Раздел 7
'14'	PERFORM USER OPERATION	Раздел 7
'20', '21'	VERIFY	7.5.6
'22'	MANAGE SECURITY ENVIRONMENT	7.5.11
'24'	CHANGE REFERENCE DATA	7.5.7
'26'	DISABLE VERIFICATION REQUIREMENT	7.5.9
'28'	ENABLE VERIFICATION REQUIREMENT	7.5.8
'2A'	PERFORM SECURITY OPERATION	Раздел 8
'2C'	RESET RETRY COUNTER	7.5.10
'44'	ACTIVATE FILE	Раздел 9
'46'	GENERATE ASYMMETRIC KEY PAIR	Раздел 8
'70'	MANAGE CHANNEL	7.1.2
'82'	EXTERNAL (/ MUTUAL) AUTHENTICATE	7.5.4
'84'	GET CHALLENGE	7.5.3
'86', '87'	GENERAL AUTHENTICATE	7.5.5
'88'	INTERNAL AUTHENTICATE	7.5.2

Окончание таблицы 4.1

Наименование команды	INS	См
PERFORM SCQL OPERATION	'10'	Раздел 7
PERFORM SECURITY OPERATION	'2A'	Раздел 8
PERFORM TRANSACTION OPERATION	'12'	Раздел 7
PERFORM USER OPERATION	'14'	Раздел 7
PUT DATA	'DA', 'DB'	7.4.3
READ BINARY	'B0', 'B1'	7.2.3
READ RECORD (S)	'B2', 'B3'	7.3.3
RESET RETRY COUNTER	'2C'	7.5.10
SEARCH BINARY	'A0', 'A1'	7.2.6
SEARCH RECORD	'A2'	7.3.7
SELECT	'A4'	7.1.1
TERMINATE CARD USAGE	'FE'	Раздел 9
TERMINATE DF	'E6'	Раздел 9
TERMINATE EF	'E8'	Раздел 9
UPDATE BINARY	'D6', 'D7'	7.2.5
UPDATE RECORD	'DC', 'DD'	7.3.5
VERIFY	'20', '21'	7.5.6
WRITE BINARY	'D0', 'D1'	7.2.4
WRITE RECORD	'D2'	7.3.4

Окончание таблицы 4.2

INS	Наименование команды	См.
'A0', 'A1'	SEARCH BINARY	7.2.6
'A2'	SEARCH RECORD	7.3.7
'A4'	SELECT	7.1.1
'B0', 'B1'	READ BINARY	7.2.3
'B2', 'B3'	READ RECORD (S)	7.3.3
'C0'	GET RESPONSE	7.6.1
'C2', 'C3'	ENVELOPE	7.6.2
'CA', 'CB'	GET DATA	7.4.2
'D0', 'D1'	WRITE BINARY	7.2.6
'D2'	WRITE RECORD	7.3.4
'D6', 'D7'	UPDATE BINARY	7.2.5
'DA', 'DB'	PUT DATA	7.4.3
'DC', 'DD'	UPDATE RECORD	7.3.5
'E0'	CREATE FILE	Раздел 9
'E2'	APPEND RECORD	7.3.6
'E4'	DELETE FILE	Раздел 9
'E6'	TERMINATE DF	Раздел 9
'E8'	TERMINATE EF	Раздел 9
'FE'	TERMINATE CARD USAGE	Раздел 9

Примечание — В межотраслевом классе любой недействительный код INS, который не определен в серии ИСО/МЭК 7816, зарезервирован для использования в будущем ИСО/МЭК СТК 1/ПК 17.

Серия стандартов ИСО/МЭК 7816 определяет использование этих команд для межотраслевого класса.

- Настоящий стандарт (см. 7) определяет команды для обмена информацией;
- ИСО/МЭК 7816-7 [4] определяет команды языка структурированных запросов для карт (SCQL);
- ИСО/МЭК 7816-8 [4] определяет команды для операций по защите информации;
- ИСО/МЭК 7816-9 [4] определяет команды для управления картами.

В межотраслевом классе бит 1 в INS указывает следующий формат поля данных:

- если бит 1 установлен на 0 (четный код INS), то никаких указаний не предусмотрено;
- если бит 1 установлен на 1 (нечетный код INS), то кодирование BER-TLV (см. 5.2.2) должно применяться следующим образом.

- в несцепленных командах с SW1, не установленным в '61', поля данных, если имеются, должны быть закодированы в BER-TLV;
- сцепление команд и/или использование SW1, установленного на '61', позволяет сделать передачу слишком длинной строки данных за одну команду. Такая операция позволяет разделить информационные объекты на поля данных, которые посылаются в виде последовательности в одном направлении, т.е. в обратном направлении поле данных не посылают. При сцеплении команд и/или использовании SW1, установленного на '61', сцепление всех последовательных полей данных в том же направлении той же последовательности должно быть закодировано в BER-TLV.

5.1.3 Байты состояний

SW1-SW2 указывают состояния обработки. Учитывая положения ИСО/МЭК 7816 любые значения, отличные от '6XXX' и '9XXX' являются недействительными; любые значения '60XX' являются также недействительными.

Значения '61XX', '62XX', '63XX', '64XX', '65XX', '66XX', '68XX', '69XX', '6AXX' и '6CXX' являются межотраслевыми. Учитывая положения ИСО/МЭК 7816-3, значения '67XX', '6BXX', '6DXX', '6EXX', '6FXX' и '9XXX' являются проприетарными, за исключением значений '6700', '6B00', '6D00', '6E00', '6F00' и '9000', которые являются межотраслевыми.

На рисунке 1 показана структурная схема значений '9000' и от '61XX'до '6FXX' для SW1-SW2.

Рисунок 1 — Структурная схема значений SW1-SW2

В таблице 5 перечислены межотраслевые значения SW1-SW2 и представлено их общее смысловое содержание. ИСО/МЭК СТК 1/ПК 17 зарезервировал для использования в будущем межотраслевые значения SW1-SW2, не определенные в серии ИСО/МЭК 7816.

Таблица 5 — Общее смысловое содержание межотраслевых значений SW1-SW2

Состояние обработки	SW1-SW2	Смысловое содержание
Нормальная обработка	'9000'	Нет дальнейшего уточнения
	'61XX'	Байт SW2 кодирует число еще доступных байтов данных (см. текст ниже)
Обработка с предупреждением	'62XX'	Состояние энергонезависимой памяти без изменений (дальнейшее уточнение в SW2)
	'63XX'	Состояние энергонезависимой памяти изменено (дальнейшее уточнение в SW2)
Ошибка выполнения	'64XX'	Состояние энергонезависимой памяти без изменений (дальнейшее уточнение в SW2)
	'65XX'	Состояние энергонезависимой памяти изменено (дальнейшее уточнение в SW2)
	'66XX'	Для сообщений, связанных с безопасностью
Ошибка контроля	'6700'	Неверно указанная длина (нет дальнейшего уточнения)
	'68XX'	Функции, указанные в байте CLA, не поддерживаются (дальнейшее уточнение в SW2)
	'69XX'	Команда не разрешена (дальнейшее уточнение в байте SW2)
	'6AXX'	Неверно указанный(е) параметр(ы) P1-P2 (дальнейшее уточнение в байте SW2)
	'6B00'	Неверно указанный(е) параметр(ы) P1-P2
	'6CXX'	Неверно указанное поле L_e: байт SW2 кодирует точное число доступных байтов данных (см. текст ниже)
	'6D00'	Командный код не поддерживается или недействителен
	'6E00'	Класс не поддерживается
	'6F00'	Нет точной диагностики

Если процесс прерван со значением SW1 от '64' до '6F', то поле данных ответа должно отсутствовать.

Если SW1 установлен в '63' или '65', то состояние энергонезависимой памяти будет изменено. Если SW1 установлен в '6X', за исключением '63' и '65', то состояние энергонезависимой памяти будет без изменений.

В ответе на команду, которая не является последней командой в цепочке (см. 5.1.1.1) межотраслевая индикация предупреждения не допускается (см. ИСО/МЭК 7816-3), т. е. SW1 не должен быть установлен ни на '62', ни на '63'.

Два межотраслевых значения SW1 зависят от протокола передачи:

- если SW1 установлен на '61', то процесс будет завершен; до выдачи какой-либо команды, команда GET RESPONSE может быть выдана с тем же CLA и с использованием SW2 (число еще доступных байтов данных) в качестве короткого поля L_e,

- если SW1 установлен на '6C', то процесс будет прерван; до выдачи какой-либо команды та же команда может быть повторно выдана с использованием SW2 (точное число доступных байтов данных) в качестве короткого поля L_e.

В таблице 6 перечислены специфические межотраслевые состояния предупреждения и состояния ошибок, используемые в серии ИСО/МЭК 7816.

Таблица 6 — Специфические межотраслевые состояние предупреждения и состояний ошибок

SW1	SW2	Смысловое содержание
'62' (предупреждение)	'00'	Информация не предоставлена
	От '02' до '80'	Активация карты (см. 8.6.1)
	'81'	Часть выдаваемых данных может быть искажена
	'82'	Конец файла или записи достигнут до считывания N_e байтов
	'83'	Выбранный файл недействителен
	'84'	Контрольная информация файла не форматирована по 5.3.3
	'85'	Выбранный файл в стадии завершении
	'86'	Нет входных данных на карту, полученных от датчиков
	'87'	Как минимум одна запись деактивирована
'63' (предупреждение)	'00'	Информация не предоставлена
	'81'	Файл заполнен при последней операции записи
	'CX'	Счетчик от 0 до 15, кодированный 'X' (точное смысловое содержание зависит от команды)
'64' (ошибка)	'00'	Ошибка выполнения
	'01'	Непосредственный ответ, требуемый картой
	От '02' до '80'	Активация карты (см. 8.6.1)
'65' (ошибка)	'00'	Ошибка выполнения
	'81'	Отказ памяти
'68' (ошибка)	'00'	Информация не предоставлена
	'81'	Логический канал не поддерживается
	'82'	Безопасный обмен сообщениями не поддерживается
	'83'	Ожидается последняя команда в цепочке
	'84'	Сцепление команд не поддерживается
'69' (ошибка)	'00'	Информация не предоставлена
	'81'	Команда несовместима со структурой файла
	'82'	Состояние защиты неудовлетворительное
	'83'	Метод аутентификации заблокирован
	'84'	Ссылочные данные не используются
	'85'	Условия использования не удовлетворены
	'86'	Команда невозможна (нет текущего EF)
	'87'	Пропадание ожидаемых информационных объектов, связанных с безопасным обменом сообщениями
	'88'	Информационные объекты, связанные с безопасным обменом сообщениями, некорректны

Окончание таблицы 6

SW1	SW2	Смысловое содержание
'69' (ошибка)	'00'	Информация не предоставлена
	'80'	Некорректные параметры в поле данных команды
	'81'	Функция не поддерживается
	'82'	Файл или приложение не найдены
	'83'	Запись не найдена
	'84'	Область памяти в файле недостаточна
	'85'	Nc не согласуется со структурой TLV
	'86'	Некорректные параметры P1-P2
	'87'	Nc не согласуется с параметрами P1-P2
	'88'	Ссылочные или ссылаемые данные не найдены (точное смысловое содержание зависит от команды)
	'89'	Файл уже существует
	'8A'	Имя DF уже существует
Примечание — Любое другое значение SW2 зарезервировано для использования в будущем ИСО/МЭК СТК 1/ПК 17.		

5.2 Информационные объекты

Каждое поле данных или сцепление полей данных, если оно(они) закодировано(ы) в TLY, представляет собой последовательность информационных объектов. В данном разделе определены две категории информационных объектов: информационные объекты SIMPLE-TLV и информационные объекты BER-TLV.

5.2.1Информационный объект SIMPLE-TLV

Каждый информационный объект SIMPLE-TLV должен состоять из двух или трех последовательных полей: обязательного поля тега, обязательного поля длины и условного поля значения. Записью (см. 7.3.1) может быть информационный объект SIMPLE-TLV.

Поле тега состоит из одиночного байта, кодирующего номер тега от 1 до 254. Значения '00' и 'FF' являются недействительными для поля тега. Если записью является информационный объект SIMPLE-TLV, то тег может использоваться как идентификатор записи.

Поле длины состоит из одного или трех последовательных байтов:

- если первый байт не установлен в 'FF', то поле длины состоит из одиночного байта, кодирующего число от нуля до 254 и обозначается как N;
- если первый байт установлен в 'FF', то поле длины продолжается последующими двумя байтами со значениями, кодирующими число от нуля до 65535 и обозначается как N.

Если N равно нулю, то поле значения отсутствует, т.е. информационный объект является пустым. В противном случае (N > 0), поле значения состоит из N последовательных байтов.

5.2.2 Информационный объект BER-TLV

Каждый информационный объект BER-TLV состоит из двух или трех последовательных полей (см. базовые правила кодирования АСН.1 по ИСО/МЭК 8825-1): обязательного поля тега, обязательного поля длины и условного поля значения:

Поле тега состоит из одного или большего числа последовательных байтов. Оно указывает класс и кодирование и кодирует номер тега. Значение '00' является недействительным для первого байта полей тега (см. ИСО/МЭК 8825-1).

Поле длины состоит из одного или большего числа последовательных байтов. Оно кодирует длину, т.е. число N.

Если N равно нулю, то поле значения отсутствует, т.е. информационный объект является пустым. В противном случае (N > 0), поле значения состоит из N последовательных байтов.

5.2.2.1 Поля тегов BER-TLV

Стандарты серии ИСО/МЭК 7816 поддерживают поля тегов из одного, двух или трех байтов; более длинные поля тегов зарезервированы для использования в будущем.

Биты 8 и 7 первого байта поля тега указывают класс:

- значение 00 указывает информационный объект универсального класса;
- значение 01 указывает информационный объект класса приложения;

- значение 10 указывает информационный объект контекстно-зависимого класса;
- значение 11 указывает информационный объект приватного класса.

Бит 6 первого байта поля тега указывает кодирование:

- значение 0 указывает простое кодирование информационного объекта, т. е. поле значения не кодировано в BER-TLV;
- значение 1 указывает составное кодирование информационного объекта, т. е. поле значения кодировано в BER-TLV.

Если биты с 5 по 1 первого байта поля тега не все установлены в состояние 1, то они должны кодировать номер тега от нуля до тридцати, при этом поле тега состоит из единственного байта.

В противном случае (биты с 5 по 1 установлены в состояние 1) поле тега должно продолжаться на один или большее число последующих байтов:

- бит 8 каждого последующего байта, за исключением последнего байта, должен быть установлен в состояние 1;
- биты с 7 по 1 первого последующего байта не должны быть все установлены в состояние 0;
- биты с 7 по 1 первого последующего байта, сцепленные с битами с 7 по 1 каждого из остальных последующих байтов, включая биты с 7 по 1 последнего байта, должны кодировать номер тега.

В таблице 7 показаны первые байты поля тега. Значение '00' является недействительным.

Таблица 7 — Первый байт BER-TLV полей тегов по ИСО/МЭК 7816.

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0	0	—	—	—	—	—	—	Универсальный класс, не определенный в ИСО/МЭК 7816
0	1	—	—	—	—	—	—	Класс приложения, идентификация определена в настоящем стандарте
1	0	—	—	—	—	—	—	Контекстно-зависимый класс, определен в ИСО/МЭК 7816
1	1	—	—	—	—	—	—	Приватный класс, не определен в ИСО/МЭК 7816
—	—	0	0	0	0	0	0	Простое кодирование
—	—	1	—	—	—	—	—	Составное кодирование
—	—	—	Не все байты установлены в состояние 1					Номер тега от нуля до тридцати (короткое поле тега, т. е. единичный байт)
—	—	—	1	1	1	1	1	Номер тега больше тридцати (длинное поле тега, т. е. два или три байта)

В полях данных, кодированных в BER-TLV, байты, установленные в '00', могут присутствовать до, между или после информационных объектов (например, за счет удаления или изменения информационных объектов в пределах EF, поддерживающего единицу данных). Такое заполнение запрещено в пределах полей значения составных информационных объектах и называется в ИСО/МЭК 7816 «шаблонами».

Если байт кодирования данных присутствует в байтах предыстории (см. 8.1.1) или в EF.ATR (см. 8.2.1.1) или в контрольной информации какого-либо файла (см. тег '82 ' в таблице 12), то этот байт (см. таблицу 87) указывает, является ли значение 'FF':

- действительным для первого байта длинного поля тега приватного класса, составного кодирования (заданного в явном виде), или
- недействительным для первого байта полей тегов (значение по умолчанию), т.е. используется для аналогичного назначения (заполнения) и при тех же условиях, что и значение '00'.

В полях тега двух или более байтах значения от '00' до '1E' и '80' являются недействительными для второго байта.

- В двухбайтных полях тега второй байт состоит из бита 8, установленного в состояние 0, и битов с 7 по 1, кодирующих число, большее тридцати. Второй байт имеет значения от '1F' до '7F'; номер тега — от 31 до 127.
- В трехбайтовых полях тега второй байт состоит из бита 8, установленного в состояние 1, и битов с 7 по 1, не все их которых установлены в состояние 0; третий байт состоит из бита 8, установленного в состояние 0, и битов с 7 по 1, принимающих любые значения. Второй байт принимает значения от '81' до 'FF', а третий байт — от '00' до '7F'; номер тега принимает значения от 128 до 16383.

5.2.2.2 Поля длины BER-TLV

В коротком формате поле длины состоит из единичного байта, в котором бит 8 установлен в состояние 0, а биты с 7 по 1 кодируют число байтов в поле значения. Таким образом, одним байтом может быть закодировано любое число от нуля до 127.

Примечание — Любое число от одного до 127 кодируется в поле длины BER-TLV так же, как в полях L_c и L_e. Кодирование отличается для ноля, 128 и более. Для примера см. кодирование информационных объектов в команде GET DATA в 7.4.2

В длинном формате поле длины состоит из двух или более байтов. Бит 8 первого байта установлен в состояние 1, а биты с 7 по 1 не должны быть все равны, таким образом происходит кодирование последующих байтов в поле длины. Эти последующие байты должны кодировать число байтов в поле значения.

В стандартах серии ИСО/МЭК 7816 не используют «неопределенную длину», указанную базовыми правилами кодирования АСН.1.

В стандартах серии ИСО/МЭК 7816 поддерживаются поля длины из одного, двух, ... до пяти байтов (см. таблицу 8). В ИСО/МЭК 7816 значения '80' и от '85' до 'FF' являются недействительными для первого байта полей длины.

Таблица 8 — Поля длины BER-TLV по ИСО/МЭК 7816

	1[A] байт	2[A] байт	3[A] байт	4[A] байт	5[A] байт	N
1 байт	От '00' до '7F'	—	—	—	—	От 0 до 127
2 байта	'81'	От '00' до 'FF'	—	—	—	От 0 до 255
3 байта	'82'	От '0000' до 'FFFF'		—	—	От 0 до 65535
4 байта	'83'	От '000000' до 'FFFFFF'			—	От 0 до 16777215
5 байтов	'84'	От '00000000' до 'FFFFFFFF'				От 0 до 4294967295

5.2.3 Поля данных, поля значений, информационные объекты и элементы данных

Каждое поле данных команды или ответа может быть закодировано в BER-TLV, например в паре команда-ответ, где CLA указывает безопасный обмен сообщениями (см. 6) или где бит 1 в INS установлен в состояние 1 (нечетный код INS, см. 5.1.2).

- Любой информационный объект BER-TLV обозначается {T-L-V} с полем тега, за которым следует поле длины, кодирующее число. В зависимости от того является ли число нулевым или нет, поле значения отсутствует (пустой информационный объект) или присутствует.

- Любой составной информационный объект обозначается {T-L-{T1-L1-V1}... -{Tn-Ln-Vn}} с полем тега, за которым следует поле длины, кодирующее число. Если число не является нулевым, то поле значения составного информационного объекта, т. е. шаблона, состоит из одного или более информационных объектов BER-TLV, каждый их которых состоит из поля тега, поля длины, кодирующего число, и если это число является нулевым, то из поля значения.

Некоторые поля данных, например команды для работы с единицами данных (см. 7.2), поля значений информационный объектов SIMPLE-TLV и поля значений некоторых простых информационных объектов состоят из элементов данных в соответствии с характеристиками команд или в соответствии с тегом информационного объекта.

Некоторые поля данных, например команды для работы с записями (см. 7.3) и поля значений некоторых простых информационных объектов BER-TLV состоят из информационных объектов SIMPLE-TLV.

Некоторые поля данных, например команды для работы с информационными объектами (см. 7.4) и поля значений составных информационных объектов BER-TLV, т. е. шаблонов, состоят из информационных объектов BER-TLV.

5.2.4 Идентификация элементов данных

Идентификация элементов данных основывается на следующих принципах:

1) Если число битов, представляющих элемент данных, не кратно восьми, то преобразование в байт или строку байтов должно быть указано в контексте соответствующего элемента данных. Если не указано иное, соответствующее число бит должно быть установлено на 1 в последнем байте, начиная с бита 1.

2) На стыке интерфейса между картой и устройством сопряжения элемент данных обычно присутствует в поле значения информационного объекта BER-TLV.

3) Для обеспечения поиска и обращения к элементу данных при обмене информацией, он должен быть связан с тегом информационного объекта BER-TLV и может быть сформирован в этом информационном объекте.

4) Обращение к элементу данных может осуществляться через непосредственно связанный с ним тег BER-TLV. Элемент данных может быть связан с другим элементом данных, который устанавливает контекст, к которому он принадлежит.

5) Один или несколько информационных объектов «команда на выполнение» могут непосредственно ссылаться на элемент данных.

6) При наличии, информационные объекты универсального класса (первый байт от '01' до '3F') принимают общее значение.

7) Все информационные объекты класса приложения (первый байт от '40' до '7F') являются межотраслевыми до тех пор, пока не будет указано иное. Настоящий стандарт и другие стандарты серии ИСО/МЭК 7816 выделяют теги для класса приложения. Каждый тег класса приложения, не определенный в серии ИСО/МЭК 7816 зарезервирован ИСО/МЭК СТК 1/ПК 17.

8) В настоящем стандарте определено множество межотраслевых элементов данных. В дополнение к определенным далее межотраслевым элементам данных, стандарты серии ИСО/МЭК 7816 содержат исчерпывающий перечень межотраслевых элементов данных, определенных в ИСО/МЭК 7816.

9) В карте может быть несколько одинаковых межотраслевых информационных объектов.

10) В поле данных команды и ответа все информационные объекты контекстно-зависимого класса (первый байт от '80' до 'BF') должны быть вложены в межотраслевые шаблоны, за исключением контрольной информации файла (см. 5.3.3) и безопасного обмена сообщениями (см. 6).

11) В последующих разделах (см. приложение А) определена схема распределения тегов для идентификации межотраслевых информационных объектов в полях данных. Эти схемы распределения тегов, используемые для идентификации межотраслевых информационных объектов, показаны в таблице 9 для уведомления органа, ответственного за распределение тегов.

Таблица 9 — Межотраслевые информационные объекты для органа распределения тегов

Тег	Смысловое значение
'06'	Идентификатор объекта (кодирование определено в ИСО/МЭК 8825-1, см. примеры в приложении А)
'41'	Код страны (кодирование определено в ИСО 3166 [1]) и, дополнительные национальные данные
'42'	Идентификационные номер эмитента (кодирование и регистрация определены в ИСО/МЭК 7812-1 [3]) и, дополнительные данные эмитента
'4F'	Идентификатор приложения (AID, кодирование определено в 8.2.1.2)

5.2.4.1 Совместимая схема распределения тегов

Эти схемы распределения тегов используют межотраслевые информационные объекты и дополнительные информационные объекты.

Эти дополнительные информационные объекты должны быть вложены в межотраслевые шаблоны, связанные с тегами от '70' до '77' (за исключением тега '73', зарезервированного для проприетарных информационых объектов, см. 5.2.4.3). Содержание тегов класса приложения в пределах этих шаблонов настоящий стандарт и стандарты серии ИСО/МЭК 7816 не определяют, за исключением тегов '41', '42' и '4F' для идентификации органа распределения тегов, как показано в таблице 9.

Использование контекстно-зависимого класса (первый байт от '80' до 'BF') в пределах межотраслевых шаблонов с тегами '65' (данные, относящиеся к держателю карты), '66' (данные карты), '67' (аутентификационные данные) и '6E' (данные, относящиеся к приложению) не рекомендуется.

Для идентификации совместимой схемы распределения тегов и органа, ответственного за эту схему, может быть использован межотраслевой шаблон, связанный с тегом '78'. При наличии, такой шаблон должен содержать один из межотраслевых информационных объектов, показанных в таблице 9, для идентификации органа распределения тегов.

- Если тег '78' имеется в строке начальных данных (см. 8.1.2) или в EF.ATR, то орган распределения тегов действителен для карты в целом.

- Если тег '78' присутствует в данных управления DF (см. 5.3.3), то орган распределения тегов является действительным в пределах этого DF.

5.2.4.2 Сосуществующая схема распределения тегов

В сосуществующих схемах распределения тегов могут использовать теги с интерпретацией, которая не определена в стандартах серии ИСО/МЭК 7816. Для идентификации сосуществующей схемы распределения тегов и органа, ответственного за эту схему, должен быть использован межотраслевой шаблон, связанный с тегом '79'. Если такой шаблон имеется, то он должен содержать один из межотраслевых объектов данных, показанных в таблице 9, для того, чтобы идентифицировать орган распределения тегов.

- Если орган распределения тегов действителен для карты в целом, тогда тег '79' должен присутствовать в строке начальных данных (см. 8.1.2) или в EF.ATR (см. 8.2.1.1).

- Если орган распределения тегов действителен в пределах DF, тогда тег '79' должен присутствовать в данных управления DF (5.3.3).

В такой схеме все межотраслевые информационные объекты должны быть вложены в межотраслевые шаблоны, связанные с тегом '7E'. Кроме того, теги '79' '7E' не должны интерпретироваться иначе, чем теги '62', '64', '6F' (шаблоны FCP, FMD и FCI, см. 5.3.3) и '7D' (шаблон SM, см. 6)

5.2.4.3 Независимые схемы распределения тегов

В этих схемах распределения тегов могут использовать теги с иной интерпретацией, чем принятая в настоящем стандарте и стандартах серии ИСО/МЭК 7816, которая не подчиняется требованиям 5.2.4.2. Такие схемы распределения тегов делают невозможным межотраслевой обмен информацией и не согласуются с настоящим стандартом.

Использование межотраслевых произвольных информационных объектов с тегами '53' для представления произвольных элементов данных и тегами '73' для вложения проприетарных информационных объектов в произвольные шаблоны дает возможность применения проприетарных элементов данных и информационных объектов, сохраняя в то же время соответствие настоящему стандарту.

5.3 Структуры приложений и данных

В настоящем подразделе определена структура приложений и данных, прослеживаемые на стыке интерфейса между картой и устройством сопряжения при обработке команд, используемых в межотраслевом классе. Размещение данных в физической памяти и структурная информация сверх той, что представлена в данном подразделе, находятся за пределами компетенции настоящего стандарта и стандартов серии ИСО/МЭК 7816.

Настоящий стандарт поддерживает следующие две категории структур: назначенный файл (DF) и элементарный файл (EF).

Файлы DF выполняют хостинг приложений и/или группы файлов и/или хранят информационные объекты. DF приложения является файлом DF, который выполняет хостинг приложения. DF может быть родительским файлом или иным файлом. Считается, что эти иные файлы находятся непосредственно под DF.

Файлы EF хранят данные. EF не может быть родительским или иным файлом. Определены следующие две категории файлов EF:

- внутренние EF хранят данные, интерпретируемые картой, т. е. данные, используемые картой в целях управления и контроля;

- рабочие EF хранят данные, не интерпретируемые картой, т. е. данные, подлежащие использованию исключительно внешними устройствами.

Предусмотрено два типа логической организации:

На рисунке 2 показана иерархия файлов DF с их соответствующей архитектурой безопасности (см. 5.4). В такой организации карт файл DF в основании иерархии называется главным файлом (MF); любой DF может быть файлом DF приложения с или без своей собственной иерархией файлов DF.

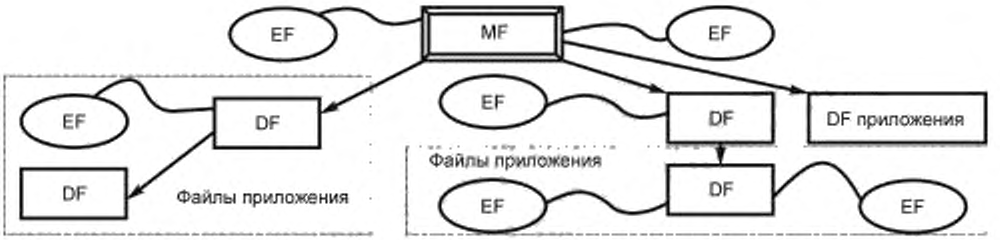


Рисунок 2 — Пример иерархии файлов DF

На рисунке 3 показаны файлы DF приложения с параллельной структурой без MF на стыке интерфейса, т.е. без видимой иерархии файлов DF. Такая организация поддерживает независимые приложения в карте, где любое DF приложения может иметь свою собственную иерархию файлов DF с соответствующей архитектурой безопасности.

Рисунок 3 — Пример независимых файлов DF приложения

5.3.1 Выбор структуры

5.3.1.1 Методы выбора структуры

Выбор структуры позволяет получить доступ к данным структуры и, если структура представляет собой DF, то и к ее подструктуре. Структура может быть выбрана неявно, т.е. автоматически после восстановления и возможного выбора протокола и параметров (см. ИСО/МЭК 7816-3). Если структура не может быть выбрана неявно, то она должна быть выбрана явно, т.е. с помощью, по меньшей мере, одного из следующих методов.

Выбор по имени DF — Обращение к любому DF может быть осуществлено по имени DF. Оно представляет собой строку до шестнадцати байтов. Любой идентификатор приложения (AID, см. 8.2.1.2) может быть использован в качестве имени DF. Для того чтобы однозначно выбирать по имени DF, например, когда выбор осуществляется путем использования идентификаторов приложений, имя каждого DF должно быть уникальным в данной карте.

Выбор посредством идентификатора файла — Обращение к любому файлу может быть осуществлено с помощью идентификатора файла. Он состоит из двух байт. Значение '3F00' зарезервировано для обращения к MF. Значение 'FFFF' зарезервировано для использования в будущем. Значение '3FFF' также зарезервировано (см. ниже и 7.4.1). Значение '0000' зарезервировано (см. 7.2.2 и 7.4.1). Для того чтобы однозначно выбирать любой файл с помощью его идентификатора, все файлы EF и DF, находящиеся в иерархии непосредственно под данным DF, должны иметь разные идентификаторы.

Выбор через путь — Обращение к любому файлу может быть осуществлено через путь. Он представляет собой сцепление идентификаторов файлов. Путь начинается с идентификатора DF (MF для абсолютного пути или текущего DF для относительного пути) и заканчивается идентификатором выбираемого файла. Между этими двумя идентификаторами путь состоит из идентификаторов последовательных (в рамках иерархии) родительских DF, если они имеются. Порядок следования идентификаторов файлов — всегда в направлении от родительского файла к дочернему. Если идентификатор текущего DF не известен, в начале пути может использоваться значение '3FFF' (зарезервированное значение). Значения '3F002F00' и '2F002F01' зарезервированы (см. 8.2.1.1). Использование пути позволяет осуществлять однозначный выбор любого файла из MF или из текущего DF (см. 8.3).

Обращение посредством короткого идентификатора EF — Обращение к любому EF может быть осуществлено с помощью короткого идентификатора EF. Он состоит из пяти бит, которые не все равны друг другу, т. е. представляют собой значение от одного до тридцати. Значение ноль, т. е. 00000, используемое в качестве короткого идентификатора EF, указывает на выбираемый в текущий момент EF. На уровне MF число тридцать, т. е. 11100 в бинарном выражении, зарезервировано (см. 8.2.1.1). Короткие идентификаторы файлов EF не могут быть использованы в последовательности пути или в качестве идентификатора файла EF (например, в команде SELECT).

Выбор посредством короткого идентификатора EF должен быть указан, если он поддерживается.

- Если первая таблица программных функций (см. таблицу 86) присутствует в байтах предыстории (см. 8.1.1) или в EF.ATR (см. 8.2.1.1), то индикация является действительной на уровне карты.

- Если короткий идентификатор EF (тег '88', см. таблицу 12) присутствует в контрольных параметрах (см. 5.3.3) файла EF, то индикация является действительной на уровне EF.

5.3.1.2 Элемент данных «ссылка на файл»

Связанный с тегом '51' данный межотраслевой элемент данных обращается к файлу. Он может иметь произвольную длину.

- Пустой информационный объект обращается к MF.

- Если длина равна единице и если биты с 8 по 4 элемента данных не все равны друг другу, и если биты с 3 по 1 установлены на 000, то биты с 8 по 4 кодируют число от одного до тридцати, которое является коротким идентификатором EF.

- Если длина равна двум, то элемент данных представляет собой идентификатор файла.

- Если длина больше двух, то элемент данных представляет собой путь:

- если длина четная и если первые два байта установлены на '3F00', то путь является абсолютным. Элемент данных представляет собой сцепление, как минимум, двух идентификаторов файла и идентификатора MF;

- если длина четная и если первые два байта не установлены на '3F00', то путь является относительным. Элемент данных представляет собой сцепление, как минимум, двух идентификаторов файла, начиная с идентификатора текущего DF;

- если длина нечетная, то путь специфицированный. Элемент данных является или абсолютным путем без '3F00', или относительным путем без идентификатора текущего DF, за которым следует байт, используемый в качестве P1 в одной или нескольких командах SELECT (см. 7.1.1 и 8.3).

В таблице 10 показан информационный объект «ссылка на файл».

Таблица 10 — Информационный объект «ссылка на файл»

Тег	Длина	Смысловое значение
'51'	0	Пустой информационный объект обращается к MF
	1	Короткий идентификатор EF (биты с 8 по 4 кодируют число от одного до тридцати; биты с 3 по 1 установлены на 000)
	2	Идентификатор файла
	Четная, >2	Абсолютный путь (первые два байта установлены на '3F00')
		Относительный путь (первые два байта не установлены на '3F00')
	Нечетная, >2	Специфицированный путь (последний байт используется в качестве P1 в одной или нескольких командах SELECT)

5.3.2 Методы обращения к данным

В файлах DF обращение к данным может осуществляться как к информационным объектам (см. 5.2). Файл DF прослеживается на стыке интерфейса между картой и устройством сопряжения как совокупность информационных объектов, имеющих доступ при помощи команд для работы с информационными объектами (см. 7.4).

В файлах EF обращение к данным может осуществляться как к единицам данных (см. 7.2.1), записям (см. 7.3.1) или информационным объектам (см. 5.2). Метод обращения к данным является характеристикой, зависящей от EF. Определены три структуры файлов EF:

- прозрачная структура, при которой файл EF прослеживается на стыке интерфейса между картой и устройством сопряжения как одиночная непрерывная последовательность единиц данных, имеющих доступ при помощи команд для работы с единицей данных (см. 7.2). Размер единицы данных является характеристикой, зависящей от EF;

- структура записи, при которой EF прослеживается на стыке интерфейса между картой и устройством сопряжения как единичная последовательность отдельно идентифицируемых записей, имеющих доступ при помощи команд для работы с записями (см. 7.3). Метод нумерации записей является характеристикой, зависящей от EF. Определены три атрибута:

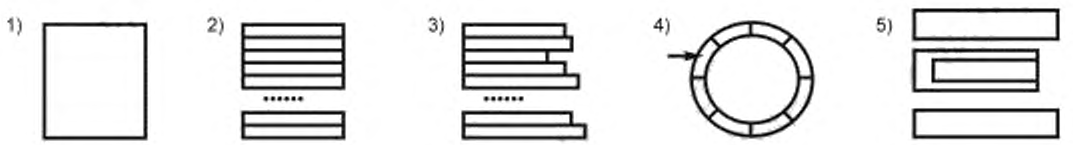
- размер записей — либо фиксированный, либо переменный;

- способ организации записей — либо в виде последовательного ряда (линейная структура), либо в виде кольца (циклическая структура);

- состояния жизненного цикла записи различаются, по крайней мере, следующими состояниями: ACTIVATED и DEACTIVATED. Кодирование состояния жизненного цикла записи выходит за рамки настоящего стандарта. В пределах некоторого EF либо все записи имеют состояние жизненного цикла записи, либо ни одна запись не имеет состояния жизненного цикла записи. Наличие состояния жизненного цикла записи указано в FCP (см. таблицу 12);

- структура TLV, при которой EF прослеживается на стыке интерфейса между картой и устройством сопряжения как совокупность информационных объектов, имеющих доступ при помощи команд для работы с информационными объектами (см. 7.4). Тип информационного объекта в EF, т. е. или SIMPLE-TLV или BER-TLV, является характеристикой, зависящей от EF.

Для обращения к данным в файлах EF карта поддерживает, по крайней мере, одну из пяти структур, показанных на рисунке 4.



1) прозрачная структура;
2) линейная структура с записями фиксированного размера,
3) линейная структура с записями переменного размера,
4) циклический EF с записями фиксированного размера (стрелка указывает на последнюю сделанную запись);
5) структура TLV

Рисунок 4 — Структуры EF

5.3.3 Контрольная информация файла

Контрольная информация файла представляет собой строку байтов данных, содержащуюся в ответе на команду SELECT (см. 7.1.1). Она может быть у любой структуры, т. е. у любого DF или EF.

Если первый байт имеет значение от '00' до 'BF', то строка байтов должна быть закодирована в BER-TLV. ИСО/МЭК СТК 1/ПК 17 зарезервировал для использования в будущем все значения в диапазоне от '00' до 'BF', которые не определены в настоящем стандарте.

Если первый байт имеет значения от 'C0' до 'FF', то строка байтов не кодируется в соответствии с настоящим стандартом.

В таблице 11 приведены три межотраслевых шаблона для вложенных информационных объектов BER-TLV контрольной информации файла.

- Шаблон FCP представляет собой совокупность контрольных параметров файла, т.е. логических, структурных атрибутов и атрибутов секретности, перечисленных в таблице 12 и определенных далее. В пределах шаблона FCP контекстно-зависимый класс (первый байт от '00' до 'BF') зарезервирован для контрольных параметров файла; теги '85' и 'A5' обращаются к произвольным данным.

- Шаблон FMD представляет собой совокупность данных управления файлом, т.е. межотраслевые информационные объекты, такие как идентификатор приложения по 8.2.1.2, уровень приложения по 8.2.1.4 и дата истечения срока действия по ИСО/МЭК 7816-6, возможно вложенные в пределах шаблона приложения по 8.2.1.3. В пределах шаблона FMD теги '53' и '73' обращаются к произвольным данным.

- Шаблон FCI представляет собой совокупность контрольных параметров файла и данных управления файлом.

Таблица 11 — Межотраслевые шаблоны для контрольной информации файла

Тег	Значение
'62'	Совокупность контрольных параметров файла (шаблон FCP)
'64'	Совокупность данных управления файлом (шаблон FMD)
'6F'	Совокупность контрольных параметров файла и контрольной информации файла (шаблон FCI)

Извлечение этих трех шаблонов может осуществляться по опциям выбора команды SELECT (см. таблицу 40).

Если установлена опция FCI, то тег FCI является необязательным для представления шаблона в поле данных ответа.

Если установлена опция FCP или FMD, то использование соответствующего тега является обязательным для представления шаблона.

В таблице 12 перечислены контрольные параметры файла (все в контекстно-зависимом классе). Если контрольный параметр имеется для файла, то в таблице указано возникает ли он только один раз (явная индикация) или он может повторяться (без индикации).

Таблица 12 — Информационные объекты «контрольные параметры файла»

Тег	L	Значение	Применимость
'8'	Переменная	Число байтов данных в файле, исключая структурную информацию	Любой EF, один раз
'81'	2	Число байтов данных в файле, включая структурную информацию, если она имеется	Любой файл, один раз
'82'	1	Байт описателя файла (см. 5.3.3.3 и таблицу 14)	Любой файл
	2	Байт описателя файла и байт кодирования данных (см. таблицу 87)	
	3 или 4	Байт описателя файла, байт кодирования данных и максимальная длина записи на одном или двух байтах	Любой EF, поддерживающий запись
	5 или 6	Байт описателя файла, байт кодирования данных, максимальная длина записи на одном или двух байтах и число записей на одном или двух байтах	
'83'	2	Идентификатор файла	Любой файл
'84'	От 1 до 16	Имя DF	Любой DF
'85'	Переменная	Проприетарная информация, не кодированная в BER-TLV	Любой файл
'86'	Переменная	Атрибуты секретности в проприетарном формате	Любой файл
'87'	2	Идентификатор файла EF, содержащего расширение контрольной информации файла	Любой DF, один раз
'88'	От 0 до 1	Короткий идентификатор EF (см. 5.3.3.1)	Любой EF, один раз
'8A'	1	Байт состояния жизненного цикла (байт LCS, см. 5.3.3.2 и таблицу 13)	Любой файл, один раз
'8B'	Переменная	Атрибут секретности, связывающий расширенный формат (см. 5.4.3.3 и таблицу 25)	Любой файл, один раз
'8C'	Переменная	Атрибут секретности в сжатом формате	Любой файл, один раз
'8D'	2	Идентификатор EF, содержащий шаблон безопасной среды (см. 6.3.4)	Любой DF

Окончание таблицы 12

Тег	L	Значение	Применимость
'8E'	1	Атрибут секретности канала (см. 5.4.3 и таблицу 13)	Любой файл, один раз
'8F'	1	Индикатор профиля (см. таблицу Изм. 1-1)	EF, поддерживающий запись, один раз
'A0'	Переменная	Шаблон «атрибут секретности» для информационных объектов (см. 5.4.3)	Любой файл, один раз
'A1'	Переменная	Шаблон «атрибут секретности» в проприетарном формате	Любой файл
'A2'	Переменная	Шаблон, состоящий из одной или нескольких пар информационных объектов: Короткий идентификатор EF (тег '88') — Ссылка на файл (тег '51', L > 2, см. 5.3.1.2)	Любой DF
'A5'	Переменная	Проприетарная информация, кодированная в BER-TL V	Любой файл
'AB'	Переменная	Шаблон «атрибут секретности» в расширенном формате (см. 5.4.3.2)	Любой файл, один раз
'AC'	Переменная	Шаблон «идентификатор криптографического алгоритма» (см. 5.4.2)	Любой DF
Примечание — В данном контексте ИСО/МЭК СТК 1/ПК 17 зарезервировал любые иные информационные объекты контекстно-зависимого класса (первый байт от '80' до 'BF')			

Часть контрольной информации файла DF может быть дополнительно представлена в файле EF под управлением приложения, ссылка на который приводится под тегом '87' в контрольных параметрах. В таком EF, если он имеется, контрольная информация файла должна быть представлена соответствующим тегом: или тегом FCP, или тегом FCI.

5.3.3.1 Короткий идентификатор EF

Для использования тега '88' в контрольных параметрах любого EF применяют следующие правила:

- если карта поддерживает выбор посредством коротких идентификаторов EF (см. 5.3.1.1) и если тег '88' отсутствует, то второй байт идентификатора файла (тег '83'), биты с 5 по 1 кодируют короткий идентификатор EF;

- если тег '88' присутствует с длиной, установленной на ноль, то EF не поддерживает короткий идентификатор;

- если тег '88' присутствует с длиной, установленной на единицу и если биты с 8 по 4 элемента данных не все равны друг другу, и если биты с 3 по 1 установлены на 000, то биты с 8 по 4 кодируют короткий идентификатор EF (т. е. число от одного до тридцати).

5.3.3.2 Байт состояния жизненного цикла

И карта, и файлы, и другие объекты имеют жизненный цикл; состояние жизненного цикла позволяет карте и устройству сопряжения идентифицировать различные логические состояния защиты при использовании карты, файлов или иных объектов в карте. Для поддержки гибкости управления жизненным циклом как атрибутом (см. ИСО/МЭК 7816-9 [4]) в настоящем разделе определены четыре основные состояния жизненного цикла в следующем порядке:

1) состояние создания;

2) состояние инициализации;

3) рабочее состояние;

4) состояние завершения.

Байт состояния жизненного цикла (байт LCS) должен быть интерпретирован в соответствии с таблицей 13:

- значения от '00' до '0F' являются межотраслевыми;

- значения от '10' до 'FF' являются проприетарными.

Таблица 13 — Байт состояний жизненного цикла

b8	b7	b6	b5	b4	b3	b2	b1	Значение
0	0	0	0	0	0	0	0	Информация не предоставлена
0	0	0	0	0	0	0	1	Состояние создания
0	0	0	0	0	0	1	1	Состояние инициализации
0	0	0	0	0	1	—	1	Рабочее состояние (активное)
0	0	0	0	0	1	—	0	Рабочее состояние (дезактивированное)
0	0	0	0	1	1	—	—	Состояние завершения
Не все равны нулю				x	x	x	x	Проприетарное
Примечание — Любые остальные значения зарезервированы для использования в будущем ИСО/МЭК СТК 1/ПК 17.								

Файл байта LCS, связанный с тегом '8A', может присутствовать в контрольных параметрах любого файла (см. таблицу 12).

Байт LCS карты может присутствовать в байтах предыстории (см. 8.1.1.3). Байт LCS карты, связанный с тегом '48', может присутствовать в EF.ATR (см. 8.2.1.1). Если он имеет MF, то карта находится, по крайней мере, в состоянии создания.

Примечание — Пока не указано иное, атрибуты секретности являются действительными для рабочего состояния.

5.3.3.3 Байт описателя файла

Элемент данных, связанный с тегом '82', может присутствовать в контрольных параметрах любого файла (см. таблицу 12).

Первым байтом элемента данных является байт описателя файла (см. таблицу 14).

Если элемент данных состоит из двух или более байтов, то вторым байтом является байт кодирования данных (см. таблицу 87). Если карта создает байты кодирования данных в нескольких местах, то индикация является действительной для заданного файла в ближайшей позиции, так что файл находится в пределах пути от MF до этого файла.

Таблица 14 — Байт описателя файла

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0	x	—	—	—	—	—	—	**Доступность файла**:
0	0	—	—	—	—	—	—	- Файл несовместного доступа
0	1	—	—	—	—	—	—	- Файл совместного доступа
0	—	1	1	1	0	0	0	**DF**:
0	—	Не все установлены на 1			—	—	—	**Категория EF**:
0	—	0	0	0	—	—	—	- Рабочий EF
0	—	0	0	0	—	—	—	- Внутренний EF
0	—	Любое иное значение			—	—	—	- Проприетарные категории файлов EF
0	—							**Структура EF**:
0	—	Не все установлены на 1			0	0	0	- Информация не предоставлена
0	—	Не все установлены на 1			0	0	1	- Прозрачная структура
0	—	Не все установлены на 1			0	1	0	- Линейная структура, фиксированный размер; нет дополнительной информации
0	—	Не все установлены на 1			0	1	1	- Линейная структура, фиксированный размер; структура TLV
0	—	Не все установлены на 1			1	0	0	- Линейная структура, переменный размер; нет дополнительной информации

Окончание таблицы 14

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0	—	Не все установлены на 1			1	0	1	- Линейная структура, переменный размер; структура TLV
0	—	Не все установлены на 1			1	1	0	- Циклическая структура, фиксированный размер, нет дополнительной информации
0	—	Не все установлены на 1			1	1	1	- Циклическая структура, фиксированный размер, структура TLV
0	—	1	1	1	0	0	1	- Структура TLV для информационных объектов BER-TLV
0	—	1	1	1	0	1	0	- Структура TLV для информационных объектов SIMPLE-TLV

Примечания

1 Любые остальные значения зарезервированы для использования в будущем ИСО/МЭК СТК 1/ПК 17.

2 «Совместного доступа» означает, что файл поддерживает, по меньшей мере, одновременный доступ по разным логическим каналам.

5.3.3.4 Индикатор профиля

При использовании тега '8F' в контрольных параметрах файла любого EF, поддерживающего запись, применяют следующие правила:

Если информационный объект с тегом '8F':

a) присутствует, то поле значения содержит индикатор профиля в соответствии с таблицей Изм.1-1;

b) отсутствует, то все записи в EF неявны в неизменном состоянии ACTIVATED.

Таблица Изм. 1-1 — Кодирование индикатора профиля

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0	x	x	x	x	x	x	x	Кодирование индикатора профиля определено ИСО/МЭК СТК 1/ПК 17
0	—	—	—	—	—	—	x	Состояние жизненного цикла записи
0	—	—	—	—	—	—	0	Все записи в данном файле находятся в неизменном состоянии ACTIVARED
0	—	—	—	—	—	—	1	Все записи в данном EF имеют свое состояние жизненного цикла записи, которое изменяется командами ACTIVARE RECORD, DEACTIVATE RECORD и ACTIVATE FILE
1	x	x	x	x	x	x	x	Проприетарное кодирование индикатора профиля

Примечание — Любые остальные значения зарезервированы для использования в будущем ИСО/МЭК СТК 1/ПК 17.

5.4 Архитектура безопасности

5.4.1 Общие положения

В настоящем подразделе рассмотрены состояние защиты, атрибуты секретности и механизмы защиты.

Состояние защиты. Состояние защиты представляет собой текущее состояние, которое может достигаться после завершения ответа на восстановление и возможного выбора протокола и параметров и/или отдельной команды или последовательности команд, возможно выполняющих процедуры аутентификации. Состояние защиты может являться также результатом завершения процедуры защиты, связанной с идентификацией участвующих сторон, если таковая применяется, например посредством проверки знания пароля (например, с использованием команды VERIFY), проверки знания ключа (например, с использованием команды GET CHALLANGE, сопровождаемой командой EXTERNAL AUTHENTICATE или последовательностью команд GENERAL AUTHENTICATE), или безопасного обмена сообщениями (например, на основе аутентификации сообщений). Рассматриваются следующие три состояния защиты.

- Глобальное состояние защиты. В карте, использующей иерархию файлов DF, глобальное состояние защиты может видоизменяться в результате завершения процедуры аутентификации, относящейся к MF (например, аутентификации участвующей стороны по паролю или ключу, присоединенному к MF).

- Состояние защиты, связанное с (конкретным) приложением. Оно может видоизменяться в результате завершения процедуры аутентификации, относящейся к приложению (например, аутентификации участвующей стороны по паролю или ключу, присоединенному к приложению); оно может быть сохранено, восстановлено или утрачено при выборе приложения; данное изменение может относиться только к приложению, к которому относится процедура аутентификации. Если применяют логические каналы, то состояние защиты, связанное с приложением может зависеть от логического канала.

- Файловое состояние защиты (т. е. ориентированное на файл). Файловое состояние защиты может видоизменяться в результате завершения процедуры аутентификации, относящейся к DF (например, аутентификации участвующей стороны по паролю или ключу, присоединенному к конкретному DF); оно может быть сохранено, восстановлено или утрачено при выборе файла; данное изменение состояния защиты может быть уместно только для приложения, с которым связана процедура аутентификации. Если применяют логические каналы, то файловое состояние защиты может зависеть от логического канала.

- Командное состояние защиты (т. е. ориентированное на команду). Командное состояние защиты имеет место лишь во время выполнения команды, предусматривающей аутентификацию с использованием безопасного обмена сообщениями; такая команда может оставлять без изменений другое состояние защиты.

- Атрибуты секретности. Атрибуты секретности, если они имеются, определяют разрешенные действия и условия. Атрибуты секретности файла зависят от его категории (DF или EF) и возможных параметров в его контрольной информации и/или в контрольной информации его родительского(их) файла(ов). Атрибуты секретности могут сопутствовать командам, информационным объектам и таблицам и представлениям. В частности, атрибуты секретности могут:

- определять состояние защиты карты, чтобы быть в действующем состоянии перед осуществлением доступа к данным;
- ограничить доступ к данным для определенных функций (например, только для чтения), если карта имеет особое состояние;
- определить, какие защитные функции должны быть выполнены, чтобы получить определенное состояние защиты.

Механизмы защиты. Настоящий стандарт устанавливает следующие механизмы защиты:

- Аутентификация участвующей стороны по паролю. Карта сравнивает данные, полученные от внешнего устройства, с секретными внутренними данными. Этот механизм может использоваться для защиты прав пользователя;

- Аутентификация участвующей стороны по ключу. Участвующая сторона, подвергаемая аутентификации, должна доказать знание соответствующего секретного или приватного ключа в ходе процедуры аутентификации (например, с использованием команды GET CHALLENGE, сопровождаемой командой EXTERNAL AUTHENTICATE и последовательностью команд GENERAL AUTHENTICATE);

- Аутентификация данных. Используя внутренние данные (или секретный ключ, или открытый ключ), карта проверяет избыточные данные, полученные от внешних устройств. В свою очередь, используя секретные внутренние данные (или секретный ключ, или приватный ключ) карта вычисляет элемент данных (криптографическую контрольную сумму или электронную цифровую подпись) и вставляет его в данные, посылаемые внешнему устройству. Данный механизм может использоваться для защиты прав провайдера;

- Шифрование данных. Используя секретные внутренние данные (или секретный ключ или приватный ключ), карта осуществляет дешифрование криптограммы, полученной в поле данных. В свою очередь, используя внутренние данные (или секретный ключ или открытый ключ), карта вычисляет криптограмму и вставляет ее в поле данных, возможно вместе с другими данными. Данный механизм может использоваться для обеспечения услуги конфиденциальности, например, для управления ключами и условного доступа. В дополнение к механизму с криптограммой, конфиденциальность данных может достигаться за счет сокрытия данных. В этом случае карта вычисляет строку скрывающих байтов и прибавляет ее с помощью операции сложения «исключающее ИЛИ» к байтам данных, полученным от внешнего устройства или посылаемым внешнему устройству. Данный механизм может использоваться для защиты личных секретных данных и для снижения возможностей фильтрации сообщений.

Результат аутентификации может регистрироваться во внутреннем EF в соответствии с требованиями приложения.

5.4.2 Шаблон «идентификатор криптографического механизма»

Связанный с тегом 'AC' один или несколько шаблонов «идентификатор криптографического механизма» может присутствовать в контрольных параметрах любого DF (см. таблицу 12). Каждый шаблон явно указывает значение ссылки криптографического механизма в DF и его иерархию. Такой шаблон должен состоять из одного или нескольких информационных объектов:

- Первый информационный объект должен быть ссылкой криптографического механизма, тег '80' (см. таблицу 33);
- Второй информационный объект должен быть идентификатором объекта, тег '06 ' (по ИСО/МЭК 8825-1). Идентифицированный объект должен быть криптографическим механизмом, определенным или зарегистрированный в стандарте, например, в стандартах серии ИСО. Примерами криптографических механизмов являются алгоритмы шифрования (например, ИСО/МЭК 18033 [18]), коды аутентификации сообщений (например, ИСО/МЭК 9797 [7]), протоколы аутентификации (например, ИСО/МЭК 9798 [8]), цифровые подписи (например, ИСО/МЭК 9796 [6] или ИСО/МЭК 14888 [16]), зарегистрированные криптографические алгоритмы (например, ИСО/МЭК 9979 [9]) и т. д.
- Последующие информационные объекты (один или более), если они имеются, должны или идентифицировать механизм, тег '06', используемый в предыдущем механизме (т. е. режим операции, например, ИСО/МЭК 10116[11], или хэш-функция, например, ИСО/МЭК 10118[12]), или указывать параметры (тег зависит от предыдущего механизма).

Примеры — См. пояснения в приложении А.

{'AC' — '0B' — {'80' — '01' — '01'} — {'06' — '06' – '28818C710201'}}

Шаблон связывает локальную ссылку '01' первого алгоритма шифрования по ИСО/МЭК 18033-2[18].

{'AC' — '11' — {'80' — '01' — '02'} — {'06' — '05' — '28CC460502'} — {'06' — '05' — '28CF060303'}}

Первый идентификатор объекта относится ко второму механизму аутентификации по ИСО/МЭК 9798-5 [8]. Второй идентификатор объектаотносится к третьей выделенной хэш-функции по ИСО/МЭК 10118-3 [12]. Поэтому шаблон связывает локальную ссылку '02' с GQ2, используя SHA-1.

5.4.3 Атрибуты секретности

Связанные с тегами '86', '8B', '8C', '8E', 'A0', 'A1', 'AB' атрибуты секретности могут присутствовать в контрольных параметрах любого файла (см. таблицу 12). Любой объект в карте (например, команда, файл, информационный объект, таблица и представление) может быть связан с больше чем одним атрибутом секретности и/или со ссылкой, содержащейся в атрибуте секретности.

Шаблон «атрибуты секретности», связанный с тегом 'A0', для информационного объекта может присутствовать в контрольных параметрах любого файла. Такой шаблон является сцеплением информационного объекта «атрибуты секретности » (теги '86', '8B', '8C', '8E', 'A0', 'A1', 'AB') и информационного объекта «список тегов» (тег '5C', см. 8.5.1), указывающего соответствующий информационный объект в файле.

Атрибут секретности канала, связанный с тегом '8E' (не более одного), может присутствовать в контрольных параметрах любого файла (см. таблицу 12) и в любой соответствующей безопасной среде (SE, см. 6.3.3). Он может быть интерпретирован в соответствии с таблицей 15.

- «Нет совместного доступа» означает, что должно быть доступно не более одного логического канала. Физически технология может быть ограничена.
- «Снабженный средствами защиты» означает, что SM ключи (см. 6) должны быть доступными (например, созданный при предыдущей аутентификации).
- «Пользователь аутентифицирован» означает, что пользователь должен быть аутентифицирован (например, при успешной проверке паролем).

Таблица 15 — Атрибут секретности канала

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0	0	0	0	0	—	—	1	Нет совместного доступа
0	0	0	0	0	—	1	—	Снабженный средствами защиты
0	0	0	0	0	1	—	—	Пользователь аутентифицирован
Примечание — Любые остальные значения зарезервированы для использования в будущем ИСО/МЭК СТК 1/ПК 17.								

В среде SCQL (см. ИСО/МЭК 7816-7 [4], команды языка структурированных запросов для карты) атрибуты секретности могут быть определены в операциях SCQL, например в командах CREATE TABLE и CRETE VIEW. Если используются атрибуты секретности, установленные в настоящем разделе, то они должны быть переданы в информационный объект с тегами '8B', '8C' или 'AB' в параметры атрибутов секретности SCQL операций.

Форматы. В настоящем разделе определены два формата для связанных объектов и атрибутов секретности: сжатый формат, основанный на битовом отображении, и расширенный формат, который расширяет сжатый формат с помощью управления TLV.

5.4.3.1 Сжатый формат

В сжатом формате правило доступа состоит из байта режима доступа, за которым следует один или несколько байтов условий секретности. Управление доступом к объекту организуют путем сцепления правил доступа со связанным объектом. Если в поле значения информационного объекта с тегом '8C' (см. таблицу 12) присутствуют несколько правил доступа, то они представляют собой условие логического И.

Байты режима доступа. Каждый бит с 7 по 1 указывает либо на отсутствие байта условия секретности, если байт установлен на 0, либо на присутствие байта условия секретности при той же последовательности бит (биты с 7 по 1), если он установлен на 1. Если бит 8 установлен на 1, то биты с 7 по 4 могут быть использованы для дополнительных команд, например для команд, связанных с (конкретным) приложением.

В таблицах с 16 по 19 определены байты режима доступа для файлов DF, EF, информационных объектов и таблиц и представлений, соответственно.

Таблица 16 — Байт режима доступа для файлов DF

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0	—	—	—	—	—	—	—	Биты с 7 по 1 в соответствии с данной таблицей
1	—	—	—	—	—	—	—	Биты с 3 по 1 в соответствии с данной таблицей (биты с 7 по 4 проприетарные)
0	1	—	—	—	—	—	—	DELETE FILE (автоматически)
0	—	1	—	—	—	—	—	TERMINATE CARD USAGE (MF), TERMINATE DF
0	—	—	1	—	—	—	—	ACTIVATE FILE
0	—	—	—	1	—	—	—	DEACTIVATE FILE
—	—	—	—	—	1	—	—	CREATE FILE (создание DF)
—	—	—	—	—	—	1	—	CREATE FILE (создание EF)
—	—	—	—	—	—	—	1	DELETE FILE (потомок)

Таблица 17 — Байт режима доступа для файлов EF

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0	—	—	—	—	—	—	—	Биты с 7 по 1 в соответствии с данной таблицей
1	—	—	—	—	—	—	—	Биты с 3 по 1 в соответствии с данной таблицей (биты с 7 по 4 проприетарные)
0	1	—	—	—	—	—	—	DELETE FILE
0	—	1	—	—	—	—	—	TERMINATE EF
0	—	—	1	—	—	—	—	ACTIVATE FILE, ACTIVATE RECORD
0	—	—	—	1	—	—	—	DEACTIVATE FILE, DEACTIVATE RECORD
—	—	—	—	—	1	—	—	WRITE BINARY, WRITE RECORD, APPEND RECORD
—	—	—	—	—	—	1	—	UPDATE BINARY, UPDATE RECORD, ERASE BINARY, ERASE RECORD(S)
—	—	—	—	—	—	—	1	READ BINARY, READ RECORD(S), SEARCH BINARY, SEARCH RECORD

Таблица 18 — Байт режима доступа для информационных объектов

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0	—	—	—	—	—	—	—	Биты с 7 по 1 в соответствии с данной таблицей
1	—	—	—	—	—	—	—	Биты с 3 по 1 в соответствии с данной таблицей (биты с 7 по 4 проприетарные)
0	x	x	x	x	—	—	—	000 (любые остальные значения зарезервированы для использования в будущем)
—	—	—	—	—	1	—	—	MANAGE SECURITY ENVIRONMENT
—	—	—	—	—	—	1	—	PUT DATA
—	—	—	—	—	—	—	1	GET DATA

Таблица 19 — Байт режима доступа для таблиц и представлений

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0	—	—	—	—	—	—	—	Биты с 7 по 1 в соответствии с данной таблицей
1	—	—	—	—	—	—	—	Биты с 3 по 1 в соответствии с данной таблицей (биты с 7 по 4 проприетарные)
0	1	—	—	—	—	—	—	CREATE USER, DELETE USER
0	—	1	—	—	—	—	—	GRANT, REVOKE
0	—	—	1	—	—	—	—	CREATE TABLE, CREATE VIEW, CREATE DICTIONARY
0	—	—	—	1	—	—	—	DROP TABLE, DROP VIEW
—	—	—	—	—	1	—	—	INSERT
—	—	—	—	—	—	1	—	UPDATE, DELETE
—	—	—	—	—	—	—	1	FETCH

Байт условия секретности. Каждый байт условия секретности определяет, какие механизмы защиты необходимы для соответствия правилам доступа. В таблице 20 показан байт условия секретности.

Таблица 20 — Байт условия секретности

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0	0	0	0	0	0	0	0	Нет условия
1	1	1	1	1	1	1	1	Никогда
0	—	—	—	0	0	0	0	Нет ссылки на безопасную среду
0	—	—	—	Не все равны				Идентификатор безопасной среды (байт SEID, см. 6.3.4) от одного до четырнадцати
—	—	—	—	1	1	1	1	Зарезервировано для использования в будущем
0	—	—	—	—	—	—	—	Не менее одного условия
1	—	—	—	—	—	—	—	Все условия
—	1	—	—	—	—	—	—	Безопасный обмен сообщениями
—	—	1	—	—	—	—	—	Внешняя аутентификация
—	—	—	1	—	—	—	—	Аутентификация пользователя (например, при помощи пароля)

Биты с 8 по 5 указывают требуемые условия секретности. Если не все равны, то биты с 4 по 1 идентифицируют безопасную среду (см. 6.3.4, байт SEID от одного до четырнадцати), а механизмы, определенные в безопасной среде, должны быть использованы в соответствии с указаниями в битах с 7 по 5 для команд по обеспечению безопасности и/или внешней аутентификации, и/или аутентификации пользователя.

- Если бит 8 установлен на 1, то все условия, установленные в битах с 7 по 5, должны быть выполнены.

- Если бит 8 установлен на 0, то должно быть выполнено, как минимум, одно условие, установленное в битах с 7 по 5.

- Если бит 7 установлен на 1, то шаблон управляющих ссылок (см. 6.3.1) безопасной среды, идентифицированный в битах с 4 по 1, т. е. байте SEID от одного до четырнадцати, описывает должен ли применяться безопасный обмен сообщениями для поля данных команды и/или поля данных ответа (см. использование байта квалификатора, см. таблицу 35).

5.4.3.2 Расширенный формат

В расширенном формате правило доступа состоит из информационного объекта «режима доступа», за которым следует один или несколько информационных объектов «условие секретности». Управление доступом к объекту организуют путем обращения соответствующего объекта к правилам доступа. Для таких правил шаблон с тегом 'AB' может присутствовать в контрольных параметрах любого файла (см. таблицу 12).

Информационный объект «режим доступа». Информационный объект «режим доступа» содержит или байт режима доступа (см. таблицы с 16 по 19), или список описаний команд, или проприетарное описание состояния машины; последующие информационные объекты «условие секретности» относятся ко всем указанным командам. В таблице 21 показан информационный объект «режим доступа».

Таблица 21 — Информационный объект «режим доступа»

Тег	Длина	Значение	Смысловое содержание
'80'	1	Байт режима доступа	См. таблицы с 16 по 19
От '81' до '8F'	Переменная	Описание заголовка команды	Список [части] заголовков команд (см. таблицу 22)
'9C'	Переменная		Проприетарное описание состояния машины

Если тег — от '81' до '8F', то элемент данных режима доступа представляет собой список возможных комбинаций значений четырех байтов: CLA, INS, P1 и P2 в заголовке команды. В зависимости от значения бит с 4 по 1 для тега список содержит только те значения, которые описаны в таблице 22. Для того чтобы определить набор команд, могут возникнуть несколько групп, например, значения INS P1 P2, INS P1 P2, ... для тега '87'.

Таблица 22 — Теги с '81' по '8F' для информационного объекта «режим доступа»

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
1	0	0	0	x	x	x	x	Описание команды включает:
1	0	0	0	1	—	—	—	- (CLA), т. е. значение CLA
1	0	0	0	—	1	—	—	- (INS), т. е. значение INS
1	0	0	0	—	—	1		- (P1), т. е. значение P1
1	0	0	0	—	—	—	1	- (P2), т. е. значение P2

Примечания

1 Значение CLA должно кодировать ноль как номер канала со значением, при котором описание не зависит от логических каналов.

2 Код INS должен быть четным и иметь значение, при котором описание не зависит от индикаций формата поля данных.

Информационный объект «условие секретности». В соответствии с таблицей 23 информационный объект «условие секретности» определяет действия по обеспечению безопасности, требуемые для получения доступа к объекту, защищенному определенным информационным объектом режима доступа. Шаблон управляющих ссылок (см. 6.3.1), связанный с тегом 'A4' (AT), 'B4' (CCT), 'B6' (DST) или 'B8' (CT) (при использовании в качестве условия секретности) должен содержать информационный объект «квалификатор применимости» (см. таблицу 35), указывающий действия по обеспечению безопасности.

Таблица 23 — Информационный объект «условие секретности»

Тег	Длина	Значение	Смысловое содержание
'90'	0	—	Всегда
'97'	0	—	Никогда
'9E'	1		См. таблицу 20
'A4'	Переменная	Байт условия секретности	Внешняя аутентификация или аутентификация пользователя в зависимости от квалификатора применимости
'B4', 'B6', 'B8'	Переменная	Шаблон управляющих ссылок	SM в команде и/или ответе в зависимости от квалификатора применимости
'A0'	Переменная	Шаблон управляющих ссылок	Как минимум одно условие секретности должно быть выполнено (шаблон OR)
'A7'	Переменная	Информационный объект «условие секретности»	Инверсия условия секретности (шаблон NOT)
'AF'	Переменная	Информационный объект «условие секретности»	Каждое условие секретности должно быть выполнено (шаблон AND)

К одной операции может быть прикреплено несколько информационных объектов «условие секретности»:

- если информационные объекты «условие секретности» вложены в шаблон OR (тег 'A0'), то, по крайней мере, одно условие секретности должно быть выполнено до начала работы;

- если информационные объекты условия секретности не вложены в шаблон OR (тег 'A0') или они вложены в шаблон AND (тег 'AF'), то каждое условия секретности должно быть выполнено до начала работы;

- если информационные объекты «условие секретности» вложены в шаблон NOT (тег 'A7'), то условия секретности являются истинными, пока они не выполнены.

5.4.3.3 Указатели правил доступа

Правила доступа расширенного формата могут быть сохранены в EF, поддерживающем линейную структуру с записями переменной длины. Такой EF называется EF.ARR. Одно или несколько правил доступа может храниться в каждой записи, на которую ссылается номер записи. Такой номер записи называется байтом ARR. В таблице 24 показана компоновка EF.ARR.

Таблица 24 — Компоновка EF.ARR

Число записи (байт ARR)	Содержание записи (одно или несколько правил доступа)
1	Информационный объект «режим доступа», один или несколько информационных объектов «условие секретности», информационный объект «режим доступа»
2	Информационный объект «режим доступа», один или несколько информационных объектов «условие секретности», ...

Связанные с тегом '8B' информационные объекты «атрибут секретности», ссылающиеся на расширенный формат (см. таблицу 25) могут присутствовать в контрольных параметрах любого файла (см. таблицу 12).

Если длина равна единице, то поле значения — байт ARR, который ссылается на запись в неявно известном EF.ARR.

Если длина равна трем, то поле значения — идентификатор файла, за которым следует байт ARR; идентификатор файла ссылается на EF.ARR, а байт ARR является числом записей в EF.ARR.

Если длина четная и не менее четырех, то поле значения — идентификатор файла, за которым следует одна или несколько пар байт. Каждая пара состоит из байта SEID, за которым следует байт ARR; байт SEID идентифицирует безопасную среду, в которой правила доступа ссылаются на применяемый байт ARR.

Таблица 25 — Информационные объекты «атрибут секретности», ссылающиеся на расширенный формат

Тег	Длина	Значение
'8B'	1	Байт ARR (один байт)
	3	Идентификатор файла (два байта) — байт ARR (один байт)
	Четная, >3	Идентификатор файла (два байта) — байт SEID (один байт) — байт ARR (один байт) — [байт SEID — байт ARR] — ...

Байт ARR текущей SE указывает правила доступа, действительные для текущего доступа к DF приложения.

Примечание — Если SE установлена в форме команды MANAGE SECURITY ENVIRONMENT, то SE по умолчанию будет текущая SE.

5.4.4 Элементы данных «обеспечение безопасности»

В настоящем разделе определено множество элементов данных «обеспечение безопасности» с правилами, определяющими метод обработки их данных. Элементы данных «обеспечение безопасности» расширяют и оптимизируют информационные объекты «управляющие ссылки». Карта может обеспечить базовую поддержку механизмов защиты, которые выполняет приложение. Приложения могут найти их по ссылке для безопасного обмена сообщениями и для операций по защите информации (см. ИСО/МЭК 7816-8 [4]). В настоящем разделе не определены некоторые характеристики элементов данных «обеспечение безопасности», например их длина, а также алгоритмы, которые изменяют их значение.

Принципы. Карта должна поддерживать и использовать значение элементов данных «обеспечение безопасности» следующим образом:

- обновление совершается с новыми значениями, либо вычисленными картой, либо предоставленными внешними устройствами, в соответствии со специальным правилом для специального типа элемента данных «обеспечение безопасности»;
- обновление выполняется до того, как какой-нибудь результат будет создан для команды, вызывающей обновление. Обновление не зависит от состояния завершения команды. Если значение должно использоваться приложением в операции, которая вызывает обновление, то обновление выполняется до того, как значение будет использовано;
- доступ к элементам данных «обеспечение безопасности», связанных с конкретным приложением, ограничен функциями, выполняемыми этим специфичным приложением.

Примечание — Реальная безопасность, достигаемая в паре команда-ответ, прежде всего, зависит от алгоритмов и протоколов, определяемых приложением; карта обеспечивает поддержку только с этими элементами данных и соответствующими правилами применения.

Элементы данных. Карта может поддерживать пару команда-ответ, защищенную элементами данных, называемыми значениями прогрессии. Определены два значения прогрессии — счетчик сессий карты и идентификатор сессий:

- счетчик сессий карты увеличивается один раз во время активации карты;
- идентификатор сессии вычисляется через счетчик сессий карты и из данных, предоставленных внешними устройствами.

Определены два типа значений прогрессии:

- внутренние значения прогрессии, если предусмотрены для приложения, регистрируют, сколько раз осуществляются специфичные события. Элемент данных должен увеличиться после события; карта может предоставить функцию восстановления для этих счетчиков, которые, если это предусмотрено приложением, устанавливают значение на ноль. Внутренние значения прогрессии не могут контролироваться внешними устройствами и подходят для использования как защищенное внутрикарточное приближенное представление в реальном времени. Эти значения могут использоваться в криптографических вычислениях;

- внешние значения прогрессии, если предусмотрены для приложения, должны обновляться только значением данных из внешних устройств. Новое значение должно быть численно больше, чем текущее значение, хранящееся в карте.

Ссылки. Карта может обеспечить доступ к значениям элементов данных «обеспечение безопасности» следующим образом:

- EF может присутствовать в MF, например для счетчика сессий карт, или в DF приложения, например, для значений прогрессии, связанных с конкретным приложением;

- вспомогательные информационные объекты (теги '88', '92', '93', см. таблицу 33) могут присутствовать в шаблоне управляющих ссылок. Эти теги могут использоваться, если SE поддерживает однозначное использование этих элементов данных;

- в межотраслевом шаблоне, связанном с тегом '7A', контекстно-зависимый класс (первый байт от '80' до 'BF') зарезервирован для информационного объекта «обеспечение безопасности», как перечислено в таблице 26.

Таблица 26 — Информационные объекты «обеспечение безопасности»

Тег	Значение
'7A'	Набор информационных объектов «обеспечение безопасности» со следующими тегами
'80'	Счетчик сессий карт
'81'	Идентификатор сессий карт
'82' '8E'	Счетчик выбора файла
'93'	Счетчик цифровой подписи
'9F2X'	Внутреннее значение прогрессии ('X' — специальный индекс, например индекс, ссылающийся на счетчик выбора файла)
'9F3Y'	Внешнее значение прогрессии ('X' — специальный индекс, например индекс, ссылающийся на внешнюю отметку времени)
Примечание — В данном контексте ИСО/МЭК СТК 1 ПК 17 зарезервировал любые другие информационные объекты контекстно-зависимого класса (первый байт от '80' до'BF').	

6 Безопасный обмен сообщениями

Безопасный обмен сообщениями (SM) защищает все (или части) пары команда-ответ, или сцепление последовательных полей данных (сцепление команд, см. 5.1.1.1; использование SW1, установленного на '61'), посредством обеспечения двух основных защитных функций: конфиденциальности данных и аутентификации данных. Безопасный обмен сообщениями достигается путем применения одного или большего числа механизмов защиты. Возможно идентифицированный в явном виде с помощью шаблона «идентификатор криптографического механизма» (см. 5.4.2) в параметрах управления каким-либо DF (см. 5.3.3), каждый механизм защиты включает в себя криптографический алгоритм, режим работы, ключ, аргумент (входные данные) и, зачастую, исходные данные.

- Передача и прием полей данных могут прерываться исполнением механизмов защиты. Настоящее описание не препятствует определению посредством последовательного анализа, какие механизмы и элементы защиты должны использоваться для обработки оставшейся части поля данных.

- Два или больше механизмов защиты могут использовать один и тот же криптографический алгоритм вместе с разными режимами работы. Представленные далее описания правил заполнения блоков данных незначащей информацией не препятствуют такой возможности.

6.1 Поля SM и информационные объекты SM

Любое поле данных команды или ответа в формате SM, а также шаблон SM (тег '7D') являются полем SM. Каждое поле SM должно кодироваться в BER-TLV (см. 5.2.2), где контекстно-зависимый класс (первый байт от '80' до 'BF') зарезервирован для информационного объекта SM. В паре команда-ответ формат SM может быть выбран или неявно, т. е. он известен до вызова команды, или явно, т. е. он указан CLA (см. 5.1.1).

Примечание — Сцепление команд и/или использование SW1, установленного на '61', вызывает последовательность команд, в которых поля данных (и, следовательно, информационные объекты) могут разбиваться на более маленькие последовательные поля данных. В таком случае, при использовании формата SM, сцепление всех последовательных полей данных в одном направлении в той же последовательности образует поле SM.

В таблице 27 показаны информационные объекты SM, определенные в настоящем стандарте, все в контекстно-зависимом классе. Некоторые информационные объекты SM (теги '82', '83', 'B0', 'B1' SM) являются рекурсивными, т. е. поле простого значения является полем SM.

Таблица 27 — Информационные объекты SM

Тег	Значение
'80', '81'	Простое значение, не закодированное в BER-TLV
'82', '83'	Криптограмма (простое значение, закодированное в BER-TLV и включающее информационные объекты SM, т.е. поле SM)
'84', '85'	Криптограмма (простое значение, закодированное в BER-TLV, но не включающее информационные объекты SM)
'86', '87'	Байт индикатора заполнения незначащей информацией, за которым следует криптограмма (простое значение не закодировано в BER-TLV)
'89'	Заголовок команды (CLA INS P1 P, четыре байта)
'8E'	Криптографическая контрольная сумма (не менее четырех байт)
'90', '91'	Хэш-код
'92', '93'	Сертификат (данные не кодированы в BER-TLV)
'94', '95'	Идентификатор безопасной среды (байт SEID)
'96', '97'	Один или два байта, кодирующие N_e в незащищенной паре команда-ответ (возможно пустой)
'99'	Состояние обработки (SW1-SW2, два байта; возможно пустые)
'9A', '9B'	Элемент данных ввода для вычисления цифровой подписи (поле значения подписано)
'9C', '9D'	Открытый ключ
'9E'	Цифровая подпись
'A0', 'A1'	Шаблон ввода для вычисления хэш-кода (шаблон хэширован)
'A2'	Шаблон ввода для верификации криптографической контрольной суммы (шаблон включен)
'A4', 'A5'	Шаблон управляющих ссылок аутентификации (AT)
'A6', 'A7'	Шаблон управляющих ссылок согласования ключей (KAT)
'A8'	Шаблон ввода для верификации цифровой подписи (шаблон подписан)
'AA', 'AB'	Шаблон управляющих ссылок хэш-кода (HT)
'AC', 'AD'	Шаблон ввода для вычисления цифровой подписи (сцепленные поля значений подписаны)
'AE', 'AF'	Шаблон ввода для верификации сертификата (сцепленные поля значений сертифицированы)
'B0', 'B1'	Простое значение закодировано в BER-TLV и включающее информационные объекты SM, т. е. поле SM
'B2', 'B3'	Простое значение закодировано в BER-TLV, но не включающее информационные объекты SM
'B4', 'B5'	Шаблон управляющих ссылок криптографической контрольной суммы (CCT)
'B6', 'B7'	Шаблон управляющих ссылок цифровой подписи (DST)
'B8', 'B9'	Шаблон управляющих ссылок конфиденциальности (CT)
'BA', 'BB'	Шаблон описателя ответа
'BC', 'BD'	Шаблон ввода для вычисления цифровой подписи (шаблон подписан)
'BE'	Шаблон ввода для верификации сертификата (шаблон сертифицирован)
Примечание — В данном контексте, ИСО/МЭК СТК1 ПК 17 зарезервировал любые другие информационные объекты контекстно-зависимого класса (первый байт от '80' до 'BF').	

В каждом поле SM бит 1 последнего байта поля тега (контроль по четности тега) каждого информационного объекта SM (контекстно-зависимый класс) указывает, должны ли информационные объекты SM быть включены (бит 1 установлен на 1, номер тега нечетный) или не включены (бит 1 установлен на 0, номер тега четный) при вычислении элементов данных для аутентификации (криптографическая контрольная сумма, см. 6.2.3.1, или цифровая подпись, см. 6.2.3.2). Информационные объекты других классов, если представлены, (например, межотраслевые информационные объекты) должны быть включены при вычислении. Если происходит такое вычисление, то элементом данных должно быть поле значения информационного объекта SM для аутентификации (тег SM '8E', '9E') в конце поля SM.

Существует две категории информационных объектов:

- каждый основной информационный объект SM (см. 6.2) передает либо простое значение, либо входные или результирующие данные о механизме защиты;
- каждый вспомогательный информационный объект SM (см. 6.3) передает шаблон управляющих ссылок, или идентификатор безопасной среды, или шаблон описателя ответа.

Примечание — Основные информационные объекты SM также используются для управления операций по защите информации (см. ИСО/МЭК 7816-8 [4]). Вспомогательные информационные объекты SM также используются для управления безопасной средой (см. 7.5.11). Общий подход к защите с использованием безопасного обмена сообщениями выделяет несколько вопросов, относящихся к обеспечению безопасности, по операциям по защите информации. В приложении В проиллюстрировано взаимодействие двух подходов.

6.2 Основные информационные объекты SM

6.2.1 Информационные объекты SM для инкапсуляции простых значений

Для полей SM и для данных, не закодированных в BER-TLV, инкапсуляция является обязательной. Она не является обязательной для информационных объектов BER-TLV, не включающих в себя SM. В таблице 28 показаны информационные объекты SM для инкапсуляции простых значений.

Таблица 28 — Информационные объекты SM для инкапсуляции простых значений

Тег	Значение
'B0', 'B1'	Простое значение, кодированное в BER-TLV и включающее информационные объекты SM (т. е. поле SM)
'B2', 'B3'	Простое значение, кодированное в BER-TLV, но не включающее информационные объекты SM
'80', '81'	Простое значение, не кодированное в BER-TLV
'89'	Заголовок команды (CLA INS P1 P, четыре байта)
'96', '97'	Один или два байта, кодирующие N_e в незащищенной паре команда-ответ (возможно пустой)
'99'	Состояние обработки (SW1-SW2, два байта; возможно пустые)

6.2.2 Информационные объекты SM для конфиденциальности

В таблице 29 показаны информационные объекты SM для конфиденциальности

Таблица 29 — Информационные объекты SM для конфиденциальности

Тег	Значение
'82', '83'	Криптограмма (простое значение, закодированное в BER-TLV и включающее информационные объекты SM, т.е. поле SM)
'84', '85'	Криптограмма (простое значение, закодированное в BER-TLV, но не включающее информационные объекты SM)
'86', '87'	Байт индикатора заполнения незначащей информацией, за которым следует криптограмма (простое значение не закодировано в BER-TLV)

Механизм защиты для конфиденциальности состоит из соответствующего криптографического алгоритма в соответствующем режиме операции. При отсутствии явного указания и выбранного неявно механизма для конфиденциальности должен применяться механизм, установленный по умолчанию:

- для вычисления криптограммы, которой должен предшествовать индикатор заполнения незначащей информацией, механизм по умолчанию представляет собой блочное шифрование в режиме «электронный кодовый справочник», который может включать в себя заполнение блоков данных незначащей информацией. Заполнение незначащей информацией для конфиденциальности оказывает влияние на передачу, криптограмма (один или большее число блоков) получается длиннее простого значения;
- для вычисления криптограммы, которой не должен предшествовать байт индикатора заполнения незначащей информацией, механизм по умолчанию представляет собой поточное шифрование. В этом случае криптограмма является результатом операции сложения «Исключающее ИЛИ», выполняемой над строкой байтов данных, подлежащих сокрытию, и скрывающей строкой той же длины. Сокрытие, таким образом, не требует заполнения незначащей информацией, а строкой байтов данных восстанавливается при помощи той же операции.

Если простое значение не закодировано в BER-TLV, заполнение незначащей информацией и/или информационное наполнение должно иметь индикацию. Если оно применяется, но индикации нет, то для него действуют правила, определяемые в 6.2.3.1. В таблице 30 показан байт индикатора заполнения незначащей информацией/с информационным наполнением.

Таблица 30 — Байт индикатора заполнения незначащей информацией/с информационным наполнением

Тег	Значение
'00' '01' '02'	Нет дальнейшей индикации Заполнение незначащей информацией по 6.2.3.1 Заполнение незначащей информацией отсутствует
'1X'	От одного до четырех секретных ключей для зашифрованной информации, не ключи ('X' — это битовое отображение со значениями от '0' до 'F') '11' указывает на первый ключ (например, «четное» контрольное слово в системе платного телевидения) '12' указывает на второй ключ (например, «нечетное» контрольное слово в системе платного телевидения) '13' указывает на первый ключ, за которым следует второй ключ (например, пара контрольных слов в системе платного телевидения)
'2X'	Секретный ключ для ключей шифрования, не информационный ('X' — это ссылка с любым значением от '0' до 'F') (например, в системе платного телевидения, или рабочий ключ для контрольных слов шифрования, или управление ключами для рабочих ключей шифрования)
'3X'	Приватный ключ для пары ассиметричных ключей ('X' — это ссылка с любым значением от '0' до 'F')
'4X'	Пароль (('X' — это ссылка с любым значением от '0' до 'F')
От '80' до '8E'	Проприетарное заполнение
Примечание — Любое другое значение зарезервировано для использования в будущем ИСО/МЭК СТК1 ПК 17.	

6.2.3 Информационные объекты SM для аутентификации

В таблице 31 показаны информационные объекты SM для аутентификации

Таблица 31 — Информационные объекты SM для аутентификации

Тег	Значение
'8E'	Криптографическая контрольная сумма (не менее четырех байт)
'90', '91'	Хэш-код
'92', '93'	Сертификат (данные не кодированы в BER-TLV)
'9C', '9D'	Открытый ключ
'9E'	Цифровая подпись
Информационные объекты ввода (см. также ИСО/МЭК 7816-8 [4])	
'9A', '9B'	Элемент данных ввода для вычисления цифровой подписи (поле значения подписано)
'A0', 'A1'	Шаблон ввода для вычисления хэш-кода (шаблон хэширован)
'A2'	Шаблон ввода для верификации криптографической контрольной суммы (шаблон включен)
'A8'	Шаблон ввода для верификации цифровой подписи (шаблон подписан)
'AC', 'AD'	Шаблон ввода для вычисления цифровой подписи (сцепленные поля значений подписаны)
'AE', 'AF'	Шаблон ввода для верификации сертификата (сцепленные поля значений сертифицированы)
'BC', 'BD'	Шаблон ввода для вычисления цифровой подписи (шаблон подписан)
'BE'	Шаблон ввода для верификации сертификата (шаблон сертифицирован)

6.2.3.1Элемент данных «криптографическая контрольная сумма»

Вычисление криптографической контрольной суммы предусматривает наличие исходного контрольного блока, секретного ключа и или алгоритма блочного шифрования (см. ИСО/МЭК 18033 [18]), или хэш-функции (см. ИСО/МЭК 10118 [12]).

Метод вычисления может быть частью системной спецификации. С другой стороны, шаблон «идентификатор криптографического механизма», см. 5.4.2, может идентифицировать стандарт (например, ИСО/МЭК 9797-1 [7]), устанавливающий метод вычисления.

Если не указано иное, должен использоваться следующий метод вычисления. Алгоритм под управлением связанного с ним ключа, по существу, преобразует блок текущего ввода из k байтов (обычно 8, 16 или 20) в блок текущего вывода той же длины. Вычисление выполняется последовательно по этапам.

Начальный этап. Начальный этап должен задать один из следующих блоков в качестве исходного контрольного блока:

- нулевой блок, т. е. k байтов, установленных на '00';
- связующий блок, т. е. результат предыдущих вычислений, а именно: для команды — результирующий контрольный блок предшествующей команды, для ответа — результирующий контрольный блок предшествующего ответа;
- блок с начальным значением, предоставленный, например, внешним устройством;
- вспомогательный блок, являющийся результатом преобразования вспомогательных данных под управлением ключа. Если вспомогательные данные составляют менее k байтов, то они озаглавливаются битами, установленными в ноль, до достижения длины блока.

Промежуточный этап(ы). Заголовок команды (CLA INS P1 P2) может быть инкапсулирован для защиты (тег SM '89'). Однако если биты с 8 по 6 в CLA установлены на 000, а биты с 4 по 3 на 11 (см. 5.1.1), то первый блок данных состоит из заголовка команды (CLA INS P1 P2), за которым следуют один байт, установленный на '80' и k—5 байт, установленных на '00'.

Криптографическая контрольная сумма должна включать любой информационный объект «безопасный обмен сообщениями», имеющий нечетный номер тега, и любой информационный объект с первым байтом от '80' до 'BF'. Эти информационные объекты должны быть включены поблочно в текущий контрольный блок. Разбивка на блоки данных должна осуществляться по следующим правилам:

- объединение в блоки должно быть непрерывным на границе между смежными информационными объектами, подлежащими включению;
- заполнение незначащей информацией должно применяться в конце каждого подлежащего включению информационного объекта, за которым либо следует информационный объект, не подлежащий включению, либо отсутствует дальнейший информационный объект. Незначащая информация состоит из одного обязательного байта, установленного на '80', за которым, если требуется, должны следовать от 0 до k—1 байтов, установленных на '00', пока соответствующий блок данных не будет заполнен k байтами. Заполнение незначащей информацией для аутентификации не оказывает влияния на передачу, поскольку заполняющие байты не должны передаваться.

В таком механизме, режимом работы является «последовательное блочное шифрование» (см. ИСО/МЭК 10116 [11]). Первый ввод представляет собой операцию сложения «Исключающее ИЛИ» над исходным контрольным блоком и первым блоком данных. Первый вывод является результатом первого ввода. Текущий ввод представляет собой операцию сложения «Исключающее ИЛИ» над предыдущим выводом и текущим блоком данных. Текущий вывод является результатом текущего ввода.

Заключительный этап. Результирующий контрольный блок является последним выводом. Заключительный этап выделяет криптографическую контрольную сумму (первые m байтов, но не менее четырех) из результирующего контрольного блока.

6.2.3.2 Элемент данных «цифровая подпись»

Схемы цифровой подписи основываются на асимметричных криптографических методах (см. ИСО/МЭК 9796 [6], ИСО/МЭК 14888 [16]). Вычисление предполагает использование хэш-функции (см. ИСО/МЭК 10118 [12]). Ввод данных состоит из поля значения информационного объекта ввода цифровой подписи, либо из сцепления полей значения информационных объектов, формирующих шаблон ввода цифровой подписи. Ввод данных может определяться согласно механизму, определяемому в 6.2.3.1.

6.3 Вспомогательные информационные объекты SM

В таблице 32 показаны вспомогательные информационные объекты SM.

Таблица 32 — Вспомогательные информационные объекты SM

Тег	Значение
'94', '95'	Идентификатор безопасной среды (байт SEID)
'A4 ', 'A5'	Шаблон управляющих ссылок, действительный для аутентификации (AT)
'A6', 'A7'	Шаблон управляющих ссылок, действительный для согласования ключей (KAT)
'AA', 'AB'	Шаблон управляющих ссылок, действительный для хэш-кода (HT)
'B4', 'B5'	Шаблон управляющих ссылок, действительный для криптографической контрольной суммы (CCT)
'B6', 'B7'	Шаблон управляющих ссылок, действительный для цифровой подписи (DST)
'B8', 'B9'	Шаблон управляющих ссылок, действительный для конфиденциальности (CT)
'BA', 'BB'	Шаблон описателя ответа

6.3.1 Шаблоны управляющих ссылок

Определены шесть шаблонов управляющих ссылок: действительный для аутентификации (AT), для согласования ключей (KAT), хэш-кода (HT), криптографической контрольной суммы (CCT), цифровой подписи (DST) и конфиденциальности (CT), использующие или симметричные, или ассиметричные криптографические методы (CT-sym и CT-asm).

Каждый механизм защиты включает в себя в своем режиме работы криптографический алгоритм и использует ключ и, возможно, исходные данные. Эти элементы выбирают либо неявно, т.е. они известны до выдачи команды, либо явно, т. е. с помощью информационного объекта «управляющие ссылки», вложенного в шаблоны управляющих ссылок. В рамках шаблонов управляющих ссылок контекстно-зависимый класс (первый байт от '80' до 'BF') зарезервирован для информационных объектов «управляющие ссылки».

В поле SM последняя возможная позиция шаблона управляющих ссылок — непосредственно перед первым информационным объектом, для которого применяется указываемый механизм. Например, последняя возможная позиция шаблона, действительного для криптографической контрольной суммы (CCT) располагается непосредственно перед первым информационным объектом для включения его при вычислении.

Каждая управляющая ссылка действует до тех пор, пока не будет предоставлена новая управляющая ссылка для того же механизма. Например, команда может предоставить управляющие ссылки для следующей команды.

6.3.2 Информационные объекты «управляющие ссылки» в шаблонах управляющих ссылок

Каждый шаблон управляющих ссылок (CRT) — это набор информационных объектов «управляющие ссылки»: ссылка на криптографический механизм, ссылка на файл и ключ, ссылка на исходные данные, квалификатор применимости и, только в шаблоне управляющих ссылок для конфиденциальности ссылка на содержание криптограммы.

- Ссылка на криптографический механизм обозначает криптографический алгоритм в режиме работы. Контрольные параметры любого DF (см. тег 'AC' в таблице 12) может содержать шаблон «идентификатор криптографического механизма» (см. 5.4.2). Каждый из них указывает на значение ссылки криптографического механизма.

- Ссылка на файл (тоже кодирование, что и в 5.3.1.2) обозначает файл, в котором ссылка на ключ действительна. Если ссылка на файл не представлена, то ссылка на ключ действительна в текущем DF, возможно, в DF приложения. Ссылка на ключ однозначно идентифицирует ключ, подлежащий использованию.

- Ссылка на исходные данные, при использовании для криптографических контрольных сумм, указывает исходный контрольный блок. Если ссылка на исходные данные не представлена и не выбран неявным образом исходный контрольный блок, то применяют нулевой блок.

Кроме того, перед передачей первого информационного объекта для конфиденциальности, использующего поточное шифрование, шаблон для конфиденциальности должен предоставить вспомогательные данные для инициализации вычисления строки скрывающих байтов.

В таблице 33 перечислены информационные объекты «управляющие ссылки» и указаны к каким шаблонам управляющих ссылок они относятся. Все информационные объекты «управляющие ссылки» находятся в контекстно-зависимом классе.

Таблица 33 — Информационные объекты «управляющие ссылки» в шаблонах управляющих ссылок

Тег	Значение	AT	KAT	HT	CCT	DST	CT-asym	CT-sym
'80'	**Ссылка на криптографический механизм**	x	x	x	x	x	x	x
Ссылки на файл и ключ								
'81'	- Ссылка на файл (то же кодирование, что и в 5.3.1.2)	x	x	x	x	x	x	x
'82'	- Имя DF (см. 5.3.1.1)	x	x	x	x	x	x	x
'83'	- Ссылка на секретный ключ (для прямого использования) - Ссылка на открытый ключ - Квалификатор эталонных данных	x x x	x x	x x	x	x	x	x
'84'	- Ссылка на вычисление сеансового ключа - Ссылка на приватный ключ	x x	x x		x	x	x	x
'A3'	- Шаблон применимости ключа (см. текст ниже)	x	x	x	x	x	x	x
Ссылка на исходные данные: исходный контрольный блок								
'85'	- L = 0, нулевой блок			x	x			x
'86'	- L = 0, связующий блок			x	x			x
'87'	- L = 0, предыдущий блок с начальным значением плюс один - L = k, блок с начальным значением			x	x x			x
Ссылка на исходные данные: Вспомогательные элементы данных (см. 5.4.3)								
'88'	- L = 0, данные предыдущей задачи, использованной в обмене, плюс один - L > 0, нет дальнейшей индикации				x	x	x	x
От '89' до '8D'	- L = 0, индекс проприетарного элемента данных - L > 0, значение проприетарного элемента данных				x x			x
'90'	- L = 0, хэш-код, предусмотренный картой			x		x		
'91'	- L > 0, случайное число, предусмотренное картой - L > 0, случайное число		x			x x	x x	
'92'	- L = 0, отметка времени, предусмотренная картой - L > 0, отметка времени			x		x x	x x	
'93'	- L = 0, счетчик предыдущей цифровой подписи плюс один - L > 0, счетчик цифровой подписи			x		x x	x x	x x
'94'	Задача или элемент данных для получения ключа	x			x			x
'95'	**Байт квалификатора применимости** (см. текст ниже)	x	x		x	x	x	x
'8E'	**Ссылка на содержимое криптограммы** (см. текст ниже)						x	x

Примечания

1 В данном контексте, ИСО/МЭК СТК1 ПК 17 зарезервировал любые другие информационные объекты контекстно-зависимого класса (первый байт от '80' до 'BF').

2 CRT может содержать межотраслевые информационные объекты, например, «авторизация держателя сертификата» (тег '5F4C', см. 6.3.4) в AT, список заголовков или расширенный список заголовков (теги '5D' и '4D') в HT или DST.

В любом шаблоне управляющих ссылок шаблон применимости ключа (тег 'A3') может связать ссылку на файл и ключ со счетчиком применимости ключа и/или счетчиком повтора ключа (см. таблицу 34).

Таблица 34 — Информационные объекты «применимость ключа»

Тег	Значение
'A3'	Набор информационных объектов «применимость ключа»
'80' '84' '90' '91'	Ссылки на файл и ключ, как определено в таблице 33 Счетчик применимости ключа Счетчик повтора ключа
Примечание — В данном контексте ИСО/МЭК СТК 1 ПК 17 зарезервировал любые другие информационные объекты контекстно-зависимого класса (первый байт от '80' до'BF').	

В любом шаблоне управляющих ссылок для аутентификации (АТ), для согласования ключей (КАТ), для криптографической контрольной суммы (ССТ), для конфиденциальности (СТ) или для цифровой подписи (DST) байт квалификатора применимости (тег '95') может определять использование шаблона либо в качестве условия секретности (см. 5.4.3.2 и таблицу 23), либо в соответствии с командой MANAGE SECURITY ENVIRONMENT (см. 7.5.11). В таблице 35 показан байт квалификатора применимости.

Таблица 35 — Байт квалификатора применимости

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
1	—	—	—	—	—	—	—	Верификация (DST, ССТ), Шифрование (СТ), Внешняя аутентификация (АТ), Согласование ключей (КАТ)
—	1	—	—	—	—	—	—	Вычисление (DST, ССТ), Расшифровывание (СТ), Внутренняя аутентификация (АТ), Согласование ключей (КАТ)
—	—	1	—	—	—	—	—	Безопасный обмен сообщениями в полях данных ответа (ССТ, СТ, DST)
—	—	—	1	—	—	—	—	Безопасный обмен сообщениями в полях данных команды (ССТ, СТ, DST)
—	—	—	—	1	—	—	—	Аутентификация пользователя, основанная на пароле (АТ)
—	—	—	—	—	1	—	—	Аутентификация пользователя, основанная на биометрии (АТ)
—	—	—	—	—	—	x	x	xxxx xx00 (любое другое значение зарезервировано для использования в будущем)

В любом шаблоне управляющих ссылок для конфиденциальности (СТ), ссылка на содержание криптограммы (тег '8E') может установить содержание криптограммы. Первый байт поля значения обязателен; его называют байтом описателя криптограммы. В таблице 36 показан байт описателя криптограммы.

Таблица 36 — Байт описателя криптограммы

Тег	Значение
'00'	Нет дальнейшей индикации
'1X'	От одного до четырех секретных ключей для зашифрованной информации, не ключи ('X' — это битовое отображение со значениями от '0' до 'F') '11' указывает на первый ключ (например, «четное» контрольное слово в системе платного телевидения) '12' указывае на второй ключ (например, «нечетное» контрольное слово в системе платного телевидения) '13' указывает на первый ключ, за которым следует второй ключ (например, пара контрольных слов в системе платного телевидения)

Таблица 36 — Байт описателя криптограммы

Тег	Значение
'2X'	Секретный ключ для ключей шифрования, не информационный ('X' — это ссылка с любым значением от '0' до 'F') (например, в системе платного телевидения, или рабочий ключ для контрольных слов шифрования, или управление ключами для рабочих ключей шифрования)
'3X'	Приватный ключ для пары ассиметричных ключей ('X' — это ссылка с любым значением от '0' до 'F')
'4X'	Пароль ('X' — это ссылка с любым значением от '0' до 'F')
От '80' до 'FF'	Проприетарное заполнение
Примечание — Любое другое значения зарезервировано для использования в будущем ИСО/МЭК СТК1 ПК 17.	

6.3.3 Безопасная среда

В настоящем разделе определена безопасная среда (SE) для обращения к криптографическим алгоритмам, режимам работы, протоколам, процедурам, ключам и дополнительным данным, необходимым для безопасного обмена сообщениями и для операций по защите информации (см. ИСО/МЭК 7816-8 [4]). SE состоит из элементов данных, сохраненных в карте или полученных в результате некоторого вычисления, которые должны обрабатываться специальными алгоритмами. SE может содержать механизм для инициализации непостоянных данных, которые должны использоваться в среде, например сеансовый ключ. SE может обеспечить управление обработкой результатов вычислений, например хранение в карте. Межотраслевой шаблон SE (тег '7B') описывает SE.

Идентификатор SE. Идентификатор SE (байт SEID) может ссылаться на любую безопасную среду, например для обращения к безопасному обмену сообщениями, а также для хранения и восстановления используют команды MANAGE SECURITY ENVIRONMENT (см. 7.5.11).

- Пока приложением не определено иначе, значение '00' обозначает, что среда пустая, т. е. безопасный обмен сообщениями и аутентификация не определены.
- Значение 'FF' обозначает, что никакая операция не может быть выполнена в данной среде.
- Пока приложением не определено иначе, значение '01' зарезервировано для SE, установленной по умолчанию, которая доступна всегда. В настоящем разделе не определено содержание SE, установленной по умолчанию; она может быть пустой.
- Значение 'EF' зарезервировано для использования в будущем.

Компоненты. Шаблоны управляющих ссылок (CRT) могут описывать различные компоненты SE. Любая относительная управляющая ссылка (файлы, ключи или данные), установленная с помощью механизма в определении среды, должна быть разрешена относительно DF, выбранного перед использованием механизма. В рамках SE компоненты могут иметь два аспекта: один действителен для SM в поле данных команды, а другой — для SM в поле данных ответа.

В течение всего времени операции карты текущая SE должна быть активна либо как установленная по умолчанию, либо как результат команд, выполненных картой. Текущая SE содержит один или несколько компонентов из числа следующих компонентов:

- некоторые компоненты являются частью SE, установленной по умолчанию и связанной с текущим DF;
- некоторые компоненты передаются в команды, используя безопасный обмен сообщениями;
- некоторые компоненты передаются в команды MANAGE SECURITY ENVIRONMENT;
- некоторые компоненты вызываются байтом SEID в команде MANAGE SECURITY ENVIRONMENT.

Текущая SE действительна до тех пор, пока происходит «горячее» восстановление или деактивация контактов (см. ИСО/МЭК 7816-3), изменение контекста (например, при выборе другого DF приложения) или установка команды MANAGE SECURITY ENVIRONMENT или замена текущей SE.

В SM информационные объекты «управляющая ссылка», передаваемые в CRT, должны иметь преимущественное право перед любым соответствующим информационным объектом «управляющая ссылка», присутствующим в текущей SE.

Авторизация держателя сертификата. Процедуры авторизации могут использовать сертификаты, верифицируемые картой, т. е. шаблоны, которые могут интерпретироваться и верифицироваться картой с помощью операции VERIFY CERTIFICATE, используя открытый ключ (см. ИСО/МЭК 7816-8 [4]). В таком сертификате авторизация держателя сертификата (например, идентификатор назначения) может быть передана в межотраслевой элемент данных, связанный с тегом '5F4C'. Если такой элемент данных используется в условиях секретности для того, чтобы выполнить доступ к данным или функциям, то информационный объект (тег '5F4C') должен присутствовать в шаблоне управляющих ссылок аутентификации (AT), описывающем процедуру аутентификации.

Примечание — В первом издании ИСО/МЭК 7816-9 [4], тег '5F4B' ссылается на авторизацию держателя сертификата (элемент данных из пяти или более байт). В изменении № 1 к первому изданию ИСО/МЭК 7816-9 тег '5F4B' ссылается на идентификатор изготовителя интегральной схемы (однобайтовый элемент данных). В результате тег '5F4B' исключен в серии ИСО/МЭК 7816.

Управление доступом. Карта может хранить безопасную среду, используемую для управления доступом в файлах EF (см. тег '8D' в таблице 12), содержащих межотраслевые шаблоны SE (тег '7B'). В рамках межотраслевых шаблонов SE (тег '7B') контекстно-зависимый класс (первый байт от '80' до 'BF') зарезервирован для информационных объектов «безопасная среда». Как перечислено в таблице 37, для каждой включенной SE шаблон «безопасная среда» содержит информационный объект «байт SEID» (тег '80'), дополнительный информационный объект «байт LCS» (тег '8A'), один или несколько дополнительных шаблонов «идентификатор криптографического механизма» (тег 'AC') и один или несколько CRT (теги 'A4', 'A6', 'AA', 'B4', 'B6', 'B8', как теги SM).

Таблица 37 — Информационные объекты «безопасная среда»

Тег	Значение
'7B'	Набор информационных объектов «безопасная среда» со следующими тегами
'80' '8A' 'AC' A4', 'A6', 'AA', 'B4', 'B6', 'B8'	Байт SEID, обязательный Байт LCS (см. 5.3.3.2 и таблица 13), дополнительный Шаблон «идентификатор криптографической контрольной суммы» (см. 5.4.2), дополнительный Шаблоны CRT (см. 6.3.1)
Примечание — В данном контексте ИСО/МЭК СТК 1 ПК 17 зарезервировал любые другие информационные объекты контекстно-зависимого класса (первый байт от '80' до'BF').	

Если информационный объект «байт LCS» присутствует в шаблоне SE, то он указывает какое состояние жизненного цикла SE действительно. Если SE используется для управления доступом, например, для управления доступом файла, то байт LCS файла и байт LCS для SE должны совпадать. Если информационный объект «байт LCS» не присутствует, то SE действительна для активированного рабочего состояния.

В шаблоне SE, если CRT содержит несколько информационных объектов с тем же тегом (например, информационные объекты, определяющие ссылку на ключ), то, как минимум, один из информационных объектов должен быть выполнен (условие ИЛИ).

Извлечение SE. Любой CRT в текущей SE может быть извлечен с помощью команды GET DATA с P1-P2, установленными на '004D' (расширенный список заголовка, см. 8.5.1), и поля данных команды, состоящего из шаблона SE (тег '7B'), который сцепляет одну или несколько пар, каждая их которых состоит из тега CRT, за которым следует '80' (см. 8.5.1 для использования длины, установленной на '80' в расширенном списке заголовка).

6.3.4 Шаблон описателя ответа

Каждое поле данных команды может содержать шаблон описателя ответа. Шаблон описателя ответа, если он представлен в поле данных команды, должен указывать информационные объекты SM, необходимые в поле данных ответа. Внутри шаблона описателя ответа механизмы защиты пока не применяют; получатель должен применить их для построения поля данных ответа. Элементы защиты (алгоритмы, режимы операций, ключи и исходные данные), используемые для обработки поля данных команды, могут отличаться от элементов защиты, используемых для выработки поля данных последующего ответного сообщения. Должны применяться следующие правила:

- карта должна заполнять каждый пустой простой информационный объект SM;
- каждый CRT, присутствующий в шаблоне описателе ответа, должен быть представлен в ответе на том же месте с тем же информационными объектами «управляющие ссылки» для механизмов защиты, файла и ключей:
  - если шаблон описателя ответа предоставляет вспомогательные данные, тогда соответствующий информационный объект в ответе должен быть пустым;
  - если в шаблоне описателя ответа присутствует пустой ссылочный информационный объект для вспомогательных данных, то в ответе он должен быть заполнен;
  - применяя соответствующие механизмы защиты вместе с выбранными элементами защиты, карта должна осуществлять вывод всех запрашиваемых основных информационных объектов SM.

6.4 Влияние SM на пары команда-ответ

На рисунке 5 показана пара команда-ответ.

Рисунок 5 — Пара команда-ответ

Для обеспечения защиты пары команда-ответ межотраслевого класса (см. 5.1.1) применяют следующие правила, т. е. при переключении либо бита 1 из состояния 0 в состояние 1 в CLA, где биты 8, 7 и 6 установлены на 000, либо бита 6 из состояния 0 в состояние 1 в CLA, где биты 8 и 7 установлены на 01. Запись CLA* означает, что безопасный обмен сообщениями показан в CLA.

- Защищенное поле данных команды — поле SM; оно должно формироваться следующим образом:
  - если поле данных команды присутствует ($N_c > 0$), то либо информационный объект «простое значение» (теги SM '80', '81', 'B2', 'B3'), либо информационный объект для конфиденциальности (теги SM '84', '85', '86', '87') должны передавать N_c байты;
  - заголовок команды (четыре байта) может быть инкапсулирован для защиты (тег SM '89');
  - если поле L_e присутствует, то новое поле L_e (содержащее только установленные на '00' байты) и информационный объект L_e (теги SM '96', '97') должны присутствовать. И ноль, и пустой информационный объект L_e означают максимум, т. е. 256 или 65536, в зависимости от того является ли новое поле L_e коротким или расширенным.
- Защищенное поле данных ответа — поле SM; оно должно формироваться следующим образом:
  - информационный объект «простое значение» (теги SM '80', '81', 'B2', 'B3'), если имеется, или информационный объект для конфиденциальности (теги SM '84', '85', '86', '87') передают байты данных ответа;
  - информационный объект «состояние обработки» (тег SM '99'), если имеется, передает SW1-SW2, инкапсулированные для защиты. Пустой информационный объект «состояние обработки» означает, что SW1-SW2 установлены на '9000'.

На рисунке 6 показана соответствующая защищенная пара команда-ответ.

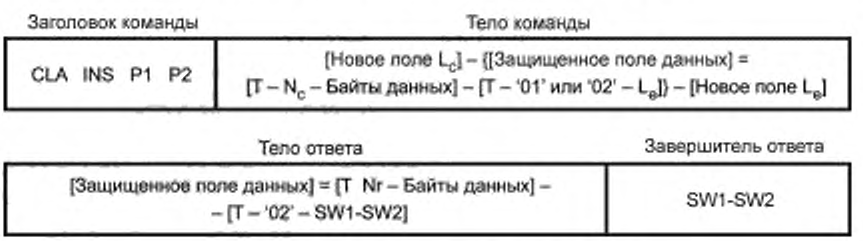


Рисунок 6 — Защищенная пара команда-ответ

Если бит 1 в INS установлен в состояние 1 (нечетный код INS, см. 5.1.2), то незащищенные поля данных кодируются в BER-TLV и теги SM 'B2', 'B3', '84' и '85' должны использоваться для их инкапсуляции. В противном случае, так как формат для полей данных для защиты не всегда видим, то рекомендованы теги SM '80', '81', '86' и '87'.

- Защищенные поля данных — поля SM; они могут содержать дополнительные или другие информационные объекты SM, например криптографическую контрольную сумму (тег SM '8E') или цифровую подпись (тег SM '9E') на конце.

- Новое поле L_c кодирует число байт в защищенном поле данных команды.

- Новое поле L_e должно быть пустым, если не ожидается поле данных в защищенном поле данных ответа; в противном случае оно должно содержать только установленные на '00' байты;

- Завершитель ответа указывает состояние принимающей стороны после обработки защищенной команды. Могут происходить следующие специфические состояния ошибок:

- если SW1-SW2 установлены на '6987', то ожидаемый информационный объект «безопасный обмен сообщениями» отсутствует;

- если SW1-SW2 установлены на '6988', то информационный объект «безопасный обмен сообщениями» с ошибкой.

В приложении В приведены проиллюстрированные примеры безопасного обмена сообщениями.

7 Команды для обмена

В настоящем разделе определены команды для обмена, представленные ниже:

1) выбор;
2) обработка блока данных;
3) обработка записи;
4) обработка информационного объекта;
5) обработка основной защиты;
6) обработка передачи.

Настоящий стандарт не устанавливает обязательное требование к соблюдению всех приведенных команд или всех опций поддерживаемых команд карт, соответствующих настоящему стандарту. Если необходим обмен информацией, то услуги карты, не зависимые от приложения, и соответствующие команды и опции должны использоваться, как определено в разделе 8.

7.1 Выбор

После ответа на восстановление MF или DF приложения неявно вызывают с помощью основного логического канала (см. 5.1.1.2), пока не будет определено иначе в байтах предыстории (см. 8.1.1) или в строке исходных данных (см. 8.1.2)

7.1.1 Команда SELECT

После своего завершения команда SELECT открывает логический канал (см. 5.1.1.2), пронумерованный в CLA (см. 5.1.1), если он до этого не был открыт, и устанавливает текущую структуру в рамках этого логического канала. Последующие команды могут неявно обращаться к текущей структуре через этот логический канал.

- Выбранный DF (MF или DF приложения) становится текущим в логическом канале. К предыдущему выбранному DF, если он имеется, больше не обращаются с помощью этого логического канала, и он становится предшествующим текущим DF.

- Выбор EF устанавливает пару текущих файлов: EF и его родительский DF.

Если не дается иных указаний, то к каждому логическому каналу в рамках иерархии файлов DF применяют следующие правила:

- если заменяется текущий EF, или текущего EF больше нет, состояние защиты, если оно применялось, ориентированное на предшествующий текущий EF, утрачивается;

- если текущий DF является потомком предшествующего текущего DF или ему идентичен, состояние защиты, ориентированное на предшествующий текущий DF, сохраняется;

- если текущий DF не является потомком предшествующего текущего DF и не идентичен ему, состояние защиты, ориентированное на предшествующий текущий DF, утрачивается. Состояние защиты, общее для всех общих предков предшествующего и нового текущего DF, сохраняется.

Пара команда-ответ SELECT представлена в таблице 38.

Таблица 38 — Пара команда-ответ SELECT

Тег	Значение
CLA INS P1 P2	Как определено в 5.1.1 'A4' См. таблицу 39 См. таблицу 40
Поле L_c	Отсутствует для кодирования $N_c = 0$, присутствует для кодирования $N_c > 0$
Поле данных	Отсутствует или идентификатор файла, или путь, или имя DF (в соответствии с P1)
Поле L_e	Отсутствует для кодирования $N_e = 0$, присутствует для кодирования $N_e > 0$

Поле данных	Отсутствует или контрольная информация файла (в соответствии с P2)
SW1-SW2	См. ИСО/МЭК 7816-4, таблицы 5 и 6, где соответствие, например '6283', '6284', '6A80', '6A81', '6A82', '6A86', '6A87'

Если P1 установлен на '00', то карта распознает, является ли выбираемый файл файлом MF, DF или EF, либо благодаря особому кодированию идентификатора файла, либо из-за контекста выполнения команды.

- Если P2 установлен на '00' и поле данных команды предоставляет идентификатор файла, то он должен быть уникальным в следующих трех средах: непосредственных потомках текущего DF, родительских DF, непосредственных потомках родительского DF.
- Если P2 установлен '00' и поле данных пустое или установлено на '3F00', то осуществляется выбор MF.

Если P1 установлен на '04', то поле данных представляет собой имя DF, возможно, укороченное справа, которым может быть идентификатор приложения (см. 8.2.1.2). Если выбор по укороченному имени DF поддерживается, то следующие одна за другой такие команды с одним и тем же полем данных должны выбирать файлы DF, чьи имена совпадают с их полем данных, т. е. с него начинаются. Если карта принимает команду SELECT с пустым полем данных, то может осуществляться последовательный выбор всех файлов DF или их подмножества.

Если поле L_e содержит только установленные на '00' байты, то все байты, соответствующие опциям выбора должны быть выданы в пределах максимума 256 для короткого поля L_e или 65536 для расширенного поля L_e. Если поле L_e отсутствует, т.е. нет контрольной информации файла для выдачи, то поле данных ответа должно также отсутствовать.

Таблица 39 — P1

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание	Поле данных команды
0	0	0	0	0	0	x	x	**Выбор посредством идентификатора файла**:	
0	0	0	0	0	0	0	0	Выбирать MF, DF или EF	Идентификатор файла или пустое
0	0	0	0	0	0	0	1	Выбирать дочерний DF	Идентификатор DF
0	0	0	0	0	0	1	0	Выбирать EF под текущим DF	Идентификатор EF
0	0	0	0	0	0	1	1	Выбирать родительский DF текущего DF	Пустое
0	0	0	0	0	1	x	x	**Выбор по имени DF**:	
0	0	0	0	0	1	0	0	Выбирать по имени	Например, [сокращенный] идентификатор приложения
0	0	0	0	1	0	x	x	**Выбор через путь**:	
0	0	0	0	1	0	0	0	Выбирать из MF	Путь без идентификатора MF
0	0	0	0	1	0	0	1	Выбирать из текущего DF	Путь без идентификатора текущего DF

Примечания

1 Любое другое значение зарезервировано для использования в будущем ИСО/МЭК СТК 1 ПК 17.

2 Первая таблица программных функций (см. таблицу 86), если она присутствует в байтах предыстории (см. 8.1.1) или в EF.ATR (см. 8.2.1.1), указывает методы выбора, поддерживаемые картой.

Таблица 40 — P2

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0	0	0	0	—	—	x	x	**Вхождение файла:**
0	0	0	0	—	—	0	0	- Первое или единственное вхождение;
0	0	0	0	—	—	0	1	- Последнее вхождение;
0	0	0	0	—	—	1	0	- Следующее вхождение;
0	0	0	0	—	—	1	1	- Предыдущее вхождение.
0	0	0	0	x	x	—	—	**Контрольная информация файла** (см. 5.3.3 и таблицу 11):
0	0	0	0	0	0	—	—	- Выдать FCI, необязательное использование тега и длины FCI;
0	0	0	0	0	1	—	—	- Выдать шаблон FCP, обязательное использование тега и длины FCP;
0	0	0	0	1	0	—	—	- Выдать шаблон FMD, обязательное использование тега и длины FMD;
0	0	0	0	1	1	—	—	- Нет данных ответа, если поле L_e отсутствует, или собственная информация, если поле L_e присутствует.
Примечание — Любое другое значение зарезервировано для использования в будущем ИСО/МЭК СТК 1 ПК 17.								

7.1.2 Команда MANAGE CHANNEL

После своего завершения команда MANAGE CHANNEL открывает и закрывает логический канал (см. 5.1.1.2), отличный от основного канала, т. е. канал нумеруется от одного до девятнадцати (большие номера зарезервированы для использования в будущем).

Функция открытия открывает новый логический канал, отличный от основного канала. Предусматриваются опции для назначения номера канала картой, либо для номера канала, который должен задаваться карте.

- Если бит 8 байта P1 установлен на 0 (т. е. P1 установлен на '00', так как остальные семь бит зарезервированы для использования в будущем), то команда MANAGE CHANNEL должна открыть канал с номером от одного до девятнадцати следующим образом:

- если P2 установлен на '00', то поле Le должно быть установлено на '01' и поле данных ответа должно состоять из одиночного байта для кодирования ненулевого канала, назначенного картой от '01' до '13';

- если P2 принимает значения от '01' до '13', то он кодирует назначенный извне номер ненулевого канала, а поле L_e должно отсутствовать.

- После того, как функция открытия приводится в действие с основного логического канала (кодирование нуля в CLA так же, как и номера канала).

- После того, как функция открытия приводится в действие с неосновного логического канала (кодирование ненулевого номера канала в CLA), текущий DF на канал, пронумерованный в CLA, должен стать текущим в новом канале.

Функция закрытия в явной форме закрывает логический канал, отличный от основного канала. Поле Le должно отсутствовать. После закрытия логический канал должен быть доступен для повторного использования.

- Если бит 8 байта P1 установлен на 1 (т. е. P1 установлен на '80', так как остальные семь бит зарезервированы для использования в будущем), то команда MANAGE CHANNEL должна закрыть канал с номером от 1 до девятнадцати следующим образом:

- если P2 установлен на '00', то канал, пронумерованный в CLA (ненулевой номер канала) должен быть закрыт;

- если P2 имеет значения от '01' до '13', то канал, пронумерованный в P2, должен быть закрыт.

Примечание — Функция закрытия может быть прервана, если CLA не указывает ни основной канал, ни канал, пронумерованный в P2.

Пара команда-ответ MANAGE CHANNEL представлена в таблице 41.

Таблица 41 — Пара команда-ответ MANAGE CHANNEL

CLA INS P1-P2	Как определено в 5.1.1 '70' '0000' для открытия логического канала, который должен быть пронумерован в поле данных ответа От '0001' до '0013' для открытия логического канала, пронумерованного в P2 '8000' для закрытия логического канала, пронумерованного в CLA (отличного от основного канала) От '8001' до '8013' для закрытия логического канала, пронумерованного в P2 (любое другое значение P1-P2 зарезервировано для использования в будущем)
Поле L_c	Отсутствует для кодирования $N_c = 0$
Поле данных	Отсутствует
Поле L_e	Отсутствует для кодирования $N_e = 0$, присутствует для кодирования $N_e = 1$

Поле данных	Отсутствует (P1-P2 не установлены на '0000'), или от '01' до '13' (P1-P2 установлены на '0000')
SW1-SW2	См. таблицы 5 и 6, где соответствие, например '6200', '6881', '6A81'

7.2 Обработка единицы данных

7.2.1 Единицы данных

В рамках каждого EF, поддерживающего единицы данных, обращение к любой единице данных должно осуществляться при помощи смещения. Начиная с нуля для первой единицы данных файла EF, смещение увеличивается на единицу для каждой последующей единицы данных. Элемент данных «смещение» бинарно кодирован при минимальном числе байт. Ссылка на единицу данных, не содержащуюся в EF, является ошибкой.

Карта может предоставить байт кодирования данных (см. таблицу 87) в байтах предыстории (см. 8.1.1), в EF.ATR (см. 8.2.1.1) и в контрольной информации любого файла (см. тег '82' в таблице 12). Байт кодирования данных фиксирует размер единицы данных.

- Если карта предоставляет байты кодирования данных в разных местах, то байт кодирования данных, действительный для данного EF, находится на ближайшей позиции к этому EF, в пределах пути от MF до этого EF.

- В отсутствие индикации в рамках пути, размер единицы данных — один байт (значение по умолчанию) для этого EF.

7.2.2 Общие положения

Любая команда из этой группы должна быть прервана, если она применяется к файлу EF, не поддерживающему единицы данных. Команда может быть выполнена в EF, только если состояние защиты удовлетворяет атрибутам секретности, определенным для функций: считывания, записи, обновления, стирания или поиска.

Каждая команда из этой группы может использовать либо короткий идентификатор EF, либо идентификатор файла. Если во время выдачи команды имеется текущий EF, то обработка может завершиться этим EF путем установки всех соответствующих бит на 0. Если обработка завершилась, то идентифицируемый EF становится текущим.

INS P1 P2 — Все команды этой группы должны использовать бит 1 для INS и бит 8 для P1 следующим образом:

- если бит 1 для INS установлен на 0, а бит 8 для P1 — на 1, то биты 7 и 6 для P1 устанавливают на 00 (RFU), биты с 5 по 1 для P1 кодируют короткий идентификатор, а P2 (восемь бит) кодируют смещение от нуля до 255;

- если бит 1 для INS установлен на 0, а бит 8 для P1 — на 0, то P1-P2 (пятнадцать бит) кодируют смещение от нуля до 32767;

- если бит 1 для INS установлен на 1, то P1-P2 должны идентифицировать EF. Если первые семь бит в P1-P2 установлены на 0 и если биты с 5 по 1 не равны и если карта и/или EF поддерживает выбор с помощью короткого идентификатора EF, то биты с 5 по 1 для P2 кодируют короткий идентификатор EF

(число от одного до тридцати). В противном случае, P1-P2 — это идентификатор файла. P1-P2, установленные на '0000' идентифицируют текущий EF. Как минимум один информационный объект «смещение» с тегом '54' должен присутствовать в поле данных команды. Данные, если они присутствуют в поле данных команды или ответа, должны быть инкапсулированы в произвольный информационный объект с тегом '53' или '73'.

В этой группе команд SW1-SW2, установленные на '63CX', указывают на успешный обмен состояния памяти, но после внутренней программы повторений; 'X' > '0' кодирует число повторных попыток; 'X' = '0' означает, что счетчик не предусмотрен.

7.2.3 Команда READ BINARY

Поле данных ответа передает содержимое (его часть) файла EF, поддерживающего единицы данных.

Если поле L_e содержит только установленные на '00' байты, то все байты до конца файла должны считываться в пределах 256 для короткого поля L_e, или 65536 для расширенного поля L_e.

Таблица 42 — Пара команда-ответ READ BINARY

CLA INS P1-P2	Как определено в 5.1.1 'B0' или 'B1' См. 7.2.2
Поле L_c	Отсутствует для кодирования $N_c = 0$, присутствует для $N_c > 0$
Поле данных	Отсутствует (INS = 'B0'), или информационный объект «смещение» (INS = 'B1')
Поле L_e	Присутствует для кодирования Ne > 0

Поле данных	Данные считывания (INS = 'B0'), или произвольный информационный объект для инкапсуляции данных считывания (INS = 'B1')
SW1-SW2	См. таблицы 5 и 6, где соответствие, например '6281', '6282', '6700', '6981', '6982', '6986', '6A81', '6A82', '6B00', '6CXX'

7.2.4 Команда WRITE BINARY

Команда инициирует одну из следующих операций в EF в соответствии с атрибутами файла:

- однократной записи бит, передаваемых в поле данных команды (команда должна прерываться, если строка единиц данных не находится в логическом состоянии после стирания);
- логического OR над битами, уже присутствующими в карте, и битами, передаваемыми в поле данных команды (логическое состояние битов файла после стирания представлено нулем);
- логического AND над битами, уже присутствующими в карте, и битами, передаваемыми в поле данных команды (логическое состояние битов файла после стирания представлено единицей).

По умолчанию, т.е. когда байт кодирования данных (см. таблицу 87) отсутствует в байтах предыстории (см. 8.1.1), в EF.ATR (см. 8.2.1.1) и в контрольных параметрах (см. тег '82' в таблице 12) каждого файла в пределах пути от MF до переданного EF, логическое OR должно применяться для этого EF.

Пара команда-ответ WRITE BINARY представлена в таблице 43.

Таблица 43 — Пара команда-ответ WRITE BINARY

CLA INS P1-P2	Как определено в 5.1.1 'D0' или 'D1' См. 7.2.2
Поле L_c	Присутствует для кодирования Nc > 0
Поле данных	Строка единиц данных, которые должны быть записаны (INS = 'D0'), или информационный объект «смещение» и произвольный информационный объект для инкапсуляции строки единиц данных, которые должны быть записаны (INS = 'D1')
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например '63CX' (см. 7.2.2),'6581, '6700', '6981', '6982', '6B00' (смещение за пределами файла EF)

7.2.5 Команда UPDATE BINARY

Команда инициирует обновление бит, уже присутствующих в EF, битами, передаваемыми в поле данных команды. Когда обработка завершается, каждый бит в каждой определенной единице данных будет иметь значение, определенное в поле данных команды.

Пара команда-ответ UPDATE BINARY представлена в таблице 44.

Таблица 44 — Пара команда-ответ UPDATE BINARY

CLA INS P1-P2	Как определено в 5.1.1 'D6' или 'D7' См. 7.2.2
Поле L_c	Присутствует для кодирования $N_c > 0$
Поле данных	Строка единиц данных, которые должны быть обновлены (INS = 'D6'), или информационный объект «смещение» и произвольный информационный объект для инкапсуляции строки необновленных единиц данных, (INS = 'D7')
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например '63CX' (см. 7.2.2),'6581, '6700', '6981', '6982', '6B00' (смещение выходит за пределы EF)

7.2.6 Команда SEARCH BINARY

Команда инициирует поиск в пределах EF, поддерживающего единицы данных. Поле данных ответа выдает смещение единицы данных: строка байт для возвращенного смещения в пределах EF должна иметь тоже значение, что и строка поиска в поле данных команды. Поле данных ответа отсутствует либо вследствие того, что поле L_e отсутствует, либо вследствие того, что не найдено соответствие. Если строка поиска отсутствует, то поле данных ответа выдает смещение первой единицы данных в логическом состоянии после стирания.

Пара команда-ответ SERCH BINARY представлена в таблице 45.

Таблица 45 — Пара команда-ответ SERCH BINARY

CLA INS P1-P2	Как определено в 5.1.1 'A0' или 'A1' См. 7.2.2
Поле L_c	Отсутствует для кодирования $N_c = 0$, присутствует для кодирования $N_c > 0$
Поле данных	Отсутствует или строка поиска (INS = 'A0'), или информационный объект «смещение» и произвольный информационный объект для инкапсуляции строки поиска (INS = 'A1')
Поле L_e	Отсутствует для кодирования $N_e = 0$, присутствует для кодирования $N_e > 0$

Поле данных	Отсутствует или смещение первой единицы данных, совпадающей с полем данных команды (INS = 'A0'), или информационный объект «смещение», указывающий первую единицу данных, совпадающую со строкой поиска (INS = 'A1')
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6282', '6982'

7.2.7 Команда ERASE BINARY

Команда устанавливает содержимое (его часть) файла EF в его логическое состояние после стирания последовательно, начиная с заданного смещения.

- Если INS = '0E', то поле данных команды, если имеется, кодирует смещение первой единицы данных, не подлежащей стиранию. Это смещение должно превышать смещение, закодированное в P1-P2. Если поле данных отсутствует, то команда стирает файл до конца.

- Если INS = '0F', то поле данных команды, если имеется, должно состоять из нуля, одного или двух информационных объектов «смещение». Если смещения нет, то команда стирает все единицы данных в

файле. Если смещение одно, то оно указывает первую единицу данных, подлежащую стиранию: далее команда выполняет стирание до конца файла. Два смещения определяют последовательность единиц данных: второе смещение указывает первую единицу данных, не подлежащую стиранию; оно должно быть выше, чем первое смещение.

Пара команда-ответ ERASE BINARY представлена в таблице 46.

Таблица 46 — Пара команда-ответ ERASE BINARY

CLA INS P1-P2	Как определено в 5.1.1 '0E' или '0F' См. 7.2.2
Поле L_c	Отсутствует для кодирования $N_c = 0$, присутствует для кодирования $N_c > 0$
Поле данных	Отсутствует или смещение первой единицы данных, не подлежащей стиранию (INS = '0E'), или Отсутствует или два информационных объекта «смещение» (INS = '0F')
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '63CX' (см. 7.2.2) '6581', '6700', '6981', '6982', '6B00' (смещение выходит за пределы EF)

7.3 Обработка записи

7.3.1 Записи

В пределах любого файла, поддерживающего запись, обращение к каждой записи может осуществляться при помощи номера записи и/или идентификатора записи. Обращение к записи, не содержащейся в EF, является ошибкой.

Обращение посредством номера записи. Каждый номер записи является уникальным и последовательным.

- В пределах каждого EF, поддерживающего линейную структуру, номера записей должны последовательно присваиваться при осуществлении операции записи или операции присоединения записи, т. е. в порядке создания; первая запись (номер один) будет представлять собой первую созданную запись.

- В пределах каждого EF, поддерживающего циклическую структуру, номера записей должны последовательно присваиваться в обратном порядке, т. е. первая запись (номер один) будет представлять собой последнюю созданную запись.

Как для линейных, так и для циклических структур установлено следующее дополнительное правило: значение ноль должно относиться к текущей записи, т. е. к записи, обращение к которой происходит с помощью указателя записи.

Обращение посредством идентификатора записи. Каждый идентификатор записи предоставляется приложением. Несколько записей могут иметь один и тот же идентификатор записи, в этом случае данные, содержащиеся в записях, могут использоваться для их различения. Если запись представляет собой информационный объект SIMPLE-TLV в поле данных, то идентификатором записи является первый байт этого информационного объекта, т.е. тег SIMPLE-TLV.

Обращение посредством идентификатора записи должно приводить в действие управление указателя записи. Процедура восстановления карты, команда SELECT и любая команда, использующая разрешенный короткий идентификатор EF для организации доступа к EF, могут воздействовать на указатель записи. Обращение посредством номера записи не должно воздействовать на указатель записи.

Всякий раз, когда обращение осуществляется с идентификатором записи, должна быть указана логическая позиция целевой записи: первое или последнее вхождение записи, следующее или предыдущее вхождение по отношению к указателю записи.

- В пределах каждого EF, поддерживающего линейную структуру, логические позиции должны последовательно присваиваться при осуществлении операции записи или операции присоединения записи, т. е. в порядке создания. Первая созданная запись будет находиться в первой логической позиции.

- В пределах каждого EF, поддерживающего циклическую структуру, логические позиции должны последовательно присваиваться в обратном порядке, т. е. в первой логической позиции будет находиться последняя созданная запись.

Как для линейных, так и для циклических структур установлены следующие дополнительные правила:

- первым вхождением должна быть запись с заданным идентификатором и в первой логической позиции; последним вхождением должна быть запись с заданным идентификатором и в последней логической позиции;

- если текущая запись имеется, то следующим вхождением должна быть ближайшая запись с заданным идентификатором, но в восходящей логической позиции по отношению к текущей записи; предыдущим вхождением должна быть ближайшая запись с заданным идентификатором, но в нисходящей логической позиции по отношению к текущей записи;

- если текущая запись отсутствует, то следующее вхождение должно быть эквивалентно первому вхождению; предыдущее вхождение должно быть эквивалентно последнему вхождению;

- значение ноль должно относиться к первой, последней, следующей или предыдущей записи в порядке нумерации, независимо от идентификатора записи.

7.3.2 Общие положения

Любая команда из этой группы должна быть прервана, если она применяется к файлу EF, не поддерживающему записи. Команда может быть выполнена в EF, только если состояние защиты удовлетворяет атрибутам секретности, определенным для функций: считывания, записи, присоединения, обновления, стирания, поиска, активирования или дезактивирования.

Записи в EF могут поддерживать состояния жизненного цикла записи. В таком случае обычно дезактивированные записи не имеют доступа с помощью команд READ RECORD, WRITE RECORD, ERASE RECORD и APPEND RECORD. Если такие команды используются, то соответствующие команды возвращаются с байтами состояний '6287' (как минимум одна из записей, к которой обращаются, дезактивирована). Кроме того, дезактивированные записи должны игнорироваться во время выполнения команды SEARCH RECORD. Дополнительные подробности и исключения из основных правил, изложенных выше, даются в следующих подразделах.

Две команды из этой группы (считать, обновить) могут использовать нечетный код INS (поля данных, закодированные в BER-TLV) для инициирования действия части данной записи (неполное считывание, неполное обновление). Далее смещение должно обращаться к каждому байту внутри записи: от нуля до первого байта записи, смещение увеличивается на единицу для каждого последующего байта записи. Обращение к байту, не содержащемуся в записи, является ошибкой. При необходимости, элемент данных «смещение» бинарно закодирован и связан с тегом '54'. Если данные присутствуют в поле данных команды или ответа, то они должны быть инкапсулированы в произвольный информационный объект с тегом '53' или '73'.

Каждая команда из этой группы может использовать короткий идентификатор EF. Если процесс завершен, то идентифицируемый EF становится текущим и указатель записи возвращается в исходное состояние. Если во время вызова команды имеется текущий EF, то процесс может завершиться без указания EF (путем установки соответствующих пяти бит на 0).

P1. Каждый номер записи или идентификатор — это число от одного до 254, закодированное с помощью значения P1 от '01' до 'FE'. Ноль (закодированный как '00') зарезервирован для особого назначения. 255 (закодированное, как 'FF') зарезервировано для использования в будущем.

P2. Биты с 8 по 4 представляют собой короткий идентификатор в соответствии с таблицей 47. Биты с 3 по 1 зависят от команды.

Таблица 47 — Короткий идентификатор в P2

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0	0	0	0	0	—	—	—	Текущий EF
Не все равны					—	—	—	Короткий идентификатор EF (число от нуля до тридцати)
1	1	1	1	1	—	—	—	Зарезервировано для использования в будущем

В этой группе команд SW1-SW2, установленные на '63CX', указывают на успешный обмен состояний памяти, но после использования внутренней программы повторений. 'X' > '0' указывает число повторных попыток; 'X' = '0' означает, что счетчик не предусмотрен.

7.3.3 Команда READ RECORD(S)

Поле данных ответа передает содержимое (или его части) указанной(ых) записи(ей) (или начальной части одной записи) файла EF.

Если какая-либо запись, обращение к которой происходит с помощью P1 и P2, находится в состоянии жизненного цикла записи DEACTIVATED, то команда завершается с байтами состояния '6287', а поле данных ответа должно быть пустым.

Если INS = 'B2' и если записи представляют собой информационные объекты SIMPLE-TLV (см. 5.2.1), то для этого случая в таблице 50 иллюстрировано поле данных ответа. Сравнение N_r со структурой TLV указывает, является ли уникальная запись (считывание одной записи) или последняя запись (считывание всех записей) незавершенной, завершенной или присоединенной.

Примечание — Если записями не являются информационные объекты, то результатом функции считывания всех записей будет получение записей без их разграничения.

Если INS = 'B3', то команда считывает по частям запись, обращение к которой происходит с помощью P1. Поле данных команды должно содержать информационный объект «смещение» (тег '54'), указывающий какой байт первым должен считываться при записи. Поле данных ответа должно содержать произвольный информационный объект (тег '53'), инкапсулирующий данные считывания.

Пара команда-ответ READ RECORD(S) и кодирование P2 представлены в таблицах 48 и 49.

Таблица 48 — Пара команда-ответ READ RECORD(S)

CLA INS P1 P2	Как определено в 5.1.1 'B2' или 'B3' Номер записи или идентификатор записи ('00' обращается на текущую запись) См. таблицу 49
Поле L_c	Отсутствует для кодирования $N_c = 0$, присутствует для кодирования $N_c > 0$
Поле данных	Отсутствует (INS = 'B2'), или информационный объект «смещение» (INS = 'B3')
Поле L_e	Присутствует для кодирования $N_e > 0$

Поле данных	Данные считывания (INS = 'B2'), или произвольный информационный объект для инкапсуляции данных считывания (INS = 'B3')
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6281', '6282', '6700', '6981', '6982', '6A81', '6A82', '6A83', '6CXX'

Таблица 49 — P2

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
x	x	x	x	x	—	—	—	Короткий идентификатор EF в соответствии с таблицей 47
—	—	—	—	—	0	x	x	**Идентификатор записи в P1**:
—	—	—	—	—	0	0	0	- Считывание первого вхождения;
—	—	—	—	—	0	0	1	- Считывание последнего вхождения;
—	—	—	—	—	0	1	0	- Считывание следующего вхождения;
—	—	—	—	—	0	1	1	- Считывание предыдущего вхождения.
—	—	—	—	—	1	x	x	**Номер записи в P1**:
—	—	—	—	—	1	0	0	- Считывание записи с номером в P1;
—	—	—	—	—	1	0	1	- Считывание всех записей, начиная с записи с номером в P1 и заканчивая последней;
—	—	—	—	—	1	1	0	- Считывание всех записей, начиная с последней и заканчивая записью с номером в P1;
—	—	—	—	—	1	1	1	Зарезервировано для использования в будущем.

Если поле L_e содержит только установленные на '00' байты, то команда должна обеспечивать полное считывание либо одной запрашиваемой записи, либо запрашиваемой последовательности записей, в зависимости от последовательности бит 3, 2 и 1 в байте P2 и в пределах максимума 256 для короткого поля L_e или 65536 для расширенного поля L_e.

Таблица 50 — Поле данных ответа с INS = 'B2'.

Случай a — Частичное считывание одной записи (поле L_e содержит не только установленные на '00' байты)

T_n (один байт)	L_n (один или три байта)	Первые байты V_n

N_r байтов

Случай b — Полное считывание одной записи (поле L_e содержит только установленные на '00' байты)

T_n (один байт)	L_n (один или три байта)	Все байты данных записи (L_n байтов)

Случай c — Частичное считывание последовательности записей (поле L_e содержит не только установленные на '00' байты)

$T_n — L_n — V_n$	...	$T_{n+m} — L_{n+m} — V_{n+m}$ (Первые байты записи)

N_r байтов

Случай d — Считывание нескольких записей до конца файла (поле L_e содержит только установленные на '00' байты)

$T_n — L_n — V_n$	...	$T_{n+m} — L_{n+m} — V_{n+m}$

7.3.4 Команда WRITE RECORD

Команда инициирует одну из следующих операций в пределах EF:

- однократную запись, передаваемую в поле данных команды (команда должна прерываться в том случае, если запись не находится в логическом состоянии после стирания);
- операцию логического OR над байтами данных записи, уже присутствующей в карте, и байтами данных записи, передаваемой в поде данных команды,
- операцию логического AND над байтами данных записи, уже присутствующей в карте, и байтами данных записи, передаваемой в поле данных команды.

Если запись, обращение к которой происходит с помощью P1 и P2, находится в состоянии жизненного цикла записи DEACTIVATED, то команда завершается с байтами состояний '6287' без изменения содержания записи.

По умолчанию, т. е., если байт кодирования данных (см. таблицу 87) отсутствует в байтах предыстории (см. 8.1.1), в EF.ATR (см. 8.2.1.1) и контрольных параметрах (см. тег '82' в таблице 12) каждого файла в пределах пути от MF до заданного EF, то для этого EF должна применяться операция логического OR.

При использовании адресации текущей записи команда должна устанавливать указатель записи на успешно записанную запись.

Если команда применяется к EF, поддерживающему циклическую структуру с записями фиксированного размера, то опция команды «предыдущая» (биты 3,2 и 1 в P2 установлены на 011) ведет себя так же, как и команда APPEND RECORD.

Если записи представляют собой информационный объект SIMPLE-TLV (см. 5.2.1), то для этого случая в таблице 53 проиллюстрировано поле данных команды.

Пара команда-ответ WRITE RECORD и кодирование P2 представлены в таблицах 51 и 52.

Таблица 51 — Пара команда-ответ WRITE RECORD

CLA INS P1 P2	Как определено в 5.1.1 'D2' Номер записи ('00' обращается на текущую запись) См. таблицу 52
Поле L_c	Присутствует для кодирования $N_c > 0$
Поле данных	Запись, подлежащая операции записи
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '63CX' (см. 7.2.2),'6581, '6700', '6981', '6982', '6986', '6A81', '6A82', '6A83', '6A84', '6A85'

Таблица 52 — P2

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
x	x	x	x	x	—	—	—	Короткий идентификатор EF в соответствии с таблицей 47
—	—	—	—	—	0	x	x	**P1, установленный на '00'**:
—	—	—	—	—	0	0	0	- Первая запись;
—	—	—	—	—	0	0	1	- Последняя запись;
—	—	—	—	—	0	1	0	- Следующая запись;
—	—	—	—	—	0	1	1	- Предыдущая запись.
—	—	—	—	—	1	0	0	**Номер записи в P1**:
Примечание — Любое другое значение SW2 зарезервировано для использования в будущем ИСО/МЭК СТК 1/ПК 17.								

Таблица 53 — Поле данных команды (полная запись одной записи)

T_n (один байт)	L_n (один или три байта)	Все байты V_n

7.3.5 Команда UPDATE RECORD

Команда инициирует обновление конкретной записи битами, передаваемыми в поле данных команды. При использовании адресации текущей записи команда должна устанавливать указатель записи на успешно обновленную запись.

- Если команда применяется к EF, поддерживающему линейную или циклическую структуру с записями фиксированного размера, то она должна прерываться в том случае, если длина передаваемой записи отличается от длины существующей записи.

- Если команда применяется к EF, поддерживающему линейную структуру с записями переменной длины, то она может быть выполнена, когда длина передаваемой записи отличается от длины существующей записи.

- Если команда применяется к EF, поддерживающему циклическую структуру с записью фиксированного размера, то опция команды «предыдущая» (биты 3, 2 и 1 в P2 установлены на 011), ведет себя так же, как команда APPEND RECORD.

- Если запись, обращение к которой происходит с помощью P1 и P2, находится в состоянии жизненного цикла записи DEACTIVATED, то команда завершается байтами состояний '6287' без изменения содержания записи.

Пара команда-ответ UPDATE RECORD представлена в таблице 54.

Таблица 54 — Пара команда-ответ UPDATE RECORD

CLA INS P1 P2	Как определено в 5.1.1 'DC' или 'DD' Номер записи ('00' обращается на текущую запись) См. таблицу 52 (INS = 'DC') или 55 (INS = 'DD')
Поле L_c	Присутствует для кодирования $N_c > 0$
Поле данных	Данные, подлежащие обновлению (INS = 'DC'), или информационный объект «смещение» и произвольный информационный объект для инкапсуляции данных, подлежащих обновлению (INS = 'DD')
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '63CX' (см. 7.2.2),'6581, '6700', '6981', '6982', '6986', '6A81', '6A82', '6A83', '6A84', '6A85'

Если INS = 'DC' и если записи являются информационными объектами SIMPLE-TLV (см. 5.2.1), то для этого случая в таблице 53 для наглядности представлено поле данных команды.

Если INS = 'DD', то команда частично обновляет запись, обращение к которой происходит с помощью P1. Поле данных команды должно содержать информационный объект «смещение» (тег '54') для указания какой байт должен быть обновлен первым в записи и произвольный информационный объект (тег '53' или '73') для инкапсуляции данных, подлежащих обновлению.

Таблица 55 – P2 с INS = 'DD'

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
x	x	x	x	x	—	—	—	Короткий идентификатор EF в соответствии с таблицей 47
—	—	—	—	—	1	x	x	**Номер записи в P1**:
—	—	—	—	—	1	0	0	- Замена;
—	—	—	—	—	1	0	1	- Логическое AND;
—	—	—	—	—	1	1	0	- Логическое OR;
—	—	—	—	—	1	1	1	- Логическое XOR.
Примечание — Любое другое значение зарезервировано для использования в будущем ИСО/МЭК СТК 1 ПК 17.								

7.3.6 Команда APPEND RECORD

Команда инициирует либо регистрацию новой записи в конце EF, поддерживающего линейную структуру, либо регистрация одного номера записи в EF, поддерживающего циклическую структуру. При использовании адресации текущей записи команда должна устанавливать указатель записи на успешно обновленную запись.

Если команда применяется к EF, поддерживающему линейную структуру, с заполненными записями, то команда прерывается из-за недостаточной области памяти в файле.

Если команда применяется к EF, поддерживающему циклическую структуру, с заполненными записями, то запись с наибольшим номером заменяется. Она становится записью номер один. Если запись с наибольшим номером записи находится в состоянии жизненного цикла записи DEACTIVATED, то команда завершается байтами состояний '6287' без изменения содержания записи или номера записи.

Если записи в EF имеют состояния жизненного цикла записи, то состояние добавленной записи должно быть установлено в ACTIVATED до тех пор, пока не будет указано иное.

Если записи являются информационными объектами SIMPLE-TLV (см. 5.2.1), то для этого случая в таблице 53 для наглядности представлено поле данных команды.

Пара команда-ответ APPEND RECORD представлена в таблице 56.

Таблица 56 — Пара команда-ответ APPEND RECORD

CLA INS P1 P2	Как определено в 5.1.1 'E2' '00' (любое другое значение не действительно) См. таблицу 47 с битами с 3 по 1, установленными на 000 (любое другое значение зарезервировано для использования в будущем)
Поле L_c	Присутствует для кодирования $N_c > 0$
Поле данных	Записи, подлежащие присоединению
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '63CX' (см. 7.2.2),'6581, '6700', '6981', '6982', '6986', '6A81', '6A82', '6A83', '6A84', '6A85'

7.3.7 Команда SEARCH RECORD

Команда инициирует простой или расширенный или проприетарный поиск записей, сохраненных в пределах EF. Поиск может быть ограничен записями с заданным идентификатором или записями с номером, большим или меньшим, чем заданный номер. Он может выполняться по возрастающему или убывающему порядку номера записи. Поиск начинается либо с первого байта записи (простой поиск), либо с заданного смещения в пределах записи (расширенный поиск), либо с первого вхождения заданного байта в пределах записи (расширенный поиск). Поле данных ответа выдает номера записей, совпадающих по критерию поиска в пределах EF, поддерживающего записи. Команда должна установить указатель записи на первую запись, совпадающую по критерию поиска.

В EF, поддерживающем записи переменной длины с линейной структурой, поиск не должен учитывать записи, которые короче чем строка поиска. В EF, поддерживающем записи фиксированного размера с линейной или циклической структурой, если строка поиска длиннее, чем записи, то карта должна прервать команду.

Записи с состоянием жизненного цикла записи, установленным в DEACTIVATED, должны игнорироваться во время поиска.

Пара команда-ответ SERCH RECORD и кодирование P2 представлены в таблицах 57 и 58.

Таблица 57 — Пара команда-ответ SERCH RECORD

CLA INS P1 P2	Как определено в 5.1.1 'A2' Номер записи или идентификатор записи ('00' обращается на текущую запись) См. таблицу 58
Поле L_c	Присутствует для кодирования $N_c > 0$
Поле данных	Строка поиска (биты 3 и 2 в P1 не установлены на 11, простой поиск), или Индикация поиска (2 байта) с последующей строкой поиска (биты 3,2 и 1 в P2 установлены на 110, расширенный поиск), или Проприетарное (биты 3,2 и 1 в P2 установлены на 111, проприетарный поиск)
Поле L_e	Отсутствует для кодирования $N_e = 0$, присутствует для кодирования $N_e > 0$

Поле данных	Отсутствует или номера записей
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6282', '6982', '6CXX'

Примечания

1 Поле данных ответа отсутствует либо потому, что поле L_e отсутствует, либо потому что не найдено соответствия.

2 Поле данных ответа не выдает идентификаторов записи, потому что они могут быть не уникальными.

Таблица 58 — P2

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
x	x	x	x	x	—	—	—	Короткий идентификатор EF в соответствии с таблицей 47
—	—	—	—	—	0	x	x	**Простой поиск с идентификатором записи в P1**:
—	—	—	—	—	0	0	0	- В прямом направлении от первого вхождения;
—	—	—	—	—	0	0	1	- В обратном направлении от последнего вхождения;
—	—	—	—	—	0	1	0	- В прямом направлении от следующего вхождения;
—	—	—	—	—	0	1	1	- В обратном направлении от предыдущего вхождения.
—	—	—	—	—	1	0	x	**Простой поиск с номером записи в P1**:
—	—	—	—	—	1	0	0	- В прямом направлении от P1;
—	—	—	—	—	1	0	1	- В обратном направлении от P1;
—	—	—	—	—	1	1	0	**Расширенный поиск** (см. таблицу 59)
—	—	—	—	—	1	1	1	**Проприетарный поиск**

В расширенном поиске (биты 3, 2 и 1 в P2 установлены на 110) поле данных команды состоит из индикации поиска из двух байтов с последующей строкой поиска. В таблице 59 определен первый байт индикации поиска. В соответствии с первым байтом индикации поиска второй байт представляет собой либо смещение, либо значение, т. е. поиск в записях должен начаться либо с этого смещения (абсолютная позиция), либо после первого вхождения этого значения.

Таблица 59 — P2

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0	0	0	0	0	—	—	—	Последующий байт — смещение (начало с этой позиции)
0	0	0	0	1	—	—	—	Последующий байт — значение (начало после первого вхождения)
—	—	—	—	—	0	x	x	**Идентификатор записи в P1**:
—	—	—	—	—	0	0	0	- В прямом направлении от первого вхождения;
—	—	—	—	—	0	0	1	- В обратном направлении от последнего вхождения;
—	—	—	—	—	0	1	0	- В прямом направлении от следующего вхождения;
—	—	—	—	—	0	1	1	- В обратном направлении от предыдущего вхождения.
—	—	—	—	—	1	x	x	**Номер записи в P1**:
—	—	—	—	—	1	0	0	- В прямом направлении от P1;
—	—	—	—	—	1	0	1	- В обратном направлении от P1;
—	—	—	—	—	1	1	0	- В прямом направлении от следующей записи;
—	—	—	—	—	1	1	1	- В обратном направлении от предыдущей записи.

Примечание — Любое другое значение зарезервировано для использования в будущем ИСО/МЭК СТК1 ПК 17.

7.3.8 Команда ERASE RECORD(S)

Команда устанавливает одну или несколько записей в EF в логическое состояние после стирания, либо запись, обращение к которой происходит с помощью P1, либо последовательность записей из P1 последовательно до конца файла. Стертые записи не должны удаляться, доступ к ним еще может быть осуществлен с помощью команд WRITE RECORD и UPDATE RECORD.

Если какая-нибудь запись, обращение к которой происходит с помощью P1 и P2, находится в состоянии жизненного цикла записи DEACTIVATED, то команда завершается с байтами состояний '6287' без изменения содержания записи.

Пара команда-ответ ERASE RECORD(S) и кодирование P2 представлены в таблицах 60 и 61.

Таблица 60 — Пара команда-ответ ERASE RECORD(S)

CLA INS P1 P2	Как определено в 5.1.1 '0C' Номер записи См. таблицу 61
Поле L_c	Присутствует для кодирования $N_c > 0$
Поле данных	Отсутствует
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '63CX' (см. 7.2.2),'6581, '6700', '6981', '6982', '6986', '6A81', '6A82', '6A83', '6A84', '6A85'

Таблица 61 — P2

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
x	x	x	x	x	—	—	—	Короткий идентификатор EF в соответствии с таблицей 47
—	—	—	—	—	1	x	x	**Номер записи в P1**:
—	—	—	—	—	1	0	0	- Стереть запись P1;
—	—	—	—	—	1	0	1	- Стереть все записи, начиная с записи с номером в P1 и заканчивая последней;
Примечание — Любое другое значение зарезервировано для использования в будущем ИСО/МЭК СТК1 ПК 17.								

7.3.9 Команда ACTIVATE RECORD

Команда ACTIVATE RECORD устанавливает запись, обращение к которой происходит с помощью P1 и P2, в состояние жизненного цикла записи ACTIVATED. Команда не должна воздействовать на указатель записи.

Если EF, обращение к которому происходит с помощью P2, не поддерживает состояния жизненного цикла записи, то команда должна быть прервана с байтами состояния '6981'.

Если адресуемая запись уже активирована, то команда должна вернуть байты состояния '9000'.

Пара команда-ответ ACTIVATE RECORD и кодирование P2 представлены в таблицах Изм. 1-2 и Изм. 1-3.

Таблица Изм. 1-2 — Пара команда-ответ ACTIVATE RECORD(S)

CLA INS P1 P2	Как определено в 5.1.1 '08' Номер записи См. таблицу Изм. 1-3
Поле L_c	Отсутствует
Поле данных	Отсутствует
Поле L_e	Отсутствует

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6981', '6982', '6986', '6A82', '6A83'

Таблица Изм. 1-3 — P2

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
x	x	x	x	x	—	—	—	Короткий идентификатор EF в соответствии с таблицей 47
—	—	—	—	—	1	x	x	**Номер записи в P1**:
—	—	—	—	—	1	0	0	- Активировать запись P1;
Примечание — Любое другое значения зарезервировано для использования в будущем ИСО/МЭК СТК1 ПК 17.								

7.3.10 Команда DEACTIVATE RECORD

Команда DEACTIVATE RECORD устанавливает запись, обращение к которой осуществляется с помощью P1 и P2, в состояние жизненного цикла записи DEACTIVATED. Команда не должна воздействовать на указатель записи.

Если EF, обращение к которому происходит с помощью P2, не поддерживает состояния жизненного цикла записи, то команда должна быть прервана с байтами состояния '6981'.

Если адресуемая запись уже дезактивирована, то команда должна вернуть байты состояния '9000'.

Для активации всех записей в состояние жизненного цикла DEACTIVATED, может использоваться команда ACTIVATE FILE. Независимо от изменения состояния жизненного цикла файла, присутствующего дополнительно (см. ИСО/МЭК 7816-9), все записи будут активированы. Для активации личных записей должна использоваться команда ACTIVATE RECORD.

Пара команда-ответ DEACTIVATE RECORD и кодирование P2 представлены в таблицах Изм. 1-4 и Изм. 1-5.

Таблица Изм. 1-4 — Пара команда-ответ DEACTIVATE RECORD

CLA INS P1 P2	Как определено в 5.1.1 '08' Номер записи См. таблицу Изм. 1-5
Поле L_c	Отсутствует
Поле данных	Отсутствует
Поле L_e	Отсутствует

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6981', '6986', '6A82', '6A83'

Таблица Изм. 1-5 — P2

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
x	x	x	x	x	—	—	—	Короткий идентификатор EF в соответствии с таблицей 47
—	—	—	—	—	1	x	x	**Номер записи в P1**:
—	—	—	—	—	1	0	0	- Дезактивировать запись P1;
Примечание — Любое другое значения зарезервировано для использования в будущем ИСО/МЭК СТК1 ПК 17.								

7.4 Обработка информационного объекта

7.4.1 Общие положения

Любая команды из этой группы может быть прервана, если она применяется к структуре (DF или EF), не поддерживающей информационные объекты. Она может быть выполнена только, если состояние защиты удовлетворяет условиям секретности, определенным для приложения в пределах контекста функции.

INS P1 P2. Все команда этой группы могут использовать нечетный код INS (см. 5.1.2). Бит 1 в INS должен использоваться вместе с P1-P2 в соответствии с таблицей 62.

Таблица 62 — P1-P2

Условие	Значение P1-P2	Смысловое содержание
Четный код INS	'0000'	Используется для демпинга файла (см. 8.4), или для строк байтов, представленных картой (см. 8.6)
	От '0040' до '00FF'	Тег BER-TLV (один байт) в P2
	От '0100' до '01FF'	Проприетарный
	От '0200' до '02FF'	Тег SIMPLE-TLV в P2
	От '4000' до 'FFFF'	Тег BER-TLV (два байта) в P1-P2
Нечетный код INS	Любое значение	Идентификатор файла или короткий идентификатор EF (см. текст ниже)
Примечание — Любое другое значение зарезервировано для использования в будущем ИСО/МЭК СТК 1 ПК 17.		

- Если бит 1 в INS установлен на 0, а P1 — на '00', то P2 от '40' до 'FE' должен являться тегом BER-TLV, состоящим из одного байта. Если тег BER-TLV действителен и указывает на составное кодирование, то команда устанавливает соответствующий шаблон в качестве текущего контекста. Значение '00FF' используется либо для получения всех общих информационных объектов BER-TLV, читаемых в контексте, либо для индикации, что это поле данных команды закодировано в BER-TLV.

- Если бит 1 в INS установлен на 0, а P1 — на '01', то P2 от '00' до 'FF' должен быть идентификатором для внутренних испытаний карты и для проприетарных услуг, значимых в пределах заданного контекста приложения.

- Если бит 1 установлен на 0, а P1 — на '02', то P2 от '01' до 'FE' должен быть тегом SIMPLE-TLV. Значение '0200' зарезервировано для использования в будущем. Значение '02FF' используется либо для получения всех общих информационных объектов SIMPLE-TLV, читаемых в контексте, либо для индикации, что поле данных команды закодировано в SIMPLE-TLV.

- Если бит 1 в INS установлен на 0 и если значение P1-P2 находится в диапазоне от '4000' до 'FFFF', то оно должно быть тегом BER-TLV, состоящим из двух байтов. Если тег BER-TLV действителен и указывает на составное кодирование, то команда устанавливает соответствующий шаблон в качестве текущего контекста. Значения, которые не действительны для тега BER-TLV, состоящего из двух байтов (см. 5.2.2.1), зарезервированы для использования в будущем, например, '4000' и 'FFFF'.

- Если бит 1 в INS установлен на 1, то байты P1-P2 должны идентифицировать файл. Если первые одиннадцать бит в P1-P2 установлены на 0 и если биты с 5 по 1 в P2 не все равны и если карта и/или файл поддерживают выбор по короткому идентификатору EF, то биты с 5 по 1 в P2 кодируют короткий идентификатор EF (число от одного до тридцати). В противном случае, байты P1-P2 представляют собой идентификатор файла. Байты P1-P2, установленные на '0000', идентифицируют текущий EF до тех пор, пока поле данных команды обеспечивает информационный объект «ссылка на файл» (тег '51', 5.3.1.2) идентификацией файла. Когда процесс завершается, то идентифицируемый файл становится текущим.

Поля данных. Команды из этой группы должны использовать поля данных следующим образом:

- если бит 1 в INS установлен на 0 и если требуется информационный объект, или он предусмотрен в пределах текущего контекста (например, среды, связанной с конкретным приложением, или текущего DF), то поле данных или сцепление полей данных должно содержать поле значения информационного объекта, т. е. либо соответствующий элемент данных в случае информационного объекта SIMPLE-TLV или простой информационный объект BER-TLV, или соответствующий шаблон в случае составного информационного объекта BER-TLV;

- с обоими кодами INS, если предусмотрен набор информационных объектов или если требуется содержание EF, то соответствующее поле данных должно содержать информационный(е) объект(ы).

7.4.2 Команда GET DATA

Команда извлекает либо содержание EF, поддерживающего информационные объекты, либо один информационный объект, возможно составной, в пределах текущего контекста (например, среды, связанной с конкретным приложением, или текущего DF).

Примечание — Если информация слишком длинная для одного поля данных ответа, то карта должна вернуть начало информации с последующими SW1-SW2, установленными на '61XX'. Тогда следующая команда GET RESPONSE обеспечивает 'XX' байты информации. Процесс может повторяться до того, как карта пошлет SW1-SW2, установленные на '9000'.

Если INS = 'CB', то поле данных команды должно содержать либо информационный объект «список тегов», либо информационный объект «список заголовков», либо информационный объект «расширенный список заголовков» (теги '5C', '5D', '4D', см. 8.5.1).

- В случае списка тегов поле данных ответа должно представлять собой сцепление информационных объектов, указанных в списке тегов, в том же порядке (один или несколько информационных объектов могут отсутствовать). Пустой список тегов требует все доступные информационные объекты.

- В случае списка заголовков поле данных ответа должно представлять собой сцепление усеченных информационных объектов, указанных в списке заголовков, в том же порядке (один или несколько информационных объектов могут отсутствовать).

- В случае расширенного списка заголовков поле данных ответа должно представлять собой сцепление информационных объектов, полученных из расширенного списка заголовков в соответствии с 8.5.1.

Когда имеется несколько вхождений тега, то настоящий раздел не определяет, какой информационный объект возвращается, так как это зависит от определения, сущности или содержания информационного объекта.

Если физический интерфейс не позволяет карте ответить на восстановление, например, универсальная последовательная шина или доступ с использованием радиочастотной связи, то команда GET DATA может извлечь определенную информацию в карте в соответствии с P1-P2. Из карты может быть извлечена следующая определенная информация:

- при INS = 'CA' о:

- теге '5F51' — Ответ-на-Восстановление — строка из не более 32 байтов в соответствии с ИСО/МЭК 7816-3;

- теге '5F52' — байты предыстории — строка из не более 15 байтов в соответствии с 8.1.1, либо

- при INS = 'CB' и пустом списке тегов об:

- идентификаторе файла '2F00' — содержание EF.DIR представляет собой набор информационных объектов BER-TLV в соответствии с 8.2.1.1;

- идентификаторе файла '2F01' — содержание EF.ATR представляет собой набор информационных объектов BER-TLV в соответствии с 8.2.1.1.

Примечания

1 (справочная информация) В соответствии с физическим интерфейсом, определенным в ИСО/МЭК 7816-3, карта отвечает на любую операцию «холодного» или «горячего» восстановления с помощью контактов. Ответ на восстановление — это последовательность асинхронных знаков. Начальный знак TS указывает правила для декодирования байтов в знаках и предлагает альтернативное измерение элементарной единицы времени. Несмотря на то, что он может быть декодирован в соответствии с указанными правилами, TS представляет собой последовательность синхронизирующих импульсов, а не байт. В соответствии в ИСО/МЭК 7816-3, Ответ-на-Восстановление — это строка не более 32 байтов, передаваемых при ответе на восстановление, а именно обязательный байт формата T0, за которым следует дополнительные байты интерфейса, дополнительные байты предыстории (не более 15 байтов предыстории, кодированные в соответствии с 8.1.1) и условный контрольный байт TCK. Если TCK присутствует, то операции OR над всеми байтами от T0 до TCK включительно являются нулями.

2 Для информации ATR, если поле L_e кодирует число, меньшее чем точная длина, то, карта предпочтительно должна прервать команду путем возврата только SW1-SW2, установленных на '6CYY', для указания точного числа доступных байтов данных, нежели вернуть начало информации с последующими SW1-SW2, установленными на '61XX'. Однако '6C00' обозначает 256 байтов или более. Если имеется более 256 байтов, то SW1-SW2, установленные на '61XX', указывают, что 'XX' байты все еще доступны.

Если поле L_e содержит только установленные на '00' байты, то вся требуемая информация должна быть возвращена в пределах максимума 256 для короткого поля L_e, или 65536 для расширенного поля L_e.

Пара команда-ответ GET DATA представлена в таблице 63.

Таблица 63 — Пара команда-ответ GET DATA

CLA INS P1-P2	Как определено в 5.1.1 'CA' или 'CB' См. таблицу 62
Поле L_c	Отсутствует для кодирования $N_c = 0$, присутствует для кодирования $N_c > 0$
Поле данных	Отсутствует (INS = 'CA'), или информационный объект «список тегов» или информационный объект «(расширенный) список заголовков» (INS = 'CB')
Поле L_e	Присутствует для кодирования $N_e > 0$

Поле данных	Байты данных в соответствии с P1-P2 (INS = 'CA'), или сцепление информационных объектов BER-TLV (INS = 'CB')
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '61XX', от '6202' до '6280', '6281', '6700', '6981', '6982', '6985', '6A81', '6A88' (информационный объект не найден, т.е. ссылочные данные не найдены), '6CXX'

7.4.3 Команда PUT DATA

Команда инициирует управление либо содержанием EF, поддерживающего информационные объекты, либо одним информационным объектом, возможно составным, в пределах текущего контекста (например, среды, связанной с конкретным приложением, или текущего DF). Например, она позволяет посылать команду на выполнение (тег '52') или сертификат держателя карты (тег '7A21'), которые могут быть слишком длинными для одной команды. Если информационный объект является слишком длинным для одной команда, то необходимо применять цепочку команд (см. 5.1.1.1); поле значения информационного объекта является сцеплением полей данных команды.

Определение, сущность или содержание информационных объектов должны вызывать точные функции управления, например, однократная запись, и/или обновление и/или присоединение.

SW1-SW2, установленные на '63CX', указывают на успешный обмен состояния памяти, но после внутренней программы повторений; 'X' > '0' кодирует число повторных попыток; 'X' = '0' означает, что счетчик не предусмотрен.

Пара команда-ответ PUT DATA представлена в таблице 64.

Таблица 64 — Пара команда-ответ PUT DATA

CLA INS P1-P2	Как определено в 5.1.1 'DA' или 'DB' См. таблицу 62
Поле L_c	Присутствует для кодирования $N_c > 0$
Поле данных	Байты данных в соответствии с P1-P2 (INS = 'DA'), или сцепление информационных объектов BER-TLV (INS = 'BD')
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '63CX', '6581', '6700', '6981', '6982', '6985', '6A80', '6A81', '6A84', '6A85'

7.5 Основные средства защиты

7.5.1 Общие положения

Команды этой группы резервируют P1-P2 для обращения к алгоритму и некоторым соответствующим эталонным данным (например, ключу). Если имеется текущий ключ и текущий алгоритм, то команда может их использовать в неявном виде.

P1. Если не указано иное, то P1 обращается к алгоритму, который должен использоваться: либо к криптографическому алгоритму, либо к биометрическому алгоритму (см. ИСО/МЭК 7816-11 [4]). P1, установленный на '00', означает, что информация не предоставлена, т. е. указатель либо известен до подачи команды, либо содержится в поле данных.

P2. Если не указано иное, то P2 классифицирует ссылочные данные в соответствии с таблицей 65. P2, установленный на '00', означает, что информация не предоставлены, т. е. квалификатор либо известен до подачи команды, либо содержится в поле данных. Квалификатором может быть, например, номер пароля, или номер ключа, или короткий идентификатор EF.

Таблица 65 — P2

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
0	0	0	0	0	0	0	0	Информация не предоставлена
0	—	—	—	—	—	—	—	Глобальные ссылочные данные (например, специальный пароль или ключ MF)
1	—	—	—	—	—	—	—	Специфичные ссылочные данные (например, специальный пароль или ключ DF)
—	x	x	—	—	—	—	—	00 (любое другое значение зарезервировано для использования в будущем)
—	—	—	x	x	x	x	x	Квалификатор, т. е. номер эталонных данных или номер секрета

Примечание — Команда MANAGE SECURITY ENVIRONMENT может установить ссылку на алгоритм и/или ссылку на квалификатор данных.

В этой группе команд SW1-SW2, установленные на '6300' или '63CX', указывают, что верификация завершилась неуспешно, 'X' > '0' кодирует число дальнейших разрешенных попыток. SW1-SW2, установленные на '6A88', означают «ссылочные данные не найдены».

7.5.2 Команда INTERNAL AUTHENTICATE

Команда INTERNAL AUTHENTICATE инициирует вычисление картой аутентификационных данных с использованием данных задачи, посылаемой с устройства сопряжения, и соответствующего секрета (например, ключа), хранящегося в карте.

- Если соответствующий секрет присоединен к MF, то команда может применяться для аутентификации карты в целом.

- Если соответствующий секрет присоединен к другому DF, то команда может применяться для аутентификации этого DF.

Успешное выполнение аутентификации может подчиняться успешному завершению предшествующих команд (например, команд VERIFY, SELECT) или выбору (например, соответствующего секрета).

Карта может записать, сколько раз команда была вызвана для того, чтобы ограничить число дальнейших использований соответствующего секрета или алгоритма.

Примечание — Поле данных ответа может включать в себя данные, пригодные для использования дальнейшими защитными функциями (например, случайное число).

Пара команда-ответ INTERNAL AUTHENTICATE представлена в таблице 66.

Таблица 66 — Пара команда-ответ INTERNAL AUTHENTICATE

CLA INS P1-P2	Как определено в 5.1.1 '88' См. 7.5.1 и таблицу 65
Поле L_c	Присутствует для кодирования $N_c > 0$
Поле данных	Данные, касающиеся аутентификации (например, задача)
Поле L_e	Присутствует для кодирования $N_e > 0$

Поле данных	Данные, касающиеся аутентификации (например, ответ на задачу)
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6300' (см. 7.5.1), '63CX' (см. 7.5.1), '6581', '6700', '6982', '6983', '6984', '6A81', '6A82' '6A86', '6A88' (см. 7.5.1)

7.5.3 Команда GET CHALLENGE

Команда требует выдачи задачи (например, в виде случайного числа для криптографической аутентификации или фразы для запроса биометрической аутентификации, использующей запись голоса) для использования в процедуре, связанной с защитой (например, в команде EXTERNAL AUTHENTICATE).

Данная задача действительна, по меньшей мере, для следующей команды; настоящий раздел не определяет дополнительные условия.

Пара команда-ответ GET CHALLENGE представлена в таблице 67.

Таблица 67 — Пара команда-ответ GET CHALLENGE

CLA INS P1 P2	Как определено в 5.1.1 '84' См. 7.5.1 '00' (любое другое значение зарезервировано для использования в будущем)
Поле L_c	Отсутствует для кодирования $N_c = 0$
Поле данных	Отсутствует
Поле L_e	Присутствует для кодирования $N_e > 0$

Поле данных	Задача
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6300' (см. 7.5.1), '63CX' (см. 7.5.1), '6581', '6700', '6982', '6983', '6984', '6A81', '6A82' '6A86', '6A88' (см. 7.5.1)

7.5.4 Команда EXTERNAL AUTHENTICATE

Команда условно обновляет состояние защиты, используя результат (да или нет) вычисления, осуществляемого картой и основанного на данных задачи, ранее выданной картой (например, на команду GET CHALLENGE), ключе, возможно, секретном, хранящемся в карте, и аутентификационных данных, передаваемых устройством сопряжения.

Любое успешное выполнение аутентификации требует использования данных последней задачи, полученной с карты. Карта может регистрировать безуспешные операции аутентификации (например, с целью ограничения числа дальнейшего использования ссылочных данных).

Отсутствие поля данных команды может быть использовано либо для получения числа дальнейших разрешенных попыток 'X' (SW1-SW2, установлены на '63CX'), либо для проверки требуется ли верификация или нет (SW1-SW2 установлены на '9000').

Пара команда-ответ GET CHALLENGE представлена в таблице 68.

Таблица 68 — Пара команда-ответ GET CHALLENGE

CLA INS P1-P2	Как определено в 5.1.1 '82' См. 7.5.1 и таблицу 65
Поле L_c	Отсутствует для кодирования $N_c = 0$, присутствует для кодирования $N_c > 0$
Поле данных	Отсутствует для данных, касающихся аутентификации (например, ответ на задачу)
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6300' (см. 7.5.1), '63CX' (см. 7.5.1), '6581', '6700', '6982', '6983', '6984', '6A81', '6A82' '6A86', '6A88' (см. 7.5.1)

Функция MUTUAL AUTHENTICATE. Функция MUTUAL AUTHENTICATE использует те же функциональные возможности, что и команды EXTERNAL и INTERNAL AUTHENTICATE. Она опирается на предыдущую команду GET CHALLENGE и ключ, возможно секретный, хранящийся в карте. Карта и устройство сопряжения распределяют данные, касающиеся аутентификации и включающие две задачи: одна выдается картой, другая — устройством сопряжения.

Примечание — Команда может использоваться для выполнения аутентификации согласно разделам 2 и 3 ИСО/МЭК 9798 [8].

Операция может быть выполнена, только если состояние защиты удовлетворяет атрибутам секретности для этой операции.

Пара команда-ответ для функции MUTUAL AUTHENTICATE представлена в таблице 69.

Таблица 69 — Пара команда-ответ для функции MUTUAL AUTHENTICATE

CLA INS P1-P2	Как определено в 5.1.1 '82' См. 7.5.1 и таблицу 65
Поле L_c	Присутствует для кодирования $N_c > 0$
Поле данных	Данные, касающиеся аутентификации
Поле L_e	Присутствует для кодирования $N_e > 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6300' (см. 7.5.1), '63CX' (см. 7.5.1), '6581', '6700', '6982', '6983', '6984', '6A81', '6A82' '6A86', '6A88' (см. 7.5.1)

7.5.5 Команда GENERAL AUTHENTICATE

Команда оптимизирует функции EXTERNAL, INTERNAL и MUTUAL AUTHENTICATE; а именно: либо объект во внешней среде аутентифицирует объект в карте (функция INTERNAL AUTHENTICATE), либо объект в карте аутентифицирует объект во внешней среде (функция EXTERNAL AUTHENTICATE), либо и то и другое (функция MUTUAL AUTHENTICATE). Пока предназначенные для аутентификации механизмы включают в себя пары задача-ответ, команды EXTERNAL и INTERNAL AUTHENTICATE исключают механизмы аутентификации, включающие в себя тройку свидетельство-задача-ответ (см. ИСО/МЭК 9798-5 [8]). При обмене тройкой требуется две или несколько пар команда-ответ GENERAL AUTHENTICATE: такие пары команда-ответ могут быть сцеплены (см. 5.1.1.1).

Функция (либо EXTERNAL, либо INTERNAL, либо MUTUAL AUTHENTICATE) может быть выполнена, только если состояние защиты удовлетворяет атрибутам секретности для этой операции. Любое успешное выполнение аутентификации может подчиняться успешному завершению предшествующих команд (например, команд VERIFY, SELECT) или выбору (например, соответствующего секрета). Результат (да или нет) проверки, выполняемой картой, может при определенных условиях обновить состояние защиты. Карта может регистрировать, сколько раз функция была вызвана для того, чтобы ограничить число дальнейших использований соответствующего секрета или алгоритма. Карта может регистрировать безуспешные операции аутентификации (например, с целью ограничения числа дальнейшего использования ссылочных данных).

Пара команда-ответ GENERAL AUTHENTICATE представлена в таблице 70.

Таблица 70 — Пара команда-ответ GENERAL AUTHENTICATE

CLA INS P1-P2	Как определено в 5.1.1 '86' или '87' См. 7.5.1 и таблицу 65
Поле L_c	Присутствует для кодирования $N_c > 0$
Поле данных	Данные, касающиеся аутентификации
Поле L_e	Отсутствует для кодирования $N_e = 0$, присутствует для кодирования $N_e > 0$

Поле данных	Отсутствует (либо вследствие отсутствия поля L_e, например, последняя команда функции EXTERNAL AUTHENTICATE, либо процесс прерван), или данные, касающиеся аутентификации
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6300' (см. 7.5.1), '63CX' (см. 7.5.1), '6581', '6700', '6982', '6983', '6984', '6A81', '6A82' '6A86', '6A88' (см. 7.5.1)

Если присутствует, то каждое поле данных должно содержать межотраслевой шаблон, обращение к которому происходит с помощью тега '7C'. В шаблоне динамической аутентификации контекстно-зависимый класс (первый байт от '80' до 'BF') зарезервирован для информационных объектов «динамическая аутентификация», как перечислено в таблице 71.

Таблица 71 — Информационные объекты «динамическая аутентификация»

Тег	Значение
'7B'	Набор информационных объектов «динамическая аутентификация» со следующими тегами
'80'	Свидетельство (например, одно или несколько положительных чисел меньше, чем используемые открытые модули)
'81'	Задача (например, одно или несколько чисел, возможно нулей, меньше, чем используемая открытая экспонента)
'82'	Ответ (например, одно или несколько положительных чисел меньше, чем используемые открытые модули)
'83'	Фиксированная задача (например, хэш-код или большие случайные числа, включающие одну или несколько задач)
'84'	Код аутентификации (например, хэш-код одного или нескольких полей данных и информационный объект «свидетельство»)
'85'	Экспоненциал (например, положительное число для установления сеансового ключа с помощью метода согласования ключа)
'A0'	Шаблон идентификационных данных
Примечание — В данном контексте ИСО/МЭК СТК 1 ПК 17 зарезервировал любой другой информационный объект контекстно-зависимого класса (первый байт от '80' до 'BF').	

В пределах межотраслевого шаблона для динамической аутентификации применяют следующие правила:

- если информационный объект пустой в шаблоне, то он должен быть заполнен в шаблоне следующего поля данных;

- в первом поле данных команды шаблон указывает функцию динамической аутентификации следующим образом:

- запрос свидетельства, например пустое свидетельство, обозначает функцию INTERNAL AUTHENTICATE;

- запрос задачи, например пустая задача, обозначает функцию EXTERNAL AUTHENTICATE;

- отсутствие пустого информационного объекта обозначает функцию MUTUAL AUTHENTICATE. Далее, пока карта не прервет процесс, шаблон в поле данных ответа должен содержать те же самые информационные объекты, что и шаблон в поле данных команды. Функция MUTUAL AUTHENTICATE позволяет двум объектам согласовать сессионные ключи, используя пару элементов данных «Экспоненциал», обращение к которым происходит с помощью тега '85' (см. методы согласования ключей по ИСО/МЭК 11770-3 [14]).

Динамическая аутентификация может защитить поля данных, обмениваемые во время сессии. Оба объекта поддерживают текущий хэш-код, обновленный при включении одного поля данных команды или ответа в одно время. Информационный объект с тегом '84' передает код аутентификации, который является результатом обновления текущего кода при включении информационного объекта «свидетельство» с тегом '80'. Устройство верификации последовательно воспроизводит свидетельство и код аутентификации: если реконструируемое свидетельство не является нулем и если два кода идентичны, то аутентификация считается успешной.

В приложении С проиллюстрированы пары команда-ответ GENERAL AUTHENTICATE для выполнения функций EXTERNAL, INTERNAL и MUTUAL AUTHENTICATE, с расширением на аутентификацию поля данных и согласование ключей.

7.5.6 Команда VERIFY

Команда инициирует сравнение в карте ссылочных данных, хранимых в карте, с данными верификации, посылаемыми с устройства сопряжения (например, паролем) или с датчика на карте (например, отпечаток пальца). В результате сравнения состояние защиты может изменяться. Безуспешные операции сравнения могут регистрироваться в карте (например, с целью ограничения числа дальнейших попыток использования ссылочных данных).

- Если INS = '20', то поле данных команды обычно присутствует для передачи данных верификации. Отсутствие поля данных команды используется для проверки, требуется ли верификация (SW1-SW2 = '63CX', где 'X' кодирует число дальнейших разрешенных попыток) или нет (SW1-SW2 = '9000').

- Если INS = '21', то поле данных команды должно передавать информационный объект «верификация» (например, тег '5F2E', см. ИСО/МЭК 7816-11 [4]), который обычно не пустой. Присутствие пустого информационного объекта «верификация» с расширенным списком заголовка (тег '4D', см. 8.5.1) означает, что данные верификации получены с помощью датчика в карте.

Пара команда-ответ VERIFY представлена в таблице 72.

Таблица 72 — Пара команда-ответ VERIFY

CLA INS P1 P2	Как определено в 5.1.1 '20','21' '00' (любое другое значение зарезервировано для использования в будущем) См. таблицу 65
Поле L_c	Отсутствует для кодирования $N_c = 0$, присутствует для кодирования $N_c > 0$
Поле данных	Данные верификации, или отсутствует (INS = '20'), или Информационный объект «верификация» и, при определенных условиях, расширенный список заголовков (INS = '21')
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6286', '6300' (см. 7.5.1), '63CX' (см. 7.5.1), '6581', '6700', '6982', '6983', '6984', '6A81', '6A82' '6A86', '6A88' (см. 7.5.1)

7.5.7 Команда CHANGE REFERENCE DATA

Команда либо заменяет ссылочные данные, хранимые в карте, новыми ссылочными данными, посылаемыми с устройства сопряжения, либо инициирует их сравнение с данными верификации, посылаемыми с устройства сопряжения и затем, при определенных условиях, заменяет их новыми ссылочными данными, посылаемыми с устройства сопряжения. Команда может выполняться только, если состояние защиты удовлетворяет атрибутам секретности этой команды.

Пара команда-ответ CHANGE REFERENCE DATA представлена в таблице 73.

Таблица 73 — Пара команда-ответ CHANGE REFERENCE DATA

CLA INS P1-P2	Как определено в 5.1.1 '24' '00' или '01' (любое другое значение зарезервировано для использования в будущем) См. таблицу 65
Поле L_c	Присутствует для кодирования $N_c > 0$
Поле данных	Данные верификации, за которыми следуют новые ссылочные данные без разграничения (P1 установлен на '00'), или Новые ссылочные данные (P1 установлен на '01')
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6300' (см. 7.5.1), '63CX' (см. 7.5.1), '6581', '6700', '6982', '6983', '6984', '6A81', '6A82' '6A86', '6A88' (см. 7.5.1)

7.5.8 Команда ENABLE VERIFICATION REQUIREMENT

Команда включает требование для сравнения ссылочных данных с данными верификации. Она может выполняться только, если состояние защиты удовлетворяет атрибутам секретности для этой команды.

Пара команда-ответ ENABLE VERIFICATION REQUIREMENT представлена в таблице 74.

Таблица 74 — Пара команда-ответ ENABLE VERIFICATION REQUIREMENT

CLA INS P1-P2	Как определено в 5.1.1 '28' '00' или '01' (любое другое значение зарезервировано для использования в будущем) См. таблицу 65
Поле L_c	Отсутствует для кодирования Nc = 0, присутствует для кодирования $N_c > 0$
Поле данных	Отсутствует (P1 установлен на '01'), или данные верификации (P1 установлен на '00')
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6300' (см. 7.5.1), '63CX' (см. 7.5.1), '6581', '6700', '6982', '6983', '6984', '6A81', '6A82' '6A86', '6A88' (см. 7.5.1)

7.5.9 Команда DISABLE VERIFICATION REQUIREMENT

Команда отключает требование для сравнения ссылочных данных с данными верификации, и, возможно, включает требование для сравнения других ссылочных данных с данными верификации. Она может выполняться только, если состояние защиты удовлетворяет атрибутам секретности для этой команды.

Пара команда-ответ DISABLE VERIFICATION REQUIREMENT представлена в таблице 75.

Таблица 75 — Пара команда-ответ DISABLE VERIFICATION REQUIREMENT

CLA INS P1 P2	Как определено в 5.1.1 '26' '00', '01' или 100xxxxx, где xxxxx — число ссылочных данных (любое другое значение зарезервировано для использования в будущем) См. таблицу 65
Поле L_c	Отсутствует для кодирования $N_c = 0$, присутствует для кодирования $N_c > 0$
Поле данных	Отсутствует (P1 установлен на '01'), или данные верификации (P1 установлен на '00' или 100x xxxx)
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6300' (см. 7.5.1), '63CX' (см. 7.5.1), '6581', '6700', '6982', '6983', '6984', '6A81', '6A82' '6A86', '6A88' (см. 7.5.1)

7.5.10 Команда RESET RETRY COUNTER

Команда либо сбрасывает счетчик повторений ссылочных данных в исходное значение, либо изменяет ссылочные данные по завершению сброса счетчика повторений ссылочных данных в исходное значение. Она может выполняться только, если состояние защиты удовлетворяет атрибутам секретности для этой команды.

Пара команда-ответ RESET RETRY COUNTER представлена в таблице 76.

Таблица 76 — Пара команда-ответ RESET RETRY COUNTER

CLA INS P1 P2	Как определено в 5.1.1 '26' '00', '01', '02' или '03'(любое другое значение зарезервировано для использования в будущем) См. таблицу 65
Поле L_c	Отсутствует для кодирования $N_c = 0$, присутствует для кодирования $N_c > 0$
Поле данных	Отсутствует (P1 установлен на '03'), или Код сброса, за которым следуют новые ссылочные данные без ограничений (P1 установлен на '00'), или Код сброса (P1 установлен на '01'), или Новые ссылочные данные (P1 установлен на '02')
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6300' (см. 7.5.1), '63CX' (см. 7.5.1), '6581', '6700', '6982', '6983', '6984', '6A81', '6A82' '6A86', '6A88' (см. 7.5.1)

7.5.11 Команда MANAGE SECURITY ENVIRONMENT

Команда подготавливает безопасный обмен сообщениями (см. 6) и команды по защите (например, EXTERNAL, INTERNAL и GENERAL AUTHENTICATE, см. также REFORM SECURITY OPERATION по ИСО/МЭК 7816-8 [4]). Команда поддерживает следующие функции:

- SET, т. е. установление или замена одного компонента в текущей SE;
- STORE, т. е. сохранение текущей SE под байтом SEID в P1;
- RESTORE, т. е. замена текущей SE на SE, хранимую в карте и идентифицируемую байтом SEID в P2;
- ERASE, т. е. стирание SE, хранимой в карте и идентифицируемой байтом SEID в P2.

Пара команда-ответ MANAGE SECURITY ENVIRONMENT и кодирование P1 и P2 представлены в таблицах 77, 78 и 79.

Таблица 77 — Пара команда-ответ MANAGE SECURITY ENVIRONMENT

CLA INS P1 P2	Как определено в 5.1.1 '22' См. таблицу 78 См. таблицу 79
Поле L_c	Отсутствует для кодирования $N_c = 0$, присутствует для кодирования $N_c > 0$
Поле данных	Отсутствует (STORE, RESTORE и ERASE), или сцепление информационных объектов «управляющие ссылки» (SET)
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6600', '6987', '6988', '6A88' (см. 7.5.1)

Таблица 78 — P1

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
—	—	—	1	—	—	—	—	Безопасный обмен сообщениями в поле данных команды
—	—	1	—	—	—	—	—	Безопасный обмен сообщениями в поле данных ответа
—	1	—	—	—	—	—	—	Вычисление, дешифровывание, внутренняя аутентификация и ключ согласования
1	—	—	—	—	—	—	—	Верификация, шифрование, внешняя аутентификация и ключ согласования
				0	0	0	1	SET
1	1	1	1	0	0	1	0	STORE
1	1	1	1	0	0	1	1	RESTORE
1	1	1	1	0	1	0	0	ERASE

Примечание — Любое другое значение зарезервировано для использования в будущем ИСО/МЭК СТК1 ПК 17.

Таблица 79 — P2

Значение	Смысловое содержание
'XX'	'Байт SEID в случаях STORE, RESTORE и ERASE (установлен на '00' в случае GET SE)
	Тег шаблона управляющих ссылок присутствует в поле данных команды в случаях SET, или GET CRT
'A4'	- Шаблон управляющих ссылок аутентификации (AT)
'A6'	- Шаблон управляющих ссылок согласования ключей (KAT)
'AA'	- Шаблон управляющих ссылок хэш-кода (HT)
'B4'	- Шаблон управляющих ссылок криптографической контрольной суммы (CCT)
'B6'	- Шаблон управляющих ссылок цифровой подписи (DST)
'B8'	- Шаблон управляющих ссылок конфиденциальности (CT)

Примечание — Любое другое значение зарезервировано для использования в будущем ИСО/МЭК СТК1 ПК 17.

Функция KEY DERIVATION. Применение концепции главных ключей требует установления ключа в карте, содержащей главный ключ. В таблице 80 показано применение команды MANAGE SECURITY ENVIRONMENT для установления ключа. Предполагается, что главный ключ и алгоритм неявно выбраны в карте (в противном случае команда MANAGE SECURITY ENVIRONMENT может дополнительно выбрать ключ и алгоритм).

Примечание — В зависимости от ссылки на алгоритм данные для передачи ключа из главного ключа могут быть частью входных данных последующей команды (например, EXTERNAL AUTHENTICATE). В этом случае использование команды MANAGE SECURITY ENVIRONMENT для установления ключа не обязательно.

Таблица 80 — Пара команда-ответ для функции KEY DERIVATION

CLA INS P1 P2	Как определено в 5.1.1 '22' 'X'(SET, см. таблицу 78) Тег CRT (например, 'A4' если следует команда EXTERNAL AUTHENTICATE, или 'B4', если следует команда VERIFY CRYPTOGRAPHIC CHECKSUM)
Поле L_c	Присутствует для кодирования $N_c > 0$
Поле данных	{'94' — L — Данные для передачи ключа (обязательно)}; информационные объекты SM могут присутствовать
Поле L_e	Отсутствует для кодирования $N_e = 0$

Поле данных	Отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6600', '6987', '6988', '6A88' (ссылочные данные не найдены)

7.6 Обработка передачи

7.6.1 Команда GET RESPONSE

Команда используется для передачи ответа APDU (или его части), который по-другому не может быть передан при помощи доступных протоколов. См. примеры в ИСО/МЭК 7816-3.

Если поле L_e содержит только установленные на '00' байты, то все доступные байты должны быть выданы в пределах максимума 256 для короткого поля L_e или 65536 для расширенного поля L_e.

Пара команда-ответ для функции GET RESPONSE представлена в таблице 81.

Таблица 81 — Пара команда-ответ для функции GET RESPONSE

CLA INS P1-P2	Как определено в 5.1.1 'C0' '0000'(Любое другое значение зарезервировано для использования в будущем)
Поле L_c	Отсутствует для кодирования $N_c = 0$
Поле данных	Отсутствует
Поле L_e	Присутствует для кодирования $N_e > 0$

Поле данных	Отсутствует при любой ошибке, или ответ APDU [или его части] в соответствии с N_e
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '61XX' ('XX' кодирует число дополнительных имеющихся байтов данных, доступных для последующей команды GET RESPONSE) '6281', '6700', '6A81', '6A82', '6A86', '6CXX'

7.6.2 Команда ENVELOPE

Команда используется либо для передачи APDU [или его части], либо информационного объекта BER-TLV, который по-другому не может быть передан при помощи доступных протоколов. См. примеры в ИСО/МЭК 7816-3.

Примечание — Использование команды ENVELOPE для безопасного обмена сообщениями представлено в приложении В.

Пара команда-ответ для функции ENVELOPE представлена в таблице 82.

Таблица 82 — Пара команда-ответ для функции ENVELOPE

CLA INS P1-P2	Как определено в 5.1.1 'C2', 'C3' '0000'(Любое другое значение зарезервировано для использования в будущем)
Поле L_c	Присутствует для кодирования $N_c > 0$
Поле данных	Командный APDU [или его часть] (INS = 'C2'), или информационный объект BER-TLV [или его часть] (INS = 'C3')
Поле L_e	Отсутствует для кодирования $N_e = 0$, присутствует для кодирования $N_e > 0$

Поле данных	Ответный APDU [или его часть] (INS = 'C2'), или отсутствует
SW1-SW2	См. таблицы 5 и 6, где соответствие, например, '6700'

8 Услуги карты, не зависимые от приложения

В настоящем разделе приводится описание услуг, предоставляемых картой, не зависимых от приложений и именуемых далее в тексте как «услуги карты» по:

1) идентификации карты;
2) выбору и идентификации приложения;
3) выбору по пути;
4) извлечению данных;
5) извлечению элемента данных;
6) строкам байтов, образованных картой.

Их назначение заключается в обеспечении механизмов обмена между картой и устройством сопряжения, не имеющих никакой информации друг о друге, за исключением той, что они оба удовлетворяют требованиям настоящего стандарта. Услуги карты являются результатом любой комбинации байтов предыстории (см. 8.1.1), содержимого EF.DIR и EF.ATR (см. 8.2.1.1) и последовательностей команд. Если не указано иное, каждое командное APDU использует CLA, установленный на '00', т. е. использует основной логический канал и не использует сцепление команд и безопасный обмен сообщениями.

Приложение не обязательно должно соответствовать положениям настоящего раздела, как только оно будет идентифицировано и выбрано в карте. Для осуществления подобных функций оно может использовать другие механизмы, совместимые с настоящим стандартом. Поэтому такие решения могут и не гарантировать обмен.

8.1 Идентификация карты

Данная услуга позволяет устройству сопряжения идентифицировать карту, а также выяснять, как с ней взаимодействовать. Байты предыстории (см. 8.1.1) обеспечивают основную поддержку для идентификации карты. Карта предоставляет информацию внешнему устройству о своем логическом содержании напрямую, например, через байт данных «услуги карты» (см. 8.1.1.2.3), и/или неявно, например, с помощью исходных данных доступа (см. 8.1.1.2.4), указывающих доступ к файлу, выбираемому неявным образом сразу после ответа на восстановление и, возможно, выбора протокола и параметров. Поэтому данные, доступные на этом этапе, т. е. строка исходных данных (см. 8.1.2), могут быть впоследствии неизвлекаемыми.

8.1.1 Байты предыстории

8.1.1.1 Назначение и общая структура

Байты предыстории указывают рабочие характеристики для карт.

- Если карта отвечает на восстановление, то Ответ-на-Восстановление может содержать байты предыстории (см. ИСО/МЭК 7816-3).

- Если физический интерфейс не позволяет карте ответить на восстановление, например, универсальная последовательная шина или доступ с использованием радиочастотной связи, то команда GET DATA (7.4.2) может извлечь байты предыстории (тег '5F52').

Первым байтом предыстории является «байт индикатора категории». Если байт индикатора категории установлен на '00', '10' или '8X', то для этого случая в таблице 83 проиллюстрирован формат байтов предыстории. Любое другое значение указывает на проприетарный формат

Таблица 83 — Байт индикатора категории

Значение	Смысловое содержание
'00'	Индикатор состояние должен быть представлен в виде последних трех байт предыстории (см. 8.1.1.3)
'10'	См. 8.1.1.4
'80'	Индикатор состояние может присутствовать в информационном объекте COMPACT-TLV (один, два или три байта, см. 8.1.1.3)
От '81' до '8F'	Зарезервировано для использования в будущем
Примечание — Любое другое значение указывает на проприетарный формат.	

- Если первый байт предыстории установлен на '00', то остальные байты предыстории состоят из необязательных последовательных информационных объектов COMPACT-TLV с последующим обязательным индикатором состояния (последние три байта, не в TLV).

- Если первый байт предыстории установлен на '80', то остальные байты предыстории состоят из необязательных последовательных информационных объектов COMPACT-TLV; последний информационный объект может передавать индикатор состояния из одного, двух или трех байтов.

Любой межотраслевой информационный объект BER-TLV, состоящий из поля тега, установленного на '4X', и поля длины, установленного на '0Y', и поля значения из Y байт может передаваться в информационный объект COMPACT-TLV, состоящий из байта, установленного на 'XY' и называемого «сжатый заголовок» и поля значения из Y байтов.

Любой межотраслевой элемент данных, определяемый далее, может быть представлен в EF.ATR (см. 8.2.1.1). Если он имеется в EF.ATR, то он должен находиться в информационном объекте BER-TLV, т. е. в поле тега, установленного на '4X', в поле длины, установленном на '0Y' и в поле значения из Y байтов.

8.1.1.2 Необязательные элементы данных

8.1.1.2.1 Индикатор страны или эмитента

Данный межотраслевой элемент, связанный с сжатым заголовком, установленным либо на '1Y', либо на '2Y', является индикатором страны или эмитента (см. также теги '41' и '42' в таблице 9). В таблице 84 показан индикатор страны или эмитента.

Таблица 84 — Индикатор страны или эмитента

Сжатый заголовок	Значение
'1Y'	Код страны (см. ИСО 3166-1 [1]) и дополнительные национальные данные
'2Y'	Идентификационный номер эмитента (см. ИСО/МЭК 7812-1 [3]) и дополнительные данные эмитента

- Индикатор страны состоит из кода страны (три четырехразрядных байта со значениями от '0' до '9', см. ИСО 3166-1 [1]), за которым следуют новые данные (как минимум, один четырехразрядный байт). Соответствующий национальный орган по стандартизации должен выбрать эти последующие данные (нечетное число четырехразрядного байта).

- Индикатор эмитента состоит из идентификационного номера эмитента (см. ИСО/МЭК 7812-1[3]), за которым возможно следуют новые данные. Эмитент карты должен выбрать эти новые байты, при их наличии (для кодирования, например, основного учетного номера)

Примечание — В ИСО/МЭК 7812-1:1993[1)] идентификационный номер может состоять из нечетного числа четырехразрядных байтов со значением от '0' до '9'. В этом случае он преобразуется в строку бит путем установки бит с 4 по 1 последнего байта на 1111.

8.1.1.2.2 Индикатор приложения

Данный межотраслевой элемент данных, связанный с сжатым заголовком, установленным на 'FY', представляет собой идентификатор приложения (AID, см. 8.2.1.2, см. также тег '4F' в таблице 9). Если AID присутствует в байтах предыстории или строке исходных данных (см. 8.1.2), то он обозначает неявно выбранное приложение (см. 8.2.2.1).

8.1.1.2.3 Данные об услугах, предоставляемых картой

Данный межотраслевой элемент данных, связанный с сжатым заголовком, установленным на '31', указывает методы, доступные в карте для поддержания услуг, описанных в разделе 9. В таблице 85 показаны байты данных «услуги карты».

[1)] ИСО/МЭК 7812-1:1993 отменен и заменен на ИСО/МЭК 7812-1:2006.

Таблица 85 — Байты данных «услуги карты»

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
x	x	—	—	—	—	—	—	**Выбор приложения:**
1	—	—	—	—	—	—	—	- по полному имени DF
—	1	—	—	—	—	—	—	- по частичному имени DF
—	—	x	x	—	—	—	—	**Информационные объекты BER-TLV** доступны:
—	—	1	—	—	—	—	—	- в EF.DIR (см. 8.2.1.1)
—	—	—	1	—	—	—	—	- в EF.ATR (см. 8.2.1.1)
—	—	—	—	x	x	x	—	**Услуги доступа EF.DIR и EF.ATR посредством**
—	—	—	—	1	0	0	—	- команды READ BINARY (прозрачная структура)
—	—	—	—	0	0	0	—	- команды READ RECORD(S) (структура записи)
—	—	—	—	0	1	0	—	- команды GET DATA (структура TLV)
—	—	—	—	Любое другое значение			—	Зарезервировано для использования в будущем
—	—	—	—	—	—	—	0	**Карты с MF**
—	—	—	—	—	—	—	1	**Карты без MF**

Если байт данных «услуги карты» присутствует в байтах предыстории или в строке исходных данных (см. 8.1.2), то он указывает, присутствуют или нет EF.DIR или EF.ATR (см. 8.2.1.1) и как получить к ним доступ. Отсутствие байта данных «услуги карты» в байтах предыстории и строке исходных данных указывает на то, что карта поддерживает только неявный выбор приложения (значение по умолчанию).

8.1.1.2.4 Исходные данные доступа

Данный межотраслевой элемент данных, связанный с сжатым заголовком, установленным на '4Y', указывает командный APDU, который, как предполагается, должен быть первой командой после ответа на восстановление и возможно выбора протокола и параметров. Командный APDU определен в 8.1.2.

8.1.1.2.5 Данные эмитента карты

Данный межотраслевой элемент данных, связанный с сжатым заголовком, установленным на '5Y', не определен в стандартах серии ИСО/МЭК 7816. Эмитент карты определяет длину, структуру и кодирование.

8.1.1.2.6 Данные, предваряющие эмиссию карты

Данный межотраслевой элемент данных, связанный с сжатым заголовком, установленным на '6Y', не определен в стандартах серии ИСО/МЭК 7816. Изготовитель карт определяет длину, структуру и кодирование для изготовителя карт, наименование интегральной схемы, изготовителя интегральной схемы, версию маски ROM, версию операционной системы и т. д. Этот межотраслевой элемент данных может содержать идентификатор изготовителя интегральной схемы (см. ИСО/МЭК 7816-6).

8.1.1.2.7 Функциональные возможности карты

Данный межотраслевой элемент данных, связанный со сжатым заголовком, установленным на '71', '72' или '73', состоит из не более трех таблиц программных функций: либо из первой таблица, либо из первых двух таблиц, либо из трех таблиц.

- Первая таблица программных функций указывает методы выбора, поддерживаемые картой.

- Вторая таблица программных функций — это «байт кодирования данных». Байт кодирования данных также может быть представлен как второй байт в контрольном параметре файла, обращение к которому происходит с помощью тега '82' (см. таблицу 12).

- Третья таблица программных функций указывает на возможность объединять в цепочку команды, обрабатывать расширенные поля L_c и L_e и управлять логическими каналами.

Таблица 86 — Первая таблица программных функций (методы выбора)

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
x	x	—	—	—	—	—	—	**Выбор DF** (см. 5.3.1):
1	—	—	—	—	—	—	—	- по полному имени DF
—	1	—	—	—	—	—	—	- по частичному имени DF
—	—	1	—	—	—	—	—	- через путь
—	—	—	1	—	—	—	—	- по идентификатору файла
—	—	—	—	1	—	—	—	Неявный выбор DF
—	—	—	—	—	1	—	—	Поддерживается короткий идентификатор EF
—	—	—	—	—	—	1	—	Поддерживается номер записи
—	—	—	—	—	—	—	1	Поддерживается идентификатор записи

Таблица 87 — Вторая таблица программных функций (байт кодирования данных)

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
	—	—	—	—	—	—	—	Поддерживается структура TLV фалов EF
1	x	x	—	—	—	—	—	**Режим работы функций записи:**
—	0	0	—	—	—	—	—	- однократная запись
—	0	1	—	—	—	—	—	- проприетарная запись
—	1	0	—	—	—	—	—	- запись с использованием операции OR
—	1	1	—	—	—	—	—	- запись с использованием операции AND
—	—	—	—	x	x	x	x	**Размер единицы данных в четырёхразрядный байтах** (от одного до 32768 четырехразрядный байтов, т. е. 16384 байтов) (степень числа 2, например, 0001 = 2 четырехразрядный байта = один байт, значение по умолчанию)
—	—	—	x	—	—	—	—	**Значение 'FF' для первого байта полей тега BER-TLV (см. 5.2.2.1):**
—	—	—	0	—	—	—	—	- Недействительно (используется для заполнения, значение по умолчанию);
—	—	—	1	—	—	—	—	- Действительно (длинные приватные теги, составное кодирование)

Таблица 88 — Третья таблица программных функций (сцепление команд, поля длины и логические каналы)

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
1	—	—	—	—	—	—	—	**Сцепление команд** (см. 5.1.1.1)
—	—	—	—	—	—	—	—	**Расширенные поля L_c и L_e** (см. 5.1)
—	—	—	x	x	—	—	—	**Назначение количества логических каналов осуществляет:**
—	—	—	1	—	—	—	—	- карта
—	—	—	—	1	—	—	—	- устройство сопряжения
—	—	—	0	0	—	—	—	Нет логического канала
—	—	—	—	—	y	z	t	**Максимальное число логических каналов** (см. 5.1.1 и 5.1.1.2): - Если y, z и t не все установлены на 1, то это означает 4y + 2z + t + 1, т. е. максимальное число логических каналов от одного до семи; - Если y = z = t = 1, то это означает, что максимальное число логических каналов равно восьми или более.
—	—	x	—	—	—	—	—	RFU

8.1.1.3 Индикатор состояния

Если байт индикатора категории установлен на '00', то последние три байта предыстории должны быть индикатором состояния, а именно: байт состояния жизненного цикла карты, обозначенный LCS, с последующими двумя байтами состояния обработки, обозначенными SW1-SW2.

Если байт индикатора категории установлен на '80', то межотраслевой элемент данных, обращение к которому происходит с помощью сжатого заголовка, установленного на '81', '82' или '83', может присутствовать как индикатор состояния на один, два или три байта (другие значения длины зарезервированы для ИСО/МЭК СТК1 ПК 17) в конце байтов предыстории.

- Если длина равна единице, то элемент данных представляет собой байт состояния жизненного цикла карты, обозначаемый LCS;

- Если длина равна двум, то элемент данных представляет собой два байта состояния обработки, обозначаемые SW1-SW2;

- Если длина равна трем, то элемент данных представляет собой LCS с последующими SW1-SW2.

LCS должен интерпретироваться в соответствии с 5.3.3.2 и таблицей 13; значение '00' указывает, что состояние не передано. SW1-SW2 должны интерпретироваться в соответствии с 5.1.3 и таблицами 5 и 6; значение '0000' состояние не передано.

8.1.1.4Ссылка на данные DIR

Если байт индикатора категории установлен на '10', то следующий за ним байт представляет собой ссылку на данные DIR. Кодирование и содержание этого байта находятся за пределами компетенции настоящего стандарта.

8.1.2 Восстановление строки исходных данных

Межотраслевой элемент данных, называемый «исходные данные доступа», обращение к которому происходит с помощью сжатого заголовка, установленного на '4Y', в байтах предыстории (см. 8.1.1.2.4) или с помощью тега '44' в EF.ATR (см. 8.2.1.1), указывает командный APDU.

- Если длина равна единице, то командный APDU представляет собой команду READ BINARY (см. 7.2.3) со следующим содержанием: CLA INS P1 P2 установлены на '00B0 0000' и поле L_e установлено на первый и единственный байт исходных данных доступа;

- Если длина равна двум, то первый байт исходных данных доступа указывает структуру (бит 8) и короткий идентификатор (биты с 5 по 1) файла EF, подлежащего считыванию, в соответствии с таблицей 89:

- если бит 8 первого байта установлен на 1, то командный APDU представляет собой команду READ BINARY (см. 7.2.3) со следующим содержанием: CLA INS установлены на '00B0', P1 установлен на первый байт исходных данных доступа, P2 установлен на '00', а поле L_e установлено на второй байт исходных данных доступа;

- если бит 8 первого байта установлен на 0, то командный APDU представляет собой команду READ RECORD(S) (см. 7.3.3) со следующим содержанием: CLA INS P1 установлены на '00B201', P2 состоит из бит с 8 по 4, установленных на биты с 5 по 1 первого байта исходных данных доступа (короткий идентификатор) и бит с 3 по 1, установленных на 110, а поле L_e установлено на второй байт исходных данных доступа.

Таблица 89 — Первый байт исходных данных доступа при длине, равной двум

b8	b7	b6	b5	b4	b3	b2	b1	Смысловое содержание
x	—	—	—	—	—	—	—	**Структура EF**
0	—	—	—	—	—	—	—	Структура записи
1	—	—	—	—	—	—	—	Прозрачная структура
—	—	—	x	x	x	x	x	Короткий идентификатор EF
—	x	x	—	—	—	—	—	x00x xxxx (любое другое значение зарезервировано для использования в будущем)

- Если длина равна пяти или более, то командный APDU состоит из Y байтов исходных данных доступа.

Командный APDU должен быть представлен карте. Если процесс завершен, то поле данных ответа представляет собой строку межотраслевых информационных объектов, в которых каждое приложение могло бы быть привлечено, называемую «строка исходных данных».

8.2 Идентификация и выбор приложения

Данная услуга позволяет устройству сопряжения знать, какое приложение активировано в карте, а также как идентифицировать и выбрать какое-либо приложение, поддерживаемое картой.

8.2.1 Идентификация приложения

8.2.1.1 EF.DIR и EF.ATR

Два специфичных файла EF обеспечивают основную поддержку идентификации и выбора приложения, а именно: EF.DIR и EF.ATR. Они содержат набор информационных объектов BER-TLV. В этих EF стертые или измененные информационные объекты BER-TLV могут вызывать заполнение незначащей информацией до, между или после информационных объектов (см. 5.2.2.1).

EF.DIR. Данный EF указывает список приложений, поддерживаемых картой. Он содержит набор шаблонов приложений и/или информационных объектов «идентификатор приложения» в произвольном порядке. Он определяет, какие команды должны выполняться для того, чтобы выбрать указанные приложения.

EF.DIR долен иметь MF в качестве родительского файла: путь '3F002F00' ссылается на EF.DIR. На уровне MF короткий идентификатор 30, т. е. 11110 в двоичном кодировании, ссылается на EF.DIR, если он присутствует.

EF.ATR. Данный EF указывает рабочие характеристики карты. Он содержит набор межотраслевых информационных объектов, которые не могут быть вложены в EF.DIR, либо потому что не относятся к выбору приложения, либо потому что нет EF.DIR.

EF.ATR должен иметь MF в качестве родительского файла: путь '3F002F01' ссылается на EF.ATR.

8.2.1.2 Идентификатор приложения

Данный межотраслевой элемент данных, обращение к которому происходит с помощью сжатого заголовка, установленного на 'FY' в байтах предыстории (см. 8.1.1), или с помощью тега '4F' в строке исходных данных (см. 8.1.2), в EF.ATR, в EF.DIR и в данных управления любым файлом DF (см. 5.3.3), идентифицирует приложение.

Идентификатор приложения (AID) состоит из не более шестнадцати байтов. Биты с 8 по 5 первого байта указывают категорию в соответствии с таблицей 90.

Таблица 90 — Категории идентификаторов приложения

Значение	Категория	Смысловое содержание
От '0' до '9'	—	Зарезервировано для совместимости с предыдущими версиями ИСО/МЭК 7812-1 [3] (см. приложение D)
'A'	Международная	Международная регистрация провайдеров приложений в соответствии с ИСО/МЭК 7816-5 [4]
'B', 'C'	—	Зарезервировано для использования в будущем ИСО/МЭК СТК 1 ПК 17
'D'	Национальная	Национальная регистрация (ИСО 3166 [1]) провайдеров приложений в соответствии с ИСО/МЭК 7816-5 [4]
'E'	Стандартная	Идентификация стандарта с помощью идентификатора объекта в соответствии с ИСО/МЭК 8825-1
'F'	Проприетарная	Регистрация провайдеров приложения не предусмотрена

На рисунке 7 показан международный AID. Он состоит из зарегистрированного идентификатора провайдера приложения (международный RID) из пяти байтов и, опционально, проприетарного расширения идентификатора приложения (PIX) из не более одиннадцати байтов.

- Международный RID должен однозначно идентифицировать провайдер приложения (см. ИСО/МЭК 7816-5 [4]):
- биты с 8 по 5 первого байта должны быть установлены на 1010, т. е. первый четырехразрядный байт должен быть установлен на 'A';
- каждый из последующих девяти четырехразрядных байтов должен быть установлен на значение от '0' до '9'.
- Расширение имеет свободное кодирование. Оно позволяет провайдеру приложения идентифицировать различные приложения.

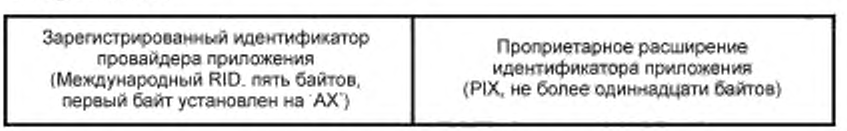


Рисунок 7 — Международный AID

На рисунке 8 показан национальный AID. Он состоит из зарегистрированного идентификатора провайдера приложения (национального RID) из пяти байтов и, опционально, проприетарного расширения идентификатора приложения (PIX) из не более одиннадцати байтов.

- Национальный RID должен однозначно идентифицировать провайдера приложения (см. ИСО/МЭК 7816-5 [4]):
- биты с 8 по 5 первого байта должны быть установлены на 1101, т. е. первый четырехразрядный байт должен быть установлен на 'D';
- последующие три четырехразрядные байта (от '0' до '9') должны формировать код страны (ИСО 3166-1 [1]);

- рекомендованное значение для каждого из последних шести четырехразрядных байтов — от '0' до '9'.

- Расширение имеет свободное кодирование. Оно позволяет провайдеру приложения идентифицировать различные приложения.

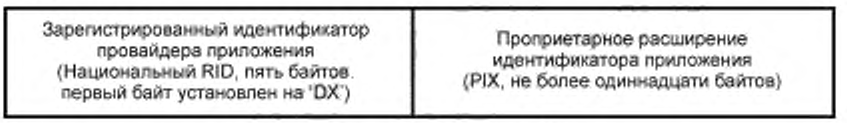


Рисунок 8 — Национальный AID

На рисунке 9 показан стандартный AID. Он состоит из не более шестнадцати байтов. Первый байт должен быть установлен на 1110 1000, т. е. на 'E8'. Значения от 'E0' до 'E7' и от 'E9' до 'EF' зарезервированы для использования в будущем ИСО/МЭК СТК 1 ПК 17. Идентификатор объекта (см. ИСО/МЭК 8825-1) должен поддерживать идентификацию стандарта, определяющего приложение (см. примеры в приложении А, например, ИСО/МЭК 7816-12[4], верификация личности биометрическими методами, ИСО/МЭК 7816-15 [4], приложение с криптографической информацией). Расширение идентификатора приложения (определенное в соответствии с идентифицированным стандартом) может поддерживать идентификацию различных реализаций.

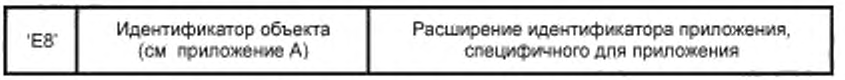


Рисунок 9 — Стандартный AID

На рисунке 10 показан проприетарный AID. Он состоит из не более шестнадцати байтов. Биты с 8 по 5 первого байта должны быть установлены на 1111, т.е. 'F'. В проприетарной категории, поскольку провайдеры приложения не зарегистрированы, то различные провайдера приложения могут использовать один и тот же AID.

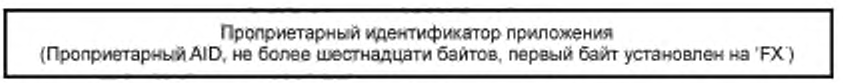


Рисунок 10 — Проприетарный AID

8.2.1.3 Шаблон приложения

Данный межотраслевой элемент данных, обращение к которому происходит с помощью тега '61', может присутствовать в EF.ATR (см. 8.2.1.1), в EF.DIR (см. 8.2.1.1) и в данных управления любым файлом DF (см. 5.3.3):

- такой шаблон должен содержать в себе один и только один идентификатор приложения. Если несколько идентификаторов приложений представляют собой действительные наименования для одного и того же DF, то каждое из них должно быть представлено в различных шаблонах приложения;

- такой шаблон может дополнительно содержать другие межотраслевые информационные объекты, касающиеся приложений, перечисленные в таблице 91 и определенные далее.

Таблица 91 — Межотраслевые информационные объекты для идентификации и выбора приложения

Тег	Значение
'4F'	Идентификатор приложения
'50'	Уровень приложения
'51'	Ссылка на файл
'52'	Командный APDU
'53', '73'	Произвольные данные, произвольный шаблон
5F50''	Унифицированный указатель ресурса (см. IETF RFC 1738[19] и IETF RFC 2396[20])
'61'	Набор информационных объектов, связанных с приложением

8.2.1.4 Другие межотраслевые элементы данных

Следующие межотраслевые элементы данных обеспечивают общую поддержку для идентификации и выбора приложения.

Уровень приложения. Данный межотраслевой элемент, обращение к которому происходит с помощью тега '50', не определен в серии стандартов ИСО/МЭК 7816. Провайдер приложения определяет его для использования в интерфейсе человек-машина, например, товарный знак для отображения.

Ссылка на файл. Данный межотраслевой элемент, обращение к которому происходит с помощью тега '51', определен в 5.3.1.2.

Произвольные данные (или шаблон). Данный межотраслевой элемент, обращение к которому происходит с помощью тега '53' (или '73'), состоит из соответствующих элементов данных (или вложений информационных объектов), определенных провайдером приложения.

Унифицированный указатель ресурса. Данный межотраслевой элемент данных, обращение к которому происходит с помощью тега '5F50', является унифицированным указателем ресурса (URL) по IETF RFC 1738 [19] и IETF RFC 2396 [20]. Он указывает часть программного обеспечения, необходимого устройству сопряжения для взаимодействия с приложением в карте.

8.2.2 Выбор приложения

Карта должна поддерживать, как минимум, один из следующих методов выбора приложения:

1) неявный выбор приложения;

2) выбор приложения с использованием идентификатора приложения (AID, см. 8.2.1.2) в качестве имени DF;

3) выбор приложения с использованием EF.DIR или EF.ATR.

8.2.2.1 Неявный выбор приложения

Если выбор приложения осуществляется неявно, то идентификатор приложения должен присутствовать в байтах предыстории (см. 8.1.1) или в строке исходных данных (см. 8.1.2). Такое наличие обозначает неявный выбор приложения. Если выбор приложения осуществляется неявно без идентификатора приложения в байтах предыстории и строке исходных данных, то идентификатор приложения должен присутствовать в EF.ATR (см. 8.2.1.1).

Примечание — Неявный выбор приложения не рекомендован для мульти-прикладных карт.

8.2.2.2 Выбор приложения с использованием AID в качестве имени DF

Карта в мульти-прикладной среде должна быть способна положительно реагировать на команду SELECT, определяющую идентификатор приложения (AID, см. 8.2.1.2) в качестве имени DF. Устройство сопряжения может таким образом неявно выбрать приложение без предварительной проверки наличия приложения в карте.

Карта должна поддерживать команду SELECT с CLA INS P1 P2, установленными на '00A4 0400', для первого выбора с заданным и предпочтительно полным идентификатором приложения в поле данных команды (см. таблицу 39). В зависимости от того, присутствует ли приложение или нет, карта должна завершить, либо прервать команду. В случае выбора с усеченным именем DF, полное имя DF будет доступно в поле данных ответа в качестве контрольного параметра файла, связанного с тегом '84' (см. таблицу 12). Если карта поддерживает выбор с усеченным именем DF, то первый выбор зависит от реализации, например, первое вхождение в статичный список или последнее активированное приложение в предыдущей сессии. Для следующего выбора, если он имеется, карта должна поддерживать команду SELECT с CLA INS P1 P2, установленными на '00A4 0402', с тем же полем данных команды.

8.2.2.3 Выбор приложения с использованием EF.DIR или EF.ATR

Для мульти-прикладного устройства сопряжения использование EF.DIR или EF.ATR более эффективно, чем предыдущий метод:

- если информационный объект «идентификатор приложения» не является частью шаблона приложения и не сопровождается ссылкой на файл или информационным объектом «команда на выполнение», то при выборе необходимо использовать AID в качестве имени DF;

- если информационный объект «идентификатор приложения» является частью шаблона приложения вместе с информационным объектом «ссылка на файл» (см. 5.3.1.2), его поле значения состоит из двух или более байтов, то выбор через путь должен выполняться в соответствии с 8.3;

- если информационный объект «идентификатор приложения» является частью шаблона приложения вместе с одним или несколькими информационными объектами «команды на выполнение», то выбор приложения выполняется с помощью перечисленных(ой) команд(ы). В случае нескольких команд, они должны быть выполнены в порядке, представленном в шаблоне.

8.3 Выбор через путь

Данная услуга позволяет сделать выбор файлов EF и безымянных DF, используя путь, т. е. элемент данных «ссылка на файл» (см. 5.3.1.2), состоящий из трех или более байтов.

- Если длина четная, то путь является либо абсолютным, либо относительным в зависимости от того, установлены ли первые два байта на '3F00' или нет. Последние два байта идентифицируют либо DF, либо EF:

- для пути к DF выбор должен быть осуществлен с использованием одной или нескольких команд SELECT с CLA INS P1 P2 L_c, установленными на '00A4 0100 02';

- для пути к EF, если длина равна четырем или более, то выбор должен быть осуществлен с использованием одной или нескольких команд SELECT с CLA INS P1 P2 L_c, установленными на '00A4 0100 02'. Последний и возможно единственный выбор использует два байта пути (идентификатор EF) с CLA INS P1 P2 L_c, установленными на '00A4 0200 02'.

- Если длина нечетная, то путь ограниченный. Он состоит либо из абсолютного пути без '3F00', либо из относительного пути без идентификатора текущего DF, за которым следует байт, подлежащий использованию в качестве P1 в одной или нескольких командах SELECT. Значение P1 фиксирует метод выбора:

- если значение P1 — '08' или '09', то карта должна поддерживать команду SELECT, в которой ограниченный путь определяет P1, L_c и поле данных и в которой P2 установлен на '00';

- в остальных случаях карта должна поддерживать одну или несколько команд SELECT с P1, установленным на последний байт ограниченного пути, и P2 L_c, установленными на '0002'. Каждый файл на всем протяжении пути должен выбираться последовательно.

8.4 Извлечение данных

Данная услуга позволяет устройству сопряжения считывать данные, хранимые в файлах DF и EF.

Если был выбран DF, то содержание, имеющее отношение к обмену, должно представлять собой поле данных ответа для команды GET DATA (см. 7.4.2), состоящее из CLA INS, установленных на '00CA', за которыми следуют байты P1-P2, установленные на '00FF' для информационных объектов BER-TLV или на '02FF' для информационных объектов SIMPLE-TLV, с последующим полем L_e, содержащим только установленные на '00' байты.

Если был выбран EF, то содержание, имеющее отношение к обмену, должно представлять собой поле данных ответа для команды READ в соответствии с байтом описателя файла (см. таблицу 14), при его наличии в контрольных параметрах.

- Команда READ BINARY (см. 7.2.3) состоит из CLA INS P1 P2, установленных на '00B0 0000', с последующим полем L_e, состоящим только из установленных на '00' байтов.

- Команда READ RECORD(S) (см. 7.3.3) состоит из CLA INS P1 P2, установленных на '00B2 0005', с последующим полем L_e, состоящим только из установленных на '00' байтов.

- Команда GET DATA (см. 7.4.2) состоит из CLA INS P1 P2, установленных на '00CA 0000', с последующим полем L_e, состоящим только из установленных на '00' байтов.

При отсутствии байта описателя файла в контрольных параметрах файла EF командный APDU представляет собой следующее:

- если первая таблица программных функций (см. таблицу 86) присутствует в байтах предыстории или в EF.ATR и если она указывает на поддержку записей, то командный APDU представляет собой команду READ RECORD(S), как указано выше;

- в противном случае, т. е., когда таблица отсутствует в байтах предыстории и в EF.ATR или если таблица не указывает на поддержку записей, то командный APDU представляет собой команду READ BINARY, как указано выше.

8.5 Извлечение элемента данных

Данная услуга позволяет устройству сопряжения извлечь межотраслевые элементы данных, используемые при обмене.

До выбора приложения межотраслевые информационные объекты должны быть извлечены прямо или косвенно из байтов предыстории (см. 8.1.1), из строки исходных данных (см. 8.1.2), из EF.ATR и EF.DIR (см. 8.2.1.1) в том порядке, в каком они представлены. Эти межотраслевые информационные объекты должны интерпретироваться в соответствии со схемами распределения тегов, определенными в 5.2.4.

- Если приложение выбрано, то межотраслевые информационные объекты должны быть извлечены прямо или косвенно из данных управления (см. 5.3.3) DF приложения и из конкретных файлов EF в пределах текущего DF:

- межотраслевые информационные объекты могут присутствовать в данных управления любым файлом (см. 5.3.3);

- межотраслевые элементы данных могут быть извлечены из файлов, обращение к которым происходит из враппера (см. 8.5.1). Выбор безымянного DF или EF, известного по его пути, определен в 8.3. Считывание данных в выбранном EF или DF определено в 8.4;

- межотраслевые информационные объекты могут быть извлечены с помощью команд GET DATA (см. 7.4.2).

8.5.1 Косвенные ссылки на элементы данных

Список элементов, список тегов, список заголовков, список расширенных заголовков и врапперы представляют собой межотраслевые элементы данных для косвенного обращения к элементам данных в строке байтов, например, содержание файлов EF, поддерживающих единицы данных, поля данных, являющиеся результатом выполнения командных APDU (см. 8.4), строки байтов для подписи (см. ИСО/МЭК 7816-8 [4]). Такой элемент данных сообщает карте, как интерпретировать поле данных команды или как сформировать поле данных ответа.

Список элементов. Данный межотраслевой элемент данных, обращение к которому происходит с помощью тега '5F41', обозначает, что информация для извлечения не представлена в качестве информационных объектов, но находится под управлением приложения. Он должен использоваться только в пределах шаблона враппера. Настоящий стандарт не определяет его структуру и обратную информацию.

Список тегов. Данный межотраслевой элемент данных, обращение к которому происходит с помощью тега '5C', представляет собой сцепление полей данных без разграничений. Строка байтов состоит из информационных объектов, находящихся в том же порядке, что и список тегов.

Список заголовков. Данный межотраслевой элемент данных, обращение к которому происходит с помощью тега '5D', представляет собой сцепление пар полей тегов и полей длин без разграничений. Строка байтов состоит из полей значений, находящихся в том же порядке, что и список заголовков.

Список расширенного заголовка. Данный межотраслевой элемент данных, обращение к которому происходит с помощью тега '4D', представляет собой сцепление пар полей тегов и полей длин без разграничений. Строка байтов формируется следующим образом:

- если тег указывает на простое кодирование, то пара поля тега и поля длины заменяется данными, обращение к которым происходит с помощью тега;

- тег, указывающий на составное кодирование, с последующей ненулевой длиной, за исключением '80', вводит поле значения, которое представляет собой список расширенных заголовков. Тег, указывающий на составное кодирование, с последующей нулевой длиной игнорируется. Тег, указывающий на составное кодирование, с последующим значением '80' означает, что полный составной информационный объект/полный шаблон включен в строку байтов;

- карта должна игнорировать элементы из списка расширенного заголовка, который не соответствует заданной структуре.

Строка байтов состоит либо из:

- полей значения простых информационных объектов, возможно усеченных в соответствии с указанными длинами (случай 1), или из

- простых информационных объектов, возможно усеченных в соответствии с указанной длиной и вложенных в соответствующий шаблон, длина которого соответствует правилам BER-TLV (случай 2);

- длина '80', если представлена, должна быть заменена актуальной длиной. Полный составной информационный объект/полный шаблон включен в строку байтов.

Кодирование строки байтов, а именно, информационных объектов или элементов данных, указано соответствующим кодом INS или соответствующим параметром команды, например соответствующее кодирование поля данных (либо составное для содержащих информационные объекты, либо простое для содержащих элементы данных), или теги 'AC' или 'BC' (см. таблицу 31), используемые в команде PERFORM SECURITY OPERATION (см. ИСО/МЭК 7816-8 [4]).

Примеры — Следующий список расширенного заголовка обращается к последующим трем простым информационным объектам.

Простой T_1	*'00'*	*Составной тег Т*	*L = 4*	*Простой T_2*	*'00'*	*Простой T_3*	*L = 5*

Простой T_1	*L_1*	*Значение$_1$*

Простой T_2	*L_2*	*Значение$_2$*

Простой T_3	*$L_3(\geq 5)$*	*Значение$_3$*

Случай 1: Строка байтов является сцеплением элементов данных.

Значение$_1$	*Значение$_2$*	*Первые пять байтов значения$_3$*

Случай 2: Строка байтов является сцеплением информационных объектов.

T_1	*L_1*	*Значение$_1$*	*Т*	*$L = L_2 + 9$*	*T_2*	*L_2*	*Значение$_2$*	*T_3*	*L = 5*	*Первые пять байтов значения$_3$*

Враппер. Данный межотраслевой шаблон, обращение которому происходит с помощью тега '63', состоит из двух информационных объектов.

- Первый информационный объект представляет собой либо список элементов (тег '5F41'), либо список тегов (тег '5C'), либо список заголовков (тег '5D'), либо список расширенных заголовков (тег '4D');
- Второй информационный объект представляет собой ссылку на файл EF (тег '51', см. 5.3.1.2) и/или один или несколько команд-на-выполнение (тег '52'). Если командных APDU несколько, то они должны обрабатываться в представленном порядке.

Информационный объект, обращение к которому происходит, например, в списке тегов, или элемент данных, обращение к которому происходит, например, в списке заголовков, должны либо содержаться в ссылочном файле, либо быть (частью) поля данных ответа на последний командный APDU. Во враппере должна быть задана только одна косвенная ссылка. Врапперов может быть несколько.

Например — Следующий шаблон враппера состоит из списка тегов и одной команды-на выполнение.

{'63' — L — {'5C' — L — (Тег1 — Тег2 — Тег3)} — {'52' — L — Командный APDU}}

8.6 Строки байтов, образованные картой

Данная услуга позволяет карте образовывать строки байтов.

Для ясности, в данном разделе говорится о запросе, как о строке байтов [или ее части], образованном картой, и об отклике, как об ответе [или его части], посылаемом объектом во внешние устройства; например, полный набор запросов может формировать командный APDU, а полный набор откликов — ответный APDU, таким образом, предоставление услуги по коммутационному взаимодействию карты и устройства сопряжения, а также между карами возможны благодаря сети.

В данном разделе определены следующие три положения:

- как карта должна использовать SW1-SW2 в качестве сигнала запуска, указывающего, что карта хочет вызвать строку байтов, на которую карта, возможно, ожидает ответ;
- как устройство сопряжения должно использовать команду GET DATA (см. 7.4.2) для извлечения запроса с карты и команду PUT DATA (см. 7.4.3) для передачи отклика, при его наличии, в карту;
- как строка байтов должна форматироваться.

8.6.1 Включение картой

Байты SW1-SW2, установленные на '62XX' со значением 'XX' от '02' до '80' означают, что у карты есть запрос из 'XX' байтов, который устройство сопряжения должно извлечь, и на который карта, возможно, ожидает получить ответ.

Байты SW1-SW2, установленные на '64XX' со значением 'XX' от '02' до '80' означают, что карта прервала команду; возможное завершение команды обусловлено восстановлением запроса из 'XX' байтов, на который карта, возможно, ожидает получить ответ.

Если SW1-SW2 присутствуют в байтах предыстории со значением, как указано выше, то эти байты должны интерпретироваться, как указано выше.

Если команда PUT DATA (см. 7.4.3) для передачи отклика прервана с SW1-SW2, установленными на '64XX', то

- со значением '64XX' от '6402' до '6480' карта хочет послать, как минимум, еще один запрос их 'XX' байтов;

- со значением '64XX', установленным на '6401' карта ожидает немедленный ответ.

8.6.2 Запросы и отклики

Для извлечения запроса из 'XX' байтов, доступных в карте, устройство сопряжения должно послать команду GET DATA с байтами P1-P2, установленными на '0000', и полем L_e, установленным на 'XX':

- байты SW1-SW2, установленные на '62XX' со значением 'XX' от '02' до '80', означают, что устройство сопряжения должно извлечь дальнейший запрос из 'XX' байтов и объединить его в цепочку с уже извлеченным запросом до обработки строки байтов, образованной картой, во внешние устройства;

- байты SW1-SW2, установленные на '9000', означают, что строка байтов, образованная картой, завершена; она может обрабатываться внешними устройствами.

Для передачи отклика карте устройство сопряжения должно послать команду PUT DATA с байтами P1-P2, установленными на '0000'. Если ответ слишком длинный для одной команды, то несколько команд PUT DATA должны быть объединены в цепочку (см. 5.1.1.1). Каждая команда PUT DATA передает отклик и сцепление откликов в ответе.

8.6.3 Форматы

Значение первого байта строки байтов, образованной картой, указывает формат следующим образом:

- если первый байт установлен на 'FF', то последующие байты должны кодировать идентификатор исходного протокола в соответствии с ИСО/МЭК ТО 9577; строки байтов должны соответствовать указанному протоколу;

- в противном случае (т. е. когда первый байт не установлен на 'FF'), строка байтов, образованная картой, и ответ вместе должны формировать пару команда-ответ.

Все условия относятся к протоколу передачи, указанному картой, за исключением надлежащего использования команды GET DATA, команды PUT DATA и байтов состояния SW1-SW2. В данном разделе не делается предположение о необходимости ответа и об объекте, отвечающем на содержание возможного ответа.

Приложение А
(справочное)

Примеры идентификаторов объекта и схем распределения тегов

А.1 Идентификаторы объекта

Для стандартов ИСО первым байтом является '28', т. е. десятичное число 40 (см. ИСО/МЭК 8825-1). Следует одна или несколько серий байтов; бит 8 установлен на 0 в последнем байте серии и на 1 в предыдущих байтах, если имеется более одного байта. Сцепление бит с 7 по 1 в байте из серии кодирует число. Каждое число должно быть закодировано с минимальным количеством байтов, т. е. значение '80' является недействительным для первого байта из серии. Первое число представляет собой номер стандарта, второе число, если оно имеется, — номер части стандарта, состоящего из нескольких частей.

В первом примере {iso(1) standard(0) ic-cards(7816)} ссылается на ИСО/МЭК 7816:

- 7816 равно '1E88', т.е. 0001 1110 1000 1000, т.е. два блока из семи бит 0111101 0001000;
- после ввода бита 8 с соответствующим значением в каждый байт, кодирование первой серии становится, таким образом, 1011 1101 0000 1000, что равно 'BD08'.

Элемент данных '28 BD08' может быть использован в AID стандартной категории (см. 8.2.1.2).

AID = 'E8 28 BD08 0B XX ... XX' (ИСО/МЭК 7816-11 определяет расширение идентификатора приложения 'XX ... XX').

AID = 'E8 28 BD08 0F XX ... XX' (ИСО/МЭК 7816-15 определяет расширение идентификатора приложения 'XX ... XX').

Во втором примере {iso(1) standard(0) e-auth(9798) part(5)} ссылается на ИСО 9798-5 [8]. Первая серия получается следующим образом:

- 9798 равно '2646', т. е. 0010 0110 0100 0110, т. е. два блока из семи бит: 1001100 1000110;
- после ввода бита 8 с соответствующим значением в каждый байт, кодирование первой серии становится, таким образом, 11001100 01000110, что равно 'CC46'.

Элемент данных '28 CC46 05 02' ссылается на второй механизм в ИСО/МЭК 9798-5 [8], т.е. GQ2. Такой идентификатор может быть передан в информационный объект (тег '06', универсальный класс, см. ИСО/МЭК 8825-1).

DO = {'06 05 28 CC 46 05 02'}.

В третьем примере {iso(1) standard(0) mess(9992) part(2)} ссылается на ИСО 9992-2 [10]. Первая серия получается следующим образом:

- 9992 равно '2708', т. е. 0010 0111 0000 1000, т. е. два блока из семи бит: 1001110 0001000;
- после ввода бита 8 с соответствующим значением в каждый байт, кодирование первой серии становится, таким образом, 1100 1110 0000 1000, что равно 'CE08'.

Элемент данных '28 CE08 02' (второй серии — '02'). Он может быть передан в информационный объект.

DO = {'06 04 28 CE 08 02'}.

А.2 Схемы распределения тегов

Пример схемы распределения тегов по умолчанию

DO1 = {'59 02 95 02'}

DO2 = {'5F 24 03 97 03 31'}

DO1 (тег '59', дата истечения срока действия карты) кодирует «Февраль 1995» в качестве даты истечения срока действия карты (см. ИСО/МЭК 7816-6).

DO2 (тег '5F24', дата истечения срока действия приложения) кодирует «31 Марта 1997» в качестве даты истечения срока действия приложения.

Примеры совместимых схем распределения тегов

DO1 = {'78 06' {'06 04 28 CE 08 02'}}

DO2 = {'5F 24 03 97 03 31'}

DO3 = {'70 04'{'80 02 XX XX'}}

DO4 = {'67 0A' {'5F 29 03 XX XX XX'}{'81 02 XX XX'}}

DO1 (тег '78', орган распределения совместимых тегов) указывает на совместимую схему распределения тегов, определенную в ИСО 9992-2 [10], к которой обращаются с помощью своего идентификатора объекта. Если DO1 появляется либо в строке исходных данных (см. 8.1.2), или в EF.ATR (см. 8.2.1.1), то орган распределения тегов является действительным для карты в целом. Если DO1 появляется в данных управления файлом DF (см. 5.3.3), то орган распределения тегов является действительным в пределах этого DF.

DO2 (тег '5F24', дата истечения срока действия приложения) кодирует «31 Марта 1997» в качестве даты истечения срока действия приложения.

DO3 (тег '70', межотраслевой шаблон в соответствии с включенным органом распределения тегов) содержит информационный объект с тегом '80', определяемый в ИСО 9992-2 [10]; значение тега '70' также определено в ИСО 9992-2 [10].

DO4 (тег '67', шаблон данных аутентификации) содержит информационный объект «профиль обмена» с тегом '5F29' и информационный объект с тегом '81', определяемый в ИСО 9992-2 [10]; содержание тега '67' определено в ИСО/МЭК 7816-6[4].

Другой пример совместимой схемы распределения тегов

DO2 = {'5F 24 03 97 03 31'}

DO3 = {'70 0C' {'06 04 28 CE 08 02'} {'80 04 XX XX XX XX'}}

DO4 = {'67 06' {'5F 29 03 XX XX XX'}}

DO2 (тег '5F24', дата истечения срока действия приложения) кодирует «31 Марта 1997» в качестве даты истечения срока действия приложения.

DO3 (тег '70', межотраслевой шаблон, определенный в соответствии с включенным идентификатором объекта) содержит информационный объект с тегом '06', который устанавливает, что последующий информационный объект с тегом '80' определен в ИСО 9992-2 [10].

DO4 (тег '67', межотраслевой шаблон данных аутентификации) содержит информационный объект «профиль обмена» с тегом '5F29'. Он не может содержать информационные объекты, определенные в ИСО 9992-2[10], из-за выбора не передавать межотраслевой информационной объект с тегом '78'.

Пример сосуществующей схемы распределения тегов

DO1 = {'79 05' {'06 03 28 XX XX'}}

DO2 = {'7E 06' {'5F 24 03 97 03 31'}}

DO3 = {'70 06' {'XX XX XX XX XX XX'}}

DO1 (тег '79', сосуществующий орган распределения тегов) указывает сосуществующую схему распределения тегов, определяемую в стандарте, к которому обращаются с помощью идентификатора объекта со значением, начинающимся с "28", т. е. стандарта ИСО. DO1 является обязательным в такой схеме и должен появляться либо

- в строке начальных данных (см. 8.1.2) или в EF.ATR (см. 8.2.1.1), если орган распределения тегов является действительным для карты в целом, либо

- в данных управления файлом DF (см. 5.3.3), если орган распределения тегов является действительным в пределах этого DF.

DO2 (тег '7E') является межотраслевым шаблоном для вложенных межотраслевых информационных объектов. Межотраслевой элемент данных «дата истечения срока действия приложения» с тегом '5F24' присутствует, кодируя «31 Марта 1997» как дату истечения срока действия приложения.

DO3 (тег '70', межотраслевой шаблон, который должен интерпретироваться в соответствии с органом распределения тегов, указанном в шаблоне '79') может быть интерпретирован только в соответствии со стандартом, указанным в идентификаторе объекта.

Приложение В
(справочное)

Примеры безопасного обмена сообщениями

В.1 Криптографическая контрольная сумма

В данном разделе показано использование безопасного обмена сообщениями (см. 6) и криптографическими контрольными суммами (см. 6.2.3.1) для каждого из четырех случаев пары команда-ответ, определенных в ИСО/МЭК 7816-3.

В примерах запись CLA* обозначает использование безопасного обмена сообщениями в полях данных: в CLA (см. 5.1.1) биты 8, 7 и 6, установлены на 000, а бит 4 установлен на 1 или биты 8, 7 и 6 установлены на 011.

В примерах запись CLA** обозначает, что биты 8, 7 и 6 в CLA установлены на 000, а биты 4 и 3 — на 11, т .е. заголовок команды должен быть включен при вычислении элемента данных для аутентификации.

С другой стороны, заголовок может быть инкапсулирован в информационный объект с тегом '89', т. е. информационный объект SM должен быть включен при вычислении элемента данных для аутентификации.

В примерах запись T* обозначает, что бит 1 последнего байта поля тега установлен на 1 (нечетный номер тега), т. е. информационный объект SM должен быть включен при вычислении элемента данных для аутентификации.

- Случай 1 — Нет данных команды, нет данных ответа

Незащищенная пара команда-ответ следующая:

Заголовок команды	Тело команды
CLA INS P1 P2	Пустое

Тело ответа	Завершитель ответа
Пустое	SW1-SW2

- Случай 1.a — Состояние не защищено

Защищенный командный APDU следующий:

Заголовок команды	Тело команды
CLA* INS P1 P2	{Новое поле Lc} — {Новое поле данных (= T — L — Криптографическая контрольная сумма)}

Если длина криптографической контрольной суммы равна четырем байтам, то новое поле L_c установлено на '06'.

Новое поле данных = Один информационный объект = {T — L — Криптографическая контрольная сумма}
Данные, охватываемые криптографической контрольной суммой (бит 3 в CLA* установлен на 1) =
Один блок = {CLA** INS P1 P2 Заполнение}
Защищенный ответный APDU следующий:

Тело ответа	Завершитель ответа
Пустое	SW1-SW2

- Случай 1.b — Состояние защищено

Защищенный командный APDU следующий:

Заголовок команды	Тело команды
CLA* INS P1 P2	{Новое поле L_c} — {Новое поле данных (= T — L — Криптографическая контрольная сумма)} — {Новое поле L_e (= '00')}

Новое поле данных = Один информационный объект = {T — L — Криптографическая контрольная сумма}
Данные, охватываемые криптографической контрольной суммой (бит 3 в байте CLA* установлен на 1) = Один блок = {CLA** INS P1 P2 Заполнение}

Тело ответа	Завершитель ответа
Новое поле данных (= {T* — L — SW1-SW2} — {T — L — Криптографическая контрольная сумма})	SW1-SW2

Новое поле данных = Два информационных объекта = {T* — L — SW1-SW2} — {T — L — Криптографическая контрольная сумма})
Данные, охватываемые криптографической контрольной суммой = Один блок = {T* — L — SW1-SW2 — Заполнение}

- Случай 2 — Нет данных команды, данные ответа

Незащищенная пара команда-ответ следующая:

Заголовок команды	Тело команды
CLA INS P1 P2	Поле L_e

Тело ответа	Завершитель ответа
Поле данных	SW1-SW2

Защищенный командный APDU следующий:

Заголовок команды	Тело команды
CLA* INS P1 P2	Новое поле L_c — Новое поле данных — {Новое поле L_e (один или несколько байтов, установленных на '00')}

Новое поле данных = Два информационных объекта = {T* — L — L_e} — {T — L — Криптографическая контрольная сумма}
Данные, охватываемые криптографической контрольной суммой =
- Один блок = {T* — L — L_e — Заполнение}, если бит 3 в CLA* установлен на 0;
- Два блока = {CLA** INS P1 P2 Заполнение} — {T* — L — L_e — Заполнение}, если бит 3 в CLA* установлен на 1.

Защищенный ответный APDU следующий:

Тело ответа	Завершитель ответа
Новое поле данных (= {T* — L — Простое значение} — {T* — L — SW1-SW2} — {T — L — Криптографическая контрольная сумма})	SW1-SW2

Новое поле данных = Три информационных объекта =
{T* — L — Простое значение} — {T* — L — SW1-SW2} — {T — L — Криптографическая контрольная сумма})
Данные, охватываемые криптографической контрольной суммой =
Один или несколько блоков = {T* — L — Простое значение — T* — L — SW1-SW2 — Заполнение}

- Случай 3 — Данные команды, нет данных ответа

Незащищенная пара команда-ответ следующая:

Заголовок команды	Тело команды
CLA INS P1 P2	Поле L_c — Поле данных

Тело ответа	Завершитель ответа
Пустое	SW1-SW2

- Случай 3.а — Состояние не защищено

Защищенный командный APDU следующий:

Заголовок команды	Тело команды
CLA* INS P1 P2	Новое поле L_c — Новое поле данных

Новое поле данных = Два информационных объекта = {T* — L — Простое значение} — {T* — L — Криптографическая контрольная сумма)

Данные, охватываемые криптографической контрольной суммой =

- Один или несколько блоков = {T* — L — Простое значение —Заполнение}, если бит 3 в CLA* установлен на 0;
- Два или несколько блоков = {CLA** INS P1 P2 Заполнение} — {T* — L — Простое значение — Заполнение}, если бит 3 в CLA* установлен на 1.

Защищенный ответный APDU следующий:

Тело ответа	Завершитель ответа
Пустое	SW1-SW2

- Случай 3.b — Состояние защищено

Защищенный командный APDU следующий:

Заголовок команды	Тело команды
CLA* INS P1 P2	Новое поле L_c — Новое поле данных — Новое поле L_e (= "00")

Новое поле данных = Два информационных объекта = {T* — L — Простое значение} — {T* — L — Криптографическая контрольная сумма)

Данные, охватываемые криптографической контрольной суммой =

- Один или несколько блоков = {T* — L — Простое значение — Заполнение}, если бит 3 в CLA* установлен на 0;
- Два или несколько блоков = {CLA** INS P1 P2 Заполнение} — {T* — L — Простое значение — Заполнение}, если бит 3 в CLA* установлен на 1.

Защищенный ответный APDU следующий:

Тело ответа	Завершитель ответа
Новое поле данных (= {T* — L — SW1-SW2} — {T — L — Криптографическая контрольная сумма})	SW1-SW2

Новое поле данных = Два информационных объекта = {T* — L — SW1-SW2} — {T — L — Криптографическая контрольная сумма}

Данные, охватываемые криптографической контрольной суммой = Один блок = {T* — L — SW1-SW2 — Заполнение}

- Случай 4 — Данные команды, данные ответа

Незащищенная пара команда-ответ следующая:

Заголовок команды	Тело команды
CLA INS P1 P2	поле L_c — Поле данных — поле L_e

Тело ответа	Завершитель ответа
Поле данных	SW1-SW2

Защищенный командный APDU следующий:

Заголовок команды	Тело команды
CLA* INS P1 P2	Новое поле L_c — Новое поле данных — Новое поле L_e (один или два байта, установленные на '00')

Новое поле данных = Три информационных объекта =

{T* — L — Простое значение} — {T* — L — L_e} — {T — L — Криптографическая контрольная сумма}

Данные, охватываемые криптографической контрольной суммой =

- Один или несколько блоков = {T* — L — Простое значение — T* — L — L_e — Заполнение}, если бит 3 в CLA* установлен на 0;
- Два или несколько блоков = {CLA** INS P1 P2 Заполнение} — {T* — L — Простое значение — T* — L — L_e — Заполнение }, если бит 3 в CLA* установлен на 1.

Защищенный ответный APDU следующий:

Тело ответа	Завершитель ответа
Новое поле данных (= {T* — L — Простое значение} — {T* — L — SW1-SW2} — {T — L — Криптографическая контрольная сумма})	SW1-SW2

Новое поле данных = Три информационных объекта =

{T* — L — Простое значение} — {T* — L — SW1-SW2} — {T — L — Криптографическая контрольная сумма}

Данные, охватываемые криптографической контрольной суммой = Один или несколько блоков =

{T* — L — Простое значение — T* — L — SW1-SW2 — Заполнение}

B.2 Криптограммы

Использование криптограмм с заполнением блоков данных незначащей информацией и без заполнения (см. 6.2.2) показано в полях данных команды и ответа. Вместо информационных объектов «простое значение», фигурирующих в предыдущих примерах, использованы информационные объекты для конфиденциальности следующим образом.

- Случай a — Простое значение, не закодированное в BER-TLV

Поле данных = {T — L — Байт индикатора заполнения незначащей информацией/с информационным наполнением — Криптограмма}

Простое значение, передаваемое криптограммой = Один или несколько блоков =

Простое значение, не закодированное в BER-TLV, возможно заполненное в соответствии с байтом индикатора.

- Случай b — Простое значение, закодированное в BER-TVL

Поле данных = {T — L — Криптограмма}

Простое значение, передаваемое криптограммой = Строка скрываемых байтов =

Информационные объекты BER-TLV (заполнение незначащей информацией, зависящей от алгоритма и режима его работы).

B.3 Управляющие ссылки

Показано использование управляющих ссылок (см.6.3.1 и 6.3.2).

Поле данных команды = {T — L — Шаблон управляющих ссылок},

Где шаблон управляющих ссылок = {T — L — Ссылка на файл} — {T — L — Ссылка на ключ}

B.4 Описатель ответа

Показано использование описателя ответа (см. 6.3.3)

Поле данных команды = {T — L — Описатель ответа},

где Описатель ответа = {T(Простое значение)} — '00' — T(Криптографическая контрольная сумма) — '00'}

Поле данных команды = {T — L — Простое значение} — {T — L — Криптографическая контрольная сумма}

B.5 Команда ENVELOPE

Показано использование команды ENVELOPE (см. 7.6.2).

Поле данных команды = {T — L — Байт индикатора заполнения незначащей информацией/с информационным наполнением — Криптограмма}

Простое значение, передаваемое криптограммой =

Командный APDU (начиная с CLA* INS P1 P2), заполнение незначащей информацией в соответствии с байтом индикатора.

Поле данных ответа = {T — L — Байт индикатора заполнения незначащей информацией/с информационным наполнением — Криптограмма}

Простое значение, передаваемое криптограммой =

Ответный APDU, заполнение незначащей информацией в соответствии с байтом индикатора.

B.6 Совместное использование безопасного обмена сообщениями и операций по защите информации

В настоящем разделе применены следующие обозначения и сокращения:

CC — криптографическая контрольная сумма (cryptographic checksum);

CG — криптограмма (cryptogram);

CLA** — CLA с индикацией SM (биты 8, 7 и 6, установленные на 000, и биты 4 и 3, установленные на 11);

DS — Цифровая подпись;

MSE — Управление безопасной средой;

PCI — Байт индикатора заполнения незначащей информацией/с информационным наполнением (padding-content indicator byte);

PSO — Выполнить операцию защиты (perform security operation);

SMC — Карта с защитным модулем (security module card);

USC — Смарт-карта пользователя (user smart card).

Пример показывает, как использовать смарт-карту пользователя (SMC), которая выполняет операции по защите для создания защищенного командного APDU, который необходимо послать в карту пользователя (USC), и для обработки соответствующего защищенного ответного APDU, полученного от USC, т. е. для получения и обработки полей данных в формате SM. Данный пример иллюстрирует совместное использование двух подходов: элементарный подход, при использовании операций по защите (см. ИСО/МЭК 7816-8 [4]) и глобальный подход при использовании безопасного обмена сообщениями (см. раздел 6).

В примере предполагается, что карты USC и SMC завершили процедуру взаимной аутентификации, основанной, например, на сертификатах, верифицируемых картой. Процедура аутентификации включает передачу ключа или механизм согласования ключей, таким образом, после этой процедуры два симметричных ключа становятся доступными в USM и в SMC:

- симметричный сеансовый ключ для вычисления криптографической контрольной суммы и
- симметричный сеансовый ключ для вычисления криптограмм.

Процедура аутентификации инициирует один или несколько счетчиков в USC и в SMC. В примере не показано сопровождение и использование таких счетчиков в USC и в SMC.

Все пары команда-ответ для SMC являются командами PSO, не использующими безопасный обмен сообщениями, но использующими информационные объекты SM (и ключи SM, установленные с помощью команд MSE).

Все пары команда-ответ для USM используют безопасный обмен сообщениями и заголовки команд, которые включены при вычислении криптографической контрольной суммы, т. е. CLA переключается на CLA**.

На рисунке В.1 показаны общие принципы для создания защищенного командного APDU.

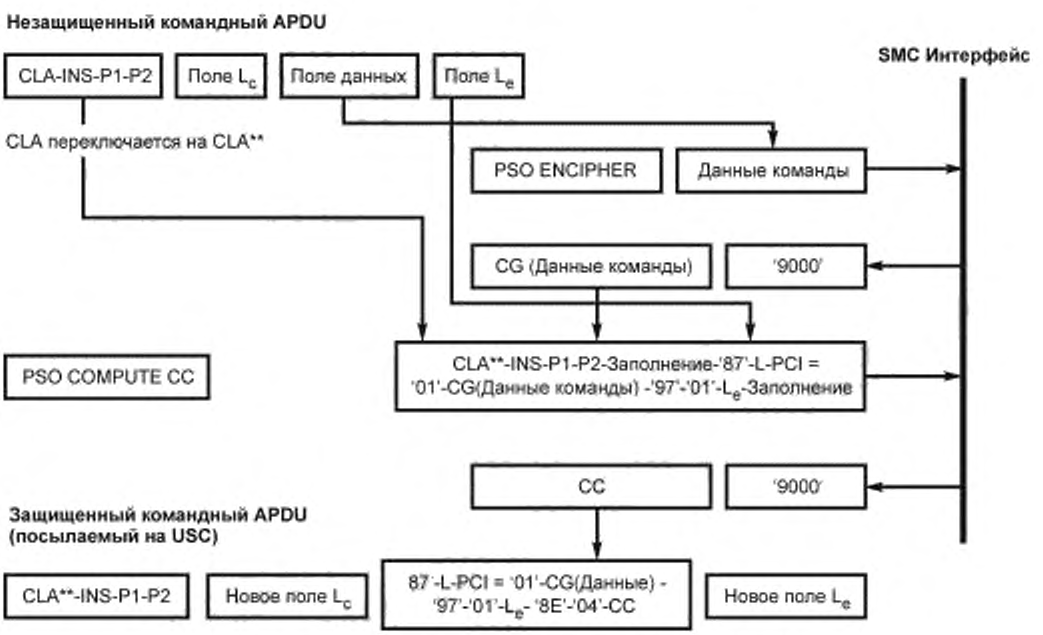


Рисунок В.1 — Создание защищенного командного APDU

На рисунке В.2 показаны общие принципы создания защищенного ответного APDU.

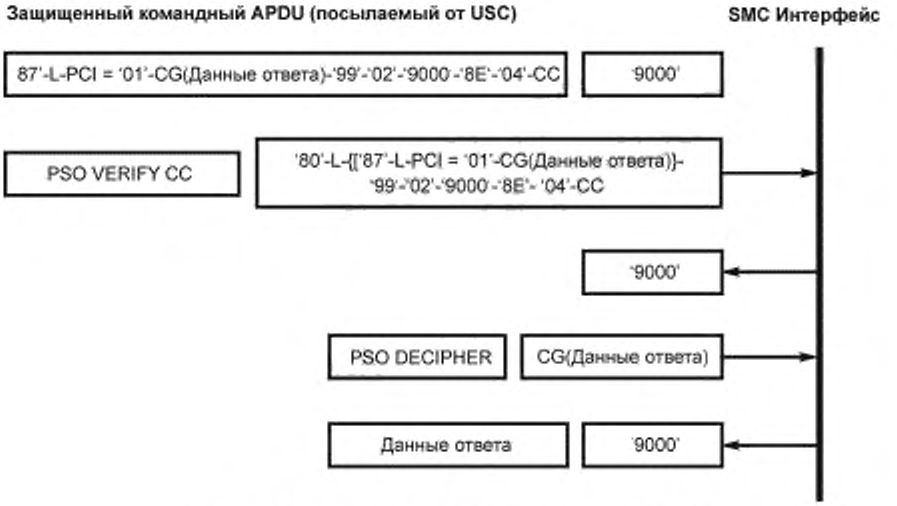


Рисунок В.2 — Создание защищенного ответного APDU

Следующий сценарий объясняет вычисление цифровой подписи (DS), где использование приватного ключа подписи требует успешного представления пароля. Сценарий проходит в три этапа:

Этап 1 — Верификация паролем

1.1 Команда для SMC: MSE SET <CT, {'83' — '01' — '81'}>

- - - В примере ссылка на сессионный ключ для вычисления криптографических контрольных сумм — '81'.

Ответ SMC: OK.

1.2 Команда для SMC: MSE SET <CCT, {'83' — '01' — '82'}>

- - - В примере ссылка на сессионный ключ для вычисления криптограмм — '82'.

Ответ SMC: OK.

1.3 Команда для SMC: PSO ENCIPHER <Пароль>

Ответ SMC: <CG(Пароль)>

1.4 Команда для SMC : PSO COMPUTE CC <CLA** — INS — P1 — P2 — Заполнение — {'87' — L — PCI — CG(Пароль)} — {'97' — '01' — L_e} — Заполнение>

Ответ SMC: <CC>

- - -Теперь устройство сопряжения создает командный APDU, защищенный с помощью VERIFY.

1.5 Команда для USC: VERIFY <{'87' — L — PCI = '01' — CG(Пароль)} — {'97' — '01' — L_e} — {'8E' — '04' — CC}>

Ответ USC: <{'99' — '02' — SW1-SW2} — {'8E' — '04' — CC}>

1.6 Команда для SMC: PSO VERIFY CC <{'80' — '04' — ('99' — '02' — SW1-SW2)} — {'8E' — '04' — CC}>

Ответ SMC: OK.

Этап 2 — Вычисление хэш-кода

2.1 Команда для SMC: PSO COMPUTE CC <CLA** — INS — P1 — P2 — Заполнение — {'81' — L — ({'90' — L — Промежуточный хэш} — {'80' — L — Последний блок})} — {'97' — '01' — L_e} — Заполнение>

Ответ SMC: <CC>

2.2 Команда для USC: PSO HASH <{'81' — L1 (=4 + L2 + L3) — ({'90' — L2 — Промежуточный хэш} — {'80' — L3 — Последний блок})} — {'8E' — '04' — CC}>

- - -USC записывает хэш-код как внутренний результат для вычисления цифровой подписи в дальнейшем.

Ответ USC: <{'99' — '02' — SW1-SW2} — {'8E' — '04' — CC}>

2.3 Команда для SMC: PSO VERIFY CC <{'80' — '04' — ({'99' — '02' — SW1-SW2})} — {'8E' — '04' — CC}>

Ответ SMC: OK

Этап 3 — Вычисление цифровой подписи

1.1 Команда для SMC: PSO COMPUTE CC <CLA** — INS — P1 — P2 — Заполнение — {'97' — '01' — '00'}

Ответ SMC: <CC>

1.2 Команда для USC: PSO COMPUTE DS <{'97' — '01' — '00'} — {'8E' — '04' — CC}>

Ответ USC: <'81' — L — DS — '8E' — '04' — CC>

1.3 Команда для SMC: PSO VERIFY CC <{'80' — L1 (=2 +L2) — ('81' — L2 — DS)} — {'8E' — '04' — CC}>

Ответ SMC: OK

Приложение С
(справочное)

Примеры функций AUTHENTICATE в командах GENERAL AUTHENTICATE

С.1 Введение

Одна или несколько пар команда-ответ GENERAL AUTHENTICATE выполняет функцию AUTHENTICATE.

- Если используют сцепление команд, то CLA устанавливают на 0xx1 xxxx в первой команде цепочки до предпоследней команды и на 0xx0 xxxx в последней команде: остальные шесть бит должны оставаться постоянными в пределах цепочки (см. 5.1.1.1);
- INS P1 P2 устанавливают либо на '86 00 00', либо на '87 00 00';
- Значение поля L_e зависит от информационных объектов в поле данных команды. В зависимости от того ожидается ли поле данных ответа или нет, поле L_e устанавливается либо на '00', либо оно отсутствует.

В настоящем приложении проиллюстрированы поля данных команд GENERAL AUTHENTICATE, реализовывающие механизмы по ИСО/МЭК 9798-5 [8], т. е. механизмы, использующие метод без сообщения конфиденциальных сведений.

- Устройству верификации известна открытая задача, а заявителю известно секретное решение этой открытой задачи.
- В результате выполнения протокола обмена с минимальной передачей конфиденциальной информации, устройство верификации удостоверяется, что заявителю известно решение этой открытой задачи. При этом решение остается секретным.

Примечание — В ИСО/МЭК 9798-5[8] определены два метода GQ:

- при заданном открытом ключе RSA, где экспонента v является простым числом, таким как $257 = 2^8 + 1$, $65537 = 2^{16} + 1$ или $2^{36} + 2^{13} + 1$, метод GQ1 позволяет верифицировать подпись RSA без использования информации о ее значении, или, в противном случае, подтвердить информацию о подписи RSA без раскрытия ее значения. Как определено в действующем стандарте на подпись RSA (например, см. ИСО/МЭК 14888-2 [16]) механизм формата переводит идентификационные данные заявителя (шаблон) в открытое число G. Соответствующее приватное число Q является подписью RSA идентификационных данных. Заявителю и устройству верификации известен открытый ключ RSA. Протокол GQ1 гарантирует, что заявителю известна подпись RSA этих идентификационных данных;
- при заданном открытом модуле n (произведение двух простых факторов) метод GQ2 позволяет верифицировать факторы без использования информации и них, или в противном случае, подтвердить информацию о факторах без раскрытия их значения. Механизм включает в себя параметр безопасности $k > 0$ и первые m простых чисел, называемых основными числами m, такими что $k \times m$ находится в пределах от 8 до 36. Каждое открытое число равно квадрату основного числа: $G = g^2$. Соответствующее приватное число Q равно модулю 2^{k+1} — корню из G. Если имеется, как минимум, одно основное число g, такое что символ Якоби для g по отношению к n равен — 1 и, если n конгруэтно 1 на модуль 4, то протокол GQ2 подтверждает, что n является составным и что заявителю известны факторы.

Протокол обычно обменивается тремя числами, а именно свидетельством, задачей и ответом:

- заявитель действует в два этапа: на первом этапе заявитель приватно выбирает новое случайное число и переводит его в свидетельство в соответствии с «формулой свидетельства»; на втором этапе, при получении задачи, заявитель получает ответ на задачу исходя из нового случайного числа и приватного числа в соответствии с «формулой ответа», и далее стирает новое случайное число.
- устройство верификации восстанавливает свидетельство исходя из задачи и ответа в соответствии с «формулой верификации».

Эта тройка состоит из трех чисел, а именно, свидетельства, задачи и ответа, верифицирующих формулу верификации. Любой объект может произвольно создать тройку в «открытом режиме» исходя из любой задачи и ответа. Эксперт или оператор не могут различать тройки, созданные в открытом режиме (субъектом, которому не известен секрет) и случайные тройки, созданные в «приватном режиме» (субъектом, которому известен секрет).

В настоящем приложении проиллюстрированы три функции:

- Функция INTERNAL AUTHENTICATE — устройство верификации во внешнем устройстве аутентифицирует заявителя в карте;
- Функция EXTERNAL AUTHENTICATE — устройство верификации в карте аутентифицирует заявителя во внешнем устройстве;
- Функция MUTUAL AUTHENTICATE — оба объекта идентифицируют друг друга.

С.2 Функция INTERNAL AUTHENTICATE

Если первое поле данных передает запрос свидетельства, а именно пустое свидетельство ('80 00'), либо пустой код аутентификации ('84 00'), то функцией является INTERNAL AUTHENTICATE.

- Основной протокол (две пары команда-ответ)

Свидетельство из карты

Поле данных команды	{'7C' — '02' — {'80' — '00'}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'80' — L2 — Свидетельство}}

Задача из внешнего устройства и ответ из карты

Поле данных команды	{'7C' — L1 (=4 + L2) — {'81' — L2 — Свидетельство}} — {'82' — '00'}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'82' — L2 — Ответ}}

Фиксированная задача (две пары команда-ответ)

Свидетельство из карты

Поле данных команды	{'7C' — L1 (=4 + L2) — {'83' — L2 — Фиксированная задача} — {'80' — '00'}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'80' — L2 — Свидетельство}}

Примечание — Фиксированная задача гарантирует, что задача и свидетельство выбраны независимо друг от друга.

Задача из внешнего устройства и ответ из карты

Поле данных команды	{'7C' — L1 (=4 + L2) — {'81' — L2 — Свидетельство}} — {'82' — '00'}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'82' — L2 — Ответ}}, если задача верная Отсутствует, если задача не верна

- Расширение для аутентификации поля данных (две пары команда-ответ)

Карта хэшировала предыдущие переданные поля данных: результатом является текущий хэш-код. Карта включает свой информационный объект «свидетельство» для получения кода аутентификации и передает его с тегом '84'.

Свидетельство из карты

Поле данных команды	{'7C' — '04' — {'84' — '00'}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'84' — L2 — Код аутентификации}}

Задача из внешнего устройства и ответ из карты

Поле данных команды	{'7C' — L1 (=4 + L2) — {'81' — L2 — Задача}} — {'82' — '00'}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'82' — L2 — Ответ}}

С.3 Функция EXTERNAL AUTHENTICATE

Если первое поле данных передает запрос задачи, а именно, либо пустую задачу ('81 00'), либо пустую фиксированную задачу ('83 00'), то функцией является EXTERNAL AUTHENTICATE.

- Основной протокол (две пары команда-ответ)

Свидетельство из внешнего устройства и задача из карты

Поле данных команды	{'7C' — L1 (=4 + L2) — {'80' — L2 — Свидетельство}} — {'81' — '00'}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'81' — L2 — Задача}}

Ответ из внешнего устройства и верификация картой

Поле данных команды	{'7C' — L1 (=2 + L2) — {'82' — L2 — Ответ}}
Поле данных ответа	Отсутствует

- Фиксированная задача (три пары команда-ответ)

Фиксированная задача из карты

Поле данных команды	{'7C' — '02' — {'83' — '00'}
Поле данных ответа	{'7C' — L1 (=4 + L2) — {'83' — L2 — Фиксированная задача} — {'80' — '00'}}

Свидетельство из внешнего устройства и задача из карты

Поле данных команды	{'7C' — L1 (=4 + L2) — {'80' — L2 — Свидетельство}} — {'80' — '00'}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'81' — L2 — Задача}}

Ответ из внешнего устройства и верификация картой

Поле данных команды	{'7C' — L1 (=2 + L2) — {'82' — L2 — Ответ}}
Поле данных ответа	Отсутствует

Расширение для аутентификации поля данных (две пары команда-ответ)

Заявитель хэшировал предыдущие переданные поля данных: результатом является текущий хэш-код. Карта включает свой информационный объект «свидетельство» для получения кода аутентификации и передает его с тегом '84'.

Свидетельство из внешнего устройства и задача из карты

Поле данных команды	{'7C' — L1 (=4 + L2) — {'84' — L2 — Код аутентификации)} — {'81' — '00'}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'81' — L2 — Задача}}

Ответ из внешнего устройства и верификация картой

Поле данных команды	{'7C' — L1 (=2 + L2) — {'82' — L2 — Ответ}}
Поле данных ответа	Отсутствует

С.4 Функция MUTUAL AUTHENTICATE

Если первое поле данных передает не пустой информационный объект, то функцией является MUTUAL AUTHENTICATE; внешнее устройство требует тот же информационный объект в поле данных ответа, что и в поле данных команды.

- Основной протокол (три пары команда-ответ)

Свидетельство

Поле данных команды	{'7C' — L1 (=2 + L2) — {'81' — L2 — Свидетельство}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'81' — L2 — Свидетельство}}

Задача

Поле данных команды	{'7C' — L1 (=2 + L2) — {'81' — L2 — Задача}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'81' — L2 — Задача}}

Ответ

Поле данных команды	{'7C' — L1 (=2 + L2) — {'82' — L2 — Ответ}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'82' — L2 — Ответ}}, если ответ верный Отсутствует, если ответ неверный

- Фиксированная задача (четыре пары команда-ответ)

Фиксированная задача

Поле данных команды	{'7C' — L1 (=2 + L2) — {'83' — L2 — Фиксированная задача}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'83' — L2 — Фиксированная задача

Свидетельство

Поле данных команды	{'7C' — L1 (=2 + L2) — {'80' — L2 — Свидетельство}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'80' — L2 — Свидетельство}}

Задача

Поле данных команды	{'7C' — L1 (=2 + L2) — {'81' — L2 — Задача}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'81' — L2 — Задача}} если задача верная Отсутствует, если задача неверная

Ответ

Поле данных команды	{'7C' — L1 (=2 + L2) — {'82' — L2 — Ответ}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'82' — L2 — Ответ}} если ответ верный Отсутствует, если ответ неверный

- Расширение для согласования ключей (четыре пары команда-ответ)

Пара элементов данных «экспоненциал» делает возможным согласование сессионных ключей (см. ИСО/МЭК 11770-3 [14]).

Первая пара команда-ответ заменяет шаблон динамической аутентификации, вложенный в элемент данных «экспоненциал». В данном примере поскольку во время сессии предварительно не было никакого обмена сообщением, то начальным хэш-кодом является нулевой блок. Далее включают поле данных команды, т. е. первый шаблон динамической аутентификации, для получения текущего хэш-кода; далее включают поле данных ответа, т.е. второй шаблон динамической аутентификации, для обновления текущего хэш-кода; текущий хэш-код должен быть одинаковым для обоих объектов. В конце включают информационный объект «свидетельство» (не нулевой и не передаваемый, отличный от каждого объекта) для получения кода аутентификации (отличный для каждого объекта).

Вторая пара команда-ответ заменяет шаблоны динамической аутентификации со вложенными кодами аутентификации с тегом '84'.

Экспоненциал

Поле данных команды	{'7C' — L1 (=2 + L2) — {'85' — L2 — Экспоненциал}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'85' — L2 — Экспоненциал}}

Свидетельство

Поле данных команды	{'7C' — L1 (=2 + L2) — {'84' — L2 — Код аутентификации}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'84' — L2 — Код аутентификации }}

Задача

Поле данных команды	{'7C' — L1 (=2 + L2) — {'81' — L2 — Задача}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'81' — L2 — Задача}}

Ответ

Поле данных команды	{'7C' — L1 (=2 + L2) — {'82' — L2 — Ответ}}
Поле данных ответа	{'7C' — L1 (=2 + L2) — {'82' — L2 — Ответ}}, если ответ верный Отсутствует, если ответ неверный

Приложение D
(справочное)

Идентификаторы приложений, использующие идентификационные номера эмитента

D.1 Справочная информация

В ИСО/МЭК 7816-5:1994[1)] была предусмотрена возможность использовать идентификационные номера эмитента в идентификаторах приложения. В настоящем приложении указан формат таких AID.

D.2 Формат

В каждом AID, в котором биты с 8 по 5 первого байта установлены на значения от '0' до '9', первое, и возможно единственное, поле должно представлять собой идентификационный номер эмитента в соответствии с ИСО/МЭК 7812-1 [3].

Примечание — В ИСО/МЭК 7812-1:1993 идентификационный номер эмитента мог состоять из нечетного числа четырехразрядного байта, имеющего значения от '0' до '9'. Далее он быть преобразован в строку байтов путем установления бит с 4 по 1 последнего байта на 1111.

Если проприетарное расширение идентификатора приложения присутствует, то байт, установленный на 'FF' должен разделять два поля.

На рисунке D.1 показан AID, использующий идентификационный номер эмитента: он состоит из шестнадцати байтов, не более.

Идентификационный номер эмитента в соответствии с ИСО/МЭК 7812-1 [3] (два и более байтов)	'FF'	Проприетарное расширение идентификатора приложения (PIX)

Рисунок D.1 — AID, использующий идентификационный номер эмитента

[1)] Заменен на ИСО/МЭК 7816-5:2004.

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО/МЭК 7816-3	IDT	ГОСТ Р ИСО/МЭК 7816-3—2013 «Карты идентификационные. Карты на интегральных схемах. Часть 3. Карты с контактами. Электрический интерфейс и протоколы передачи»
ИСО/МЭК 7816-6	IDT	ГОСТ Р ИСО/МЭК 7816-6—2013 «Карты идентификационные. Карты на интегральных схемах. Часть 6. Межотраслевые элементы данных для обмена»
ИСО/МЭК 8825-1	IDT	ГОСТ Р ИСО/МЭК 8825-1—2003 «Информационная технология. Правила кодирования АСН.1. Часть 1. Спецификация базовых (BER), канонических (CER) и отличительных (DER) правил кодирования»

Примечание — В настоящей таблице использовано следующее условное обозначение степени соответствия стандартов:
- IDT — идентичные стандарты.

Библиография

[1]	ISO 3166-1:1997[1)]	Codes for the representation of names of countries and their subdivisions — Part 1: Country codes
[2]	ISO/IEC 7810:2003	Identification cards — Physical characteristics
[3]	ISO/IEC 7812-1:2000[2)]	Identification cards — Identification of issuers — Part 1: Numbering system
[4]	ISO/IEC 7816 (all parts)	Identification cards — Integrated circuit cards
[5]	ISO/IEC TR 9577:1999	Information technology — Protocol identification in the network layer
[6]	ISO/IEC 9795 (all parts)	Information technology — Security techniques — Digital signature schemes giving message recovery
[7]	ISO/IEC 9797 (all parts)	Information technology — Security techniques — Message Authentication Codes (MACs)
[8]	ISO/IEC 9798 (all parts)	Information technology — Security techniques — Entity authentication
[9]	ISO/IEC 9979:1999[3)]	Information technology — Security techniques — Procedures for the registration of cryptographic algorithms
[10]	ISO 9992-2:1998[4)]	Financial transaction cards — Messages between the integrated circuit card and the card accepting device— Part 2: Functions, messages (commands and responses), data elements and structures
[11]	ISO/IEC 10116:1997[5)]	Information technology — Security techniques — Modes of operation for an n-bit block cipher
[12]	ISO/IEC 10118 (all parts)	Information technology — Security techniques — Hash-functions
[13]	ISO/IEC 10536 (all parts)	Identification cards — Contactless integrated circuit(s) cards — Close-coupled cards
[14]	ISO/IEC 11770 (all parts)	Information technology — Security techniques — Key management
[15]	ISO/IEC 14443 (all parts)	Identification cards — Contactless integrated circuit(s) cards — Proximity cards
[16]	ISO/IEC 14888 (all parts)	Information technology — Security techniques — Digital signatures with appendix
[17]	ISO/IEC 15693 (all parts)	Identification cards — Contactless integrated circuit(s) cards — Vicinity cards
[18]	ISO/IEC 18033 (all parts)	Information technology — Security techniques — Encryption algorithms
[19]	IETF RFC 1738:1994	Uniform resource locators (URL)
[20]	IETF RFC 2396:1998	Uniform resource locators (URL): General syntax

---

[1)] Заменен на ИСО 3166-1:2013.

[2)] Заменен на ИСО/МЭК 7812-1:2006.

[3)] Отменен.

[4)] Отменен.

[5)] Заменен на ИСО/МЭК 10116:2006.

УДК 336.77:002:006.354 ОКС 35.240.15 Э46 ОКП 40 8470

Ключевые слова: обработка данных, обмен информацией, идентификационные карты. IC-карты, сообщения, способы защиты, аутентификация, команды

Редактор *Н. А. Аргунова*
Технический редактор *В. Н. Прусакова*
Корректор *С. В. Смирнова*
Компьютерная верстка *В. Н. Романовой*

Сдано в набор 11.08.2014. Подписано в печать 17.10.2014. Формат 60×84 1/8. Бумага офсетная. Гарнитура Ариал. Печать офсетная. Усл. печ. л. 11,63. Уч.-изд. л. 10,90. Тираж 42 экз. Зак. 1312.

ФГУП «СТАНДАРТИНФОРМ». 123995 Москва, Гранатный пер., 4.
www.gostinfo.ru info@gostinfo.ru
Набрано и отпечатано в Калужской типографии стандартов, 248021 Калуга, ул. Московская, 256.