УТВЕРЖДАЮ

Заместитель генерального директора

ФГУП «Гознак»

В.А. Барилкин

«___»_____2011г

УТВЕРЖДАЮ

Начальник Управления информационно-аналитических технологий

ФФОМС России

Ю.А. Нечепоренко

«___»_____2011г

СИСТЕМА ОБЕСПЕЧЕНИЯ ВЫПУСКА ПОЛИСОВ ОБЯЗАТЕЛЬНОГО МЕДИЦИНСКОГО СТРАХОВАНИЯ

ПРАВИЛА ФОРМИРОВАНИЯ ЭЛЕКТРОННОГО СТРАХОВОГО ПРИЛОЖЕНИЯ

СОГЛАСОВАНО

Начальник управления интеграции

ФГУП «Гознак»

М.Ю. Лучинкин

«___»_____2011г

СОГЛАСОВАНО

Заместитель начальника Управления информационно-аналитических технологий

ФФОМС России

Л.В. Котельникова

«___»_____2011г

Москва 2011

Оглавление

Аннотация

Настоящий документ содержит правила формирования электронного страхового приложения обязательного медицинского страхования (ЭСП ОМС).

В документе приводится:

- краткое описание назначения, а также принципов функционирования ЭСП ОМС;
- описание структур хранения данных, а также способов организации доступа к ним.

Список сокращений

AID	Application Identifier
BCD	Binary-coded decimal
BER	Basic encoding rules
DF	Dedicated file
EF	Elementary file
FCI	File control information
FCP	File control parameters
MF	Master File
ЗОС	Защищенный обмен сообщениями
ККС	Криптографическая контрольная сумма
ОГРН	Основной государственный регистрационный номер
ОКАТО	Общероссийский классификатор объектов административно-территориального деления
ОМС	Обязательное медицинское страхование
ПИН	Персональный идентификационный номер
СНИЛС	Страховой номер индивидуального лицевого счета
ЭП	Электронный полис
ЭЦП	Электронно-цифровая подпись

Введение

Электронный полис обязательного медицинского страхования (ОМС) представляет собой микропроцессорную карту, содержащую зафиксированную в электронной форме информацию о владельце карты, используемой для удостоверения права на бесплатное оказание медицинской помощи на всей территории Российской Федерации в объеме, предусмотренном базовой программой обязательного медицинского страхования.

В текущей спецификации описаны логические структуры и средства обеспечения безопасности при хранении данных в памяти электронного полиса ОМС.

В качестве базового стандарта, используемого при организации хранения информации и доступа к ней, используется серия стандартов ГОСТ Р ИСО/МЭК 7816.

Электрические характеристики электронного полиса ОМС, а также характеристики транспортного протокола взаимодействия с программно-аппаратными средствами соответствуют ГОСТ Р ИСО/МЭК 7816 часть 3. Микроконтроллер электронного полиса ОМС должен поддерживать два протокола передачи данных для контактного интерфейса: символьный протокол T=0 и блочный протокол T=1.

Формат команд электронного полиса ОМС определяется стандартом ГОСТ Р ИСО/МЭК 7816 часть 4.

Спецификация может дорабатываться в установленном порядке.

1. Структура электронного полиса ОМС

1.1. Общая структура электронного полиса ОМС

Электронный полис ОМС должен иметь файловую структуру, приведенную на рис. 1.

Данные, хранимые в памяти электронного полиса ОМС, разделяются на два типа:

- **Неизменяемые данные**: данные, которые не полежат изменению в процессе обращения полиса. В их состав входят идентификационные сведения о владельце полиса.
- **Изменяемые данные**: данные, которые могут изменяться и дополняться в процессе эксплуатации полиса. В их состав входят сведения о выбранной владельцем страховой медицинской организации и сведения о её смене.

Рис. 1. Структура хранения данных электронного полиса ОМС

Информация о производителе микросхемы хранится в файле EF.ICCID ID=0002.

Информация о эмитенте карты хранится в файле EF.CardID ID=0003.

Файлы, содержащие неизменяемые данные страхового приложения электронного полиса ОМС, размещены в приложении (директории) с идентификатором AID=46 4F 4D 53 5F 49 44.

Файлы, содержащие изменяемые данные страхового приложения электронного полиса ОМС, размещены в приложении (директории) с идентификатором AID=46 4F 4D 53 5F 49 4E 53.

1.2. Формат файла EF.ICCID

В данном файле хранится информация о микросхеме от производителя.

Таблица 1. Структура данных файла EF.ICCID

<table>
<tr><th>Тег</th><th>Длина</th><th colspan="3">Значение</th></tr>
<tr><td rowspan="4">60</td><td rowspan="4">X</td><td colspan="3">Данные о карте</td></tr>
<tr><td>Тег</td><td>Длина</td><td>Значение</td></tr>
<tr><td>41</td><td>1</td><td>Код производителя карты</td></tr>
<tr><td>42</td><td>X</td><td>Данные производителя карты</td></tr>
</table>

Условия доступа к файлу: только чтение.

Состав данных объекта с тегом 5F02 определяется производителем микросхемы по своему усмотрению.

1.3. Формат файла EF.CardID

В данном файле хранится информация о карте от эмитента.

Таблица 2. Структура данных файла EF.CardID

<table>
<tr><th>Тег</th><th>Длина</th><th colspan="3">Значение</th></tr>
<tr><td rowspan="4">61</td><td rowspan="4">X</td><td colspan="3">Данные о карте</td></tr>
<tr><td>Тег</td><td>Длина</td><td>Значение</td></tr>
<tr><td>51</td><td>X ≥ 8
байт</td><td>Уникальный номер карты</td></tr>
<tr><td>52</td><td>1</td><td>Тип карты: должно равняться 00.</td></tr>
</table>

				Зарезервировано на будущее.
		53	2	VV1 VV2 Версия спецификации, которой соответствует карта: VV1 — старшая часть версии VV2 — младшая часть версии
		54	X	Идентификатор учреждения, персонализирующего карточку
		55	X	Дополнительные сведения о карте, определяемые эмитентом.

Состав данных объекта с тегом 55 определяется эмитентом ЭП по своему усмотрению.

2. Электронное страховое приложение (неизменяемые данные)

Электронное страховое приложение (неизменяемые данные) содержит неизменяемые в процессе эксплуатации электронного полиса ОМС данные и обеспечивает однозначную идентификацию держателя полиса.

Неизменяемые данные страхового приложения должны быть записаны на этапе персонализации электронного полиса ОМС и не подлежат изменению в течение срока действия электронного полиса ОМС.

Неизменяемые данные страхового приложения электронного полиса ОМС, должны быть подписаны электронной цифровой подписью, сформированной уполномоченным лицом в процессе персонализации электронного полиса ОМС.

2.1. Общая структура приложения

Неизменяемые данные ЭП ОМС должны храниться в файловой структуре, приведенной на Рис. 2.

Рис. 2. Структура хранения данных в страховом приложении (неизменяемые данные)

Доступ к файлам должен быть организован в соответствии с таблицей 3: чтение данных доступно всегда, а запись — никогда.

Таблица 3. Состав файлов страхового приложения (неизменяемые файлы)

Файлы	**ID**	**Размер**	**Описание**	**Условия доступа**
EF	0201	X	Сведения о владельце	Чтение – всегда Запись – никогда
EF	0202	X	Данные безопасности	Чтение – всегда Запись – никогда

2.2. Сведения о владельце

Файл со сведениями о владельце содержит идентификационную информацию о владельце карты как участнике системы ОМС.

Сведения о владельце состоят из следующих полей.

Таблица 4. Состав сведений о владельце

Поле	М/О*	**Значение**
ПОЛИС_ОМС_НОМЕР	М	Номер электронного полиса ОМС

ОПРЕДЕЛИТЕЛЬ_ОСНОВНОЙ	М	Основной определитель имени владельца. Для граждан РФ — фамилия.
ОПРЕДЕЛИТЕЛЬ_ВТОРИЧНЫЙ	М	Вторичный определитель имени владельца. Для граждан РФ — имя.
ОПРЕДЕЛИТЕЛЬ_ОСТАЛЬНОЙ	М	Остальные определители. Для граждан РФ — отчество.
ПОЛ	М	Пол владельца
ДАТА_РОЖДЕНИЯ	М	Дата рождения владельца
ГРАЖДАНСТВО	О	Гражданство владельца
СНИЛС	О	Страховой номер индивидуального лицевого счета владельца, принятый в соответствии с законодательством РФ об индивидуальном (персонифицированном) учете в системе обязательного пенсионного страхования;
ДАТА_ОКОНЧАНИЯ	О	Дата окончания срока действия электронного полиса ОМС
МЕСТО_РОЖДЕНИЯ	О	Место рождения владельца
ДАТА_ИЗГОТОВЛЕНИЯ_ЭП	О	Дата изготовления электронного полиса в Центре персонализации
ФОТО	О	Фотография владельца полиса

*М/О — Обязательные (М)/Факультативные (О)

ОПРЕДЕЛИТЕЛЬ_ОСНОВНОЙ. Наиболее важный компонент имени владельца электронного полиса ОМС. Для граждан Российской Федерации указывается фамилия.

ОПРЕДЕЛИТЕЛЬ_ВТОРИЧНЫЙ. Наиболее важный компонент имени владельца электронного полиса ОМС после основного определителя. Для граждан Российской Федерации указывается имя.

ОПРЕДЕЛИТЕЛЬ_ОСТАЛЬНОЙ. Остальные части имени владельца электронного полиса ОМС. Для граждан Российской Федерации указывается отчество.

Поля ОПРЕДЕЛИТЕЛЬ_ОСНОВНОЙ, ОПРЕДЕЛИТЕЛЬ_ ВТОРИЧНЫЙ и ОПРЕДЕЛИТЕЛЬ_ОСТАЛЬНОЙ заполняются так же, как они заполнены в официальных документах, удостоверяющих личность владельца.

Сведения о владельце хранятся в элементарном бинарном файле ID=0201 и кодируются согласно следующей таблице.

Таблица 5. Формат хранения сведений о владельце

<table>
<tr><th>Тег</th><th>Длина</th><th colspan="4">Значение</th></tr>
<tr><td rowspan="11">62</td><td rowspan="11">X</td><td colspan="4">Информация о владельце</td></tr>
<tr><td>Тег</td><td>Длина</td><td colspan="2">Значение</td></tr>
<tr><td>5F26</td><td>X</td><td colspan="2">ПОЛИС_ОМС_НОМЕР</td></tr>
<tr><td>5F21</td><td>X</td><td colspan="2">ОПРЕДЕЛИТЕЛЬ_ОСНОВНОЙ</td></tr>
<tr><td>5F22</td><td>X</td><td colspan="2">ОПРЕДЕЛИТЕЛЬ_ВТОРИЧНЫЙ</td></tr>
<tr><td>5F23</td><td>X</td><td colspan="2">ОПРЕДЕЛИТЕЛЬ_ОСТАЛЬНОЙ</td></tr>
<tr><td>5F25</td><td>1</td><td colspan="2">ПОЛ</td></tr>
<tr><td>5F24</td><td>4</td><td colspan="2">ДАТА_РОЖДЕНИЯ</td></tr>
<tr><td>7F30</td><td>X</td><td colspan="2">ГРАЖДАНСТВО</td></tr>
<tr><td colspan="2"></td><td>Тег</td><td>Длина</td><td>Значение</td></tr>
<tr><td colspan="2"></td><td>5F31</td><td>3</td><td>Код страны</td></tr>
<tr><td colspan="2"></td><td>5F32</td><td>X</td><td>Кириллическое название страны</td></tr>
<tr><td>5F27</td><td>X</td><td colspan="2">СНИЛС</td></tr>
<tr><td>5F28</td><td>4</td><td colspan="2">ДАТА_ОКОНЧАНИЯ</td></tr>
</table>

<table>
<tr><td rowspan="7"></td><td rowspan="7"></td><td></td><td></td><td colspan="3"></td></tr>
<tr><td>5F29</td><td>X</td><td colspan="3">МЕСТО_РОЖДЕНИЯ</td></tr>
<tr><td>5F2A</td><td>4</td><td colspan="3">ДАТА_ИЗГОТОВЛЕНИЯ_ЭП</td></tr>
<tr><td rowspan="4">7F40</td><td rowspan="4">X</td><td colspan="3">ФОТО</td></tr>
<tr><td>Тег</td><td>Длина</td><td>Значение</td></tr>
<tr><td>5F41</td><td>1</td><td>Идентификатор формата кодирования данных</td></tr>
<tr><td>5F42</td><td>X</td><td>Данные изображения лица</td></tr>
</table>

2.3. Форматы представлений данных

2.3.1. Поле ОПРЕДЕЛИТЕЛЬ_ОСНОВНОЙ

Значение поля представляется в кодировке UTF-8.

2.3.2. Поле ОПРЕДЕЛИТЕЛЬ_ВТОРИЧНЫЙ

Значение поля представляется в кодировке UTF-8.

2.3.3. Поле ОПРЕДЕЛИТЕЛЬ_ОСТАЛЬНОЙ

Значение поля представляется в кодировке UTF-8.

2.3.4. Поле ПОЛ

Значение поля кодируется следующим способом:

- Значение 01 — Мужчина;
- Значение 02 — Женщина.

2.3.5. Поле ДАТА_РОЖДЕНИЯ

Значение поля представляется в формате ДМГГ (4 байта) в BCD кодировке.

2.3.6. Поле ГРАЖДАНСТВО

Код страны гражданства (тег 5F31) указывается согласно ГОСТ 7.67-2003 в виде трехбуквенного латинского кода в UTF-8 кодировке. Данные о коде страны могут отсутствовать, если страна не упомянута в стандарте ГОСТ 7.67-2003.

Название страны гражданства (тег 5F32) приводится на русском языке в кодировке UTF-8. Если код страны указан, то название должно соответствовать ГОСТ 7.67-2003.

2.3.7. Поле ПОЛИС_ОМС_НОМЕР

Значение поля приводится в ASCII кодировке.

2.3.8. Поле СНИЛС

Значение поля приводится в ASCII кодировке.

2.3.9. Поле ДАТА_ОКОНЧАНИЯ

Значение поля представляется в формате ДМГГ (4 байта) в BCD кодировке.

2.3.10. Поле МЕСТО_РОЖДЕНИЯ

Значение поля представляется в кодировке UTF-8.

2.3.11. Поле ДАТА_ИЗГОТОВЛЕНИЯ_ЭП

Значение поля представляется в формате ДМГГ (4 байта) в BCD кодировке.

2.3.12. Поле ФОТО

Идентификатор формата изображения лица (тег 5F41) может принимать следующие значения:

- 00 — JPEG
- 01 — JPEG2000
- 02-FF — *зарезервировано*

Данные фотографии владельца (тег 5F42) кодируются в соответствии с указанным форматом изображения лица.

2.4. Данные безопасности

Данные безопасности хранятся в элементарном бинарном файле ID=0202. Для хранения данных используется структура `SignedData`, определенная в RFC 3369. В структуре предусмотрена возможность хранения хеш-значения и ЭЦП для обеспечения контроля целостности произвольных данных, указанных в структуре `encapContentInfo`.

Таблица 6. Формат хранения данных безопасности

Тег	Длина	Значение
63	X	Структура `SignedData` (RFC 3369)

В файле должна храниться следующая уточненная структура `SignedData`:

```
SignedData ::= SEQUENCE {
        version INTEGER {v3(3)},
        digestAlgorithms DigestAlgorithmIdentifiers,
        encapContentInfo EncapsulatedContentInfo,
        certificate Certificate,
        signerInfos SignerInfos }

SignerInfos ::= SET OF SignerInfo

EncapsulatedContentInfo ::= SEQUENCE {
     eContentType        id-AtlasKard-OMS-SecurityObject,
     eContent            DataHash
}

id-AtlasKard ::= OBJECT IDENTIFIER { 1.2.643.2.56}
id-AtlasKard-OMS ::= OBJECT IDENTIFIER{ id-AtlasKard 100}
id-AtlasKard-OMS-SecurityObject ::= OBJECT IDENTIFIER{ id-
AtlasKard-OMS 10}
```

```
DataHash ::= SEQUENCE {
     hashAlgorithm  AlgorithmIdentifier,
     hashValue      OCTET STRING
}
```

В SignerInfos должна содержаться только одна структура SignerInfo.

Хеш-значение `hashValue` вычисляется для данных файла EF ID=0201 по алгоритму, указанному в поле `hashAlgorithm`.

Подробное описание остальных полей структуры SignedData приведено в RFC 3369.

3. Электронное страховое приложение (изменяемые данные)

Электронный полис ОМС обеспечивает возможность хранения в памяти микроконтроллера сведений о 10 случаях смены застрахованным лицом страховой медицинской организации.

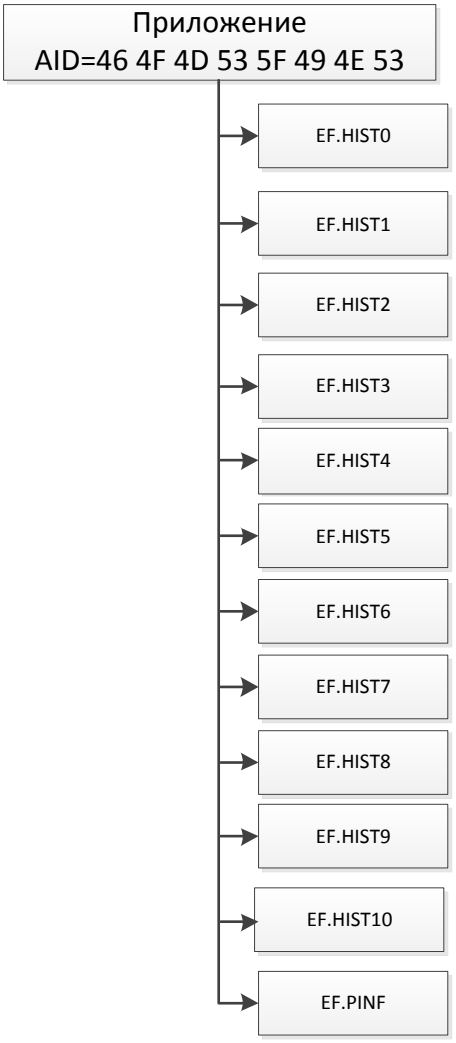

Рис. 3. Структура страхового приложения (изменяемые данные)

При каждой смене страховой медицинской организации в приложении должны быть отражены следующие данные:

- ОГРН страховой медицинской организации;
- ОКАТО субъекта Российской Федерации, на территории которого застрахован гражданин;
- Дата страхования;
- Дата окончания страхования

Дополнение информации электронного страхового приложения полиса ОМС осуществляется после заверения уполномоченным лицом соответствующих изменений с использованием электронной цифровой подписи. Внесение изменений осуществляется в режиме защищенного обмена сообщениями.

Повторная запись и изменение сохраненных данных невозможны.

3.1. Состав файлов страхового приложения (изменяемые данные)

Состав файлов страхового приложения электронного полис ОМС приведён в таблице 7.

Таблица 7. Состав файлов страхового приложения (изменяемые файлы)

Файлы		Размер	Описание
EF_{HIST0}	8010	2048b	Файл внесенного изменения 0
EF_{HIST1}	8011	2048b	Файл внесенного изменения 1
EF_{HIST2}	8012	2048b	Файл внесенного изменения 2
EF_{HIST3}	8013	2048b	Файл внесенного изменения 3
EF_{HIST4}	8014	2048b	Файл внесенного изменения 4
EF_{HIST5}	8015	2048b	Файл внесенного изменения 5
EF_{HIST6}	8016	2048b	Файл внесенного изменения 6
EF_{HIST7}	8017	2048b	Файл внесенного изменения 7
EF_{HIST8}	8018	2048b	Файл внесенного изменения 8
EF_{HIST9}	8019	2048b	Файл внесенного изменения 9
EF_{HIST10}	801A	2048b	Файл внесенного изменения 10
EF_{PINF}	0201	X	Файл сведений о электронном полисе

Файлы EF.HIST*i* являются бинарными файлами, предназначенными для хранения информации о СМО.

Каждый файл EF.HIST*i* может находиться в одном из трех состояний по отношению к хранимой в нем информации:

1. **Пустой файл** — в файле хранятся нулевые данные, т.е. запись о СМО отсутствует.
2. **Текущий файл** — в файле хранится запись о текущей СМО.
3. **Исторический файл** — в файле хранится запись о СМО, относящаяся к истории.

Права доступа к данным файла зависят от его состояния:

- Пустой файл:
  - Чтение: аутентификация на ключе СМО;
  - Запись: аутентификация и ЗОС на ключе СМО и предъявление ПИН.
- Текущий файл:
  - Чтение: всегда;
  - Запись: никогда.
- Исторический файл:
  - Чтение: ПИН или аутентификация на ключе ФОМС;
  - Запись никогда.

Считывание информации о текущей СМО осуществляется с помощью команды **GET DATA.** Данная команда возвращает двухбайтовый идентификатор файла с информацией о текущей СМО.

Внесение информации о новой текущей СМО осуществляется с помощью команды **PUT DATA**. При выполнении команды выполняются следующие действия:

1. Сохранение идентификатора нового файла с информацией о текущей СМО.
2. Изменение прав доступа к предыдущему файлу с текущей СМО в соответствии с правами доступа к историческому файлу.
3. Изменение прав доступа к новому файлу с текущей СМО в соответствии с правами доступа к текущего файла.

Таблица 8. Структура данных файлов EF.HISTi, i=0,2,…,10.

<table>
<tr><th>Тег</th><th>Длина</th><th colspan="4">Значение</th></tr>
<tr><td rowspan="8">64</td><td rowspan="8">X</td><td colspan="4">Данные о смене страховой медицинской организации</td></tr>
<tr><td>Тег</td><td>Длина</td><td colspan="2">Значение</td></tr>
<tr><td>5F51</td><td>13</td><td colspan="2">ОГРН страховой медицинской организации</td></tr>
<tr><td>5F52</td><td>32</td><td colspan="2">ОКАТО субъекта Российской Федерации, на территории которого застрахован гражданин</td></tr>
<tr><td>5F53</td><td>4</td><td colspan="2">Дата начала страхования</td></tr>
<tr><td>5F54</td><td>4</td><td colspan="2">Дата окончания страхования</td></tr>
<tr><td rowspan="3">7F60</td><td rowspan="3">X</td><td colspan="2">Сведения о заверении уполномоченным лицом</td></tr>
<tr><td>Тег</td><td>Длина</td><td>Значение</td></tr>
<tr><td>5F61</td><td>X</td><td>Значение ЭЦП</td></tr>
<tr><td>5F62</td><td>X</td><td>Сертификат уполномоченного лица</td></tr>
</table>

Файл EF.HIST0 заполняется при персонализации карты.

В файле EF_{PINF} хранится информация об электронном полисе. Доступ к данным должен быть организован следующим образом:

- Чтение — всегда;
- Запись — никогда.

Таблица 9. Структура данных файла EF.PINF

Тег	Длина	Значение
62	X	Информация об электронном полисе

		Тег	Длина	Значение
		5F26	X	ПОЛИС_ОМС_НОМЕР
		5F27	X	СНИЛС

3.2. Форматы представления данных

Цифровой номер ОГРН хранится в ASCII кодировке.

Цифровой номер ОКАТО хранится в ASCII кодировке.

Дата страхования и дата окончания страхования хранятся в формате ДМГГ (4 байта) в BCD кодировке.

Значение ЭЦП хранится в виде двоичных данных.

3.3. Внесение новых данных о страховой медицинской организации

Начальное заполнение файлов EF.HIST*i*, *i*=0,2,…,10 — нули.

Актуальным файлом с информацией о СМО является файл с идентификатором, который возвращает команда GET DATA.

Запись данных о новой страховой медицинской организации в файл производится в режиме ЗОС.

Актуальный файл с информацией о текущей СМО устанавливается командой PUT DATA.

4. Спецификация команд и функций карты

В данном разделе описывается минимальный набор команд, поддерживаемый микропроцессорной картой полиса ОМС. Все команды описаны в соответствии со стандартом ISO/IEC 7816-4.

Набор команд, необходимых для функционирования системы:

1. *Select File*
2. *Read binary*
3. *Update binary*
4. *Get Challenge*
5. *External authenticate*
6. *Mutual authenticate*
7. *Internal authenticate*
8. *Get response*
9. *Get Data*
10. *Put data*
11. *Verify*
12. *Reset Retry Counter*
13. *Security Check*

4.1. Базовый набор команд

В данном разделе описываются команды базового набора, необходимого для функционирования системы. Описаны только необходимые режимы и параметры команд. Спецификации могут быть расширены в рамках стандарта ISO/IEC 7816-4.

4.1.1. SELECT FILE

Данный раздел описывает минимальный набор требований к функционированию команды Select file.

Команда инициирует смену текущего файла (выбор файла).

- Выбор DF устанавливает только один текущий файл. Выбранный файл становится текущей директорией.
- Выбор EF устанавливает пару текущих файлов. Выбранный файл становится текущим EF, а его родительская директория становится текущей DF.

Далее следует описание минимального набора режимов функционирования команды. При реализации эти режимы могут быть расширены в соответствии со стандартом ISO/IEC 7816-4.

Командное сообщение

Таблица 10. Формат команды SELECT FILE

CLA	00h
INS	A4h
P1	См. текст ниже
P2	См. текст ниже
Lc	00h или длина данных
Поле данных	Если есть, то в соответствии с P1: • идентификатор файла • имя DF
Le	Длина ожидаемых данных

Таблица 11. Кодирование параметра P1

b8 … b5	b4	b3	b2	b1	Значение
0000	0	0	1	0	- Выбор EF под текущей DF (поле данных = идентификатор EF)
0000	0	1	0	0	- Выбор по имени DF (поле данных = имя DF)

P1=02 h – Выбор EF под текущей DF.

Команда завершится успешно, если под текущей DF есть EF с указанным в поле данных идентификатором.

P1=04 h – Выбор по имени DF.

Поле данных содержит имя DF, которое может быть усечено справа. Выбор по имени DF выполняется в двух режимах. Режим задается параметром P2 (См. Таблица)

Если выбор выполняется в режиме "первый или имеющийся в наличии", директория с указанным именем ищется под MF, поиск начинается непосредственно с MF. Заданное имя файла может быть как полным, так и усеченным.

Если выбор выполняется в режиме "следующий созданный", поиск осуществляется в родительской директории текущей DF. Поиск выполняется только под MF. Ищется файл директории с заданным именем, созданный после текущего файла DF.

Таблица 12. Кодирование параметра P2

P1	b8 … b5	b4	b3	b2	b1	Значение
04h	0000	-	-	0	0	– первый созданный или имеющийся в наличии
	0000	-	-	1	0	– следующий созданный
-	0000	X	X	-	-	Выбор контрольной информации файла
	0000	0	0	-	-	– Возврат FCP
	0000	1	1	-	-	– Ответные данные отсутствуют

Структура FCP

Данный раздел содержит минимальный набор информации, который должен содержаться в структуре FCP DF. При реализации системы этот набор может быть расширен.

Шаблон FCP DF должен соответствовать стандарту ISO/IEC 7816-4. Системой используется тэг «Данные приложения» данного шаблона для хранения версии текущего приложения.

Таблица 12 содержит часть шаблона FCP DF содержащую минимальный набор используемых дополнительных данных.

Таблица 13. Структура шаблона FCP DF

Значение		Описание	Длина
62h	+	Объект данных: FCP шаблон	1
Длина		Длина данных (С 3-го байта до конца)	1
A5h	+	Объект данных: Данные приложения	1
Длина		Длина данных приложения	1
XX…XX		Зависящие от приложения данные	XX
….		Другие объекты шаблона	…

Таблица 14. Данные приложения в шаблоне FCP DF

Значение		Описание	Длина
DF11h	+	Объект данных: Версия приложения	2
Длина		Длина версии приложения	1
XX…XX		Версия приложения	8

Ответное сообщение

Команда SELECT FILE в ответе может выдавать информацию о выбранном файле в шаблоне FCP или не выдавать никакой информации в соответствии с параметром P2.

Таблица 15. Формат ответа

Поле данных	FCP или пусто
SW1-SW2	Байты статуса

Таблица 16. Возвращаемые байты статуса

SW1-SW2(Hex)	Значение
61 XX	Команда выполнена, подготовлено XX байт ответа
62 83	Файл деактивирован
62 85	Файл в состоянии завершения
6A 82	Файл не найден
6B 00	Параметры P1, P2 некорректны
67 00	Недопустимое значение поля Lc
6C XX	Некорректная длина Le, XX - точная длина
6F 3E	Неизвестная ошибка

4.1.2. UPDATE BINARY

Данный раздел описывает минимальный набор требований к функционированию команды UPDATE BINARY.

Команда UPDATE BINARY инициирует запись данных в EF с неструктурированными данными.

Данные, хранящиеся в файле, заменяются данными, пришедшими в поле данных команды. Запись данных начинается со смещением, указанным в P1-P2.

Команда выполняется на текущем файле. Файл может быть предварительно выбран с помощью команды SELECT FILE.

Далее следует описание минимального набора режимов функционирования команды. При реализации эти режимы могут быть расширены в соответствии со стандартом ISO/IEC 7816-4.

Командное сообщение

Таблица 17. Формат команды UPDATE BINARY

CLA	00h или 0Ch
INS	D6h
P1-P2	См. текст ниже
Lc	Длина следующего поля данных
Поле данных	Строка байт для записи
Le	Пусто

- В P1 b8=0.
- P1||P2 это смещение первого байта для записи в байтах от начала файла. Причем P1 – старший байт, а P2 – младший.

Ответное сообщение

Таблица 18. Формат ответа

Поле данных SW1-SW2	Пусто Байты состояния

Таблица 19. Возвращаемые байты статуса

SW1-SW2(HEX)	Значение
90 00	Команда выполнена
67 00	Недопустимое значение поля Lc (не возможно записать Lc байт с заданного смещения)
69 81	Команда не совместима со структурой файла (текущий EF не бинарный)
69 86	Текущий файл не EF
6A 82	Файл не найден
6A 84	Свободной памяти недостаточно (для создания резервной копии)
6B 00	Параметры P1, P2 некорректны (смещение вне файла)
69 00	Команда не разрешена (запрещена значением LCSI файла)
6F 3E	Неизвестная ошибка

4.1.3. READ BINARY

Данный раздел описывает минимальный набор требований к функционированию команды READ BINARY.

В ответ на команду READ BINARY карта выдает часть или все содержимое бинарного файла EF.

Команда выполняется на текущем файле. Файл может быть предварительно выбран с помощью команды SELECT FILE.

Чтение данных начинается со смещением, указанным в P1-P2. В поле данных ответа передаются неформатированные данные.

Далее следует описание минимального набора режимов функционирования команды. При реализации эти режимы могут быть расширены в соответствии со стандартом ISO/IEC 7816-4.

Командное сообщение

Таблица 20. Формат команды READ BINARY

CLA	00h
INS	B0h
P1-P2	См. текст ниже
Lc	Пусто
Поле данных	Пусто
Le	Количество байт для чтения

- В P1 b8=0.
- P1||P2 это смещение первого байта для чтения в байтах от начала файла. Причем P1 – старший байт, а P2 – младший.

Ответное сообщение

Если поле Le содержит одни нули, то в ответ карта выдает 256 байтов данных. Если данных меньше, то карта выдает предупреждение '61 La', где La – действительная длина данных.

Таблица 21. Формат ответа

Поле данных SW1-SW2	Прочитанные данные (Le байт) Байты состояния

Таблица 22. Возвращаемые байты статуса

SW1-SW2(Hex)	Значение
90 00	Команда выполнена
6B 00	Параметры P1, P2 некорректны (смещение вне файла)
61 XX	Команда выполнена, подготовлено XX байт ответа (Le =0)
6C XX	Некорректное значение поля Le, Le> длины доступных байт, XX - размер длины доступных байт
69 81	Команда не совместима со структурой файла (Текущий файл не бинарный)
69 86	Текущий файл не EF
6A 82	Файл не найден (при выборе по короткому идентификатору)
69 00	Команда не разрешена (запрещена значением LCSI файла)
6F 3E	Неизвестная ошибка

4.1.4. GET CHALLENGE

Данный раздел описывает минимальный набор требований к функционированию команды GET CHALLENGE.

Команда GET CHALLENGE инициирует выдачу случайного числа карты для использования в процедурах, ориентированных на обеспечение безопасности (например, в команде EXTERNAL AUTENTICATE).

Выданное случайное число действительно как минимум для следующей команды.

Далее следует описание минимального набора режимов функционирования команды. При реализации эти режимы могут быть расширены в соответствии со стандартом ISO/IEC 7816-4.

Командное сообщение

Таблица 23. Формат команды GET CHALLENGE

CLA	00h
INS	84h
P1-P2	0000h
Lc	Пусто
Поле данных	Пусто
Le	Длина ожидаемого ответа 08h -F0h (кратна 8-ми)

Ответное сообщение

Таблица 24. Формат ответа

Поле данных SW1-SW2	Случайное число Байты состояния

Таблица 25. Возвращаемые байты состояния

SW1-SW2(Hex)	Значение
90 00	Команда выполнена
61 10	Команда выполнена, подготовлено 16 байтов ответа (Le=0)
67 00	Недопустимое значение поля Le
6B 00	Некорректный параметр P1/ P2
6F 3E	Неизвестная ошибка

4.1.5. EXTERNAL AUTHENTICATE, MUTUAL AUTHENTICATE

Данный раздел описывает минимальный набор требований к функционированию команд EXTERNAL AUTHENTICATE и MUTUAL AUTHENTICATE.

Далее следует описание минимального набора режимов функционирования команды. При реализации эти режимы могут быть расширены в соответствии со стандартом ISO/IEC 7816-4.

Команда обновляет статус безопасности. В результате успешного исполнения команды подтверждается подлинность внешнего устройства и в зависимости от варианта исполнения подтверждается подлинность карты, и вырабатываются сеансовые ключи.

Поддерживается два варианта выполнения команды:

- EXTERNAL AUTHENTICATE – внешняя аутентификация.
- MUTUAL AUTHENTICATE – взаимная аутентификация с генерацией сеансовых ключей по протоколу взаимной аутентификации карты и модуля безопасности терминального оборудования с использованием программного модуля Ф.

Вариант исполнения команды выбирается в зависимости от размера поля данных и параметров команды.

EXTERNAL AUTHENTICATE

Если длина входных данных равна 8 или 6 байтам, выполняется команда EXTERNAL AUTHENTICATE.

В процессе исполнения сравнивается результат (да или нет) вычислений, сделанных картой, основанных на случайном числе, выданном картой ранее в ответ на команду GET CHALLENGE и ключе, хранящемся в карте, с данными аутентификации, переданными внешним устройством.

Ключ необходимый для вычислений определяется параметром P2.

MUTUAL AUTHENTICATE

Если длина входных данных равна 22 байтам и P2 = 0, выполняется команда MUTUAL AUTHENTICATE. В процессе исполнения команды, карта и внешнее устройство обмениваются криптограммами, что позволяет выполнить взаимное установление подлинности в одной команде.

Вычисления данных аутентификации и сеансовых ключей выполняются по специальным алгоритмам с использованием составного случайного числа, сформированного из 16-байтового случайного числа карты, выработанного в ответ на команду GET CHALLENGE и 16-байтового случайного числа, выработанного внешним устройством и переданного в поле данных команды.

После успешного проведения аутентификации вычисляется один сеансовый ключ, который используется для шифрования/расшифровывания данных и для вычисления криптографической контрольной суммы при обмене данными по каналу безопасной связи.

В результате успешного исполнения команды подтверждается подлинность карты и внешнего устройства, вырабатываются сеансовые ключи и открывается канал безопасной связи.

Командное сообщение

Таблица 26. Формат команды EXTERNAL AUTHENTICATE

CLA	00h
INS	82h
P1	00h
P2	Ссылка на ключ: 01h — ключ СМО 02h — ключ ФОМС
Lc	06/08h
Поле данных	Зашифрованное случайное число
Le	Пусто

Таблица 27. Формат команды MUTUAL AUTHENTICATE

CLA	00h
INS	82h
P1	00h
P2	00h
Lc	16h
Поле данных	16-байтовое случайное число ТО + криптограмма
Le	06h

Ответное сообщение

Таблица 28.Формат ответа

Поле данных	EXTERNAL –пусто MUTUAL – зашифрованное случайное число
SW1-SW2	Байты состояния

Таблица 29. Возвращаемые байты статуса

SW1-SW2(Hex)	Значение
90 00	Команда выполнена
61 08	Команда выполнена, подготовлено 8 байт ответа
63 CX	Аутентификации не выполнена. Осталось X попыток
63 00	Аутентификации не выполнена. Ключ заблокирован
67 00	Недопустимое значение поля Lc
6A 88	Ссылочные данные не найдены
69 84	Ссылочные данные некорректны
69 85	Файл ключей не активирован. Не было команды GET CHALLENGE
6F 3E	Неизвестная ошибка

4.1.6. INTERNAL AUTHENTICATE

Данный раздел описывает минимальный набор требований к функционированию команды INTERNAL AUTHENTICATE.

Команда INTERNAL AUTHENTICATE требует от карты выдачи данных аутентификации (криптограммы), вычисленных с использованием случайного числа, переданного в команде и ключа, хранящегося в карте. Ключ необходимый для вычислений определяется параметром P2

Далее следует описание минимального набора режимов функционирования команды. При реализации эти режимы могут быть расширены в соответствии со стандартом ISO/IEC 7816-4.

Командное сообщение

Таблица 30. Формат команды INTERNAL AUTHENTICATE

CLA	00h
INS	88h
P1	00h
P2	Ссылка на ключ
Lc	08h
Поле данных	Случайное число ТО
Le	06h

Ответное сообщение

Таблица 31. Формат ответа

Поле данных SW1-SW2	Криптограмма Байты состояния

Таблица 31. Возвращаемые байты статуса

SW1-SW2(Hex)	Значение
90 00	Команда выполнена
67 00	Недопустимое значение поля Lc, Le
6A 88	Ссылочные данные не найдены
69 84	Ссылочные данные некорректны
69 85	файл ключей не активирован
6F 3E	Неизвестная ошибка

4.1.7. GET DATA

Данный раздел описывает минимальный набор требований к функционированию команды GET DATA.

Команда позволяет получить значение двухбайтового идентификатора файла с записью о текущей СМО.

Командное сообщение

Таблица 33 Формат команды GET DATA

CLA	00h
INS	CAh
P1	01h
P2	B0h
Lc	—
Поле данных	—
Le	02h

Ответное сообщение

Таблица 34. Формат ответа

Поле данных SW1-SW2	XX XX — идентификатор файла Байты статуса

Таблица 35. Возвращаемые байты статуса

SW1-SW2(Hex)	Значение
90 00	Команда выполнена
67 00	Недопустимое значение поля Lc
6B 00	Параметры P1, P2 некорректны
6D 00	Код команды не идентифицирован
69 00	Команда не разрешена (не для текущей фазы жизни)
69 82	Условия доступа не удовлетворены
6A 82	Файл не найден
6F xx	Системная ошибка, xx – код ошибки

4.1.8. PUT DATA

Данный раздел описывает минимальный набор требований к функционированию команды PUT DATA.

Команда позволяет выполнить некоторые процедуры, перечень которых приведен в Таблица 33. В P2 указывается код выполняемой процедуры.

Для выполнения данных процедур должны быть удовлетворены условия: аутентификация на ключе СМО.

Процедуры могут быть дополнены при реализации системы.

Командное сообщение

Таблица 36 Формат команды PUT DATA

CLA	00h
INS	DAh
P1	01h
P2	Код процедуры См. Таблица ниже
Lc	Длина поля данных
Поле данных	Данные
Le	Пусто

Таблица 37. Кодирование P2 и длина данных

P2	Функция	Lc(байт)
B0h	WRITE CURRENT IMO	2

P2=B0h WRITE CURRENT IMO

Процедура WRITE CURRENT IMO позволяет внести изменение о текущей СМО.

Таблица 38. Формат поля данных

Описание	Длина
Двухбайтовый идентификатор файла	2

Ответное сообщение

Таблица 39. Формат ответа

Поле данных SW1-SW2	Пусто Байты статуса

Таблица 40. Возвращаемые байты статуса

SW1-SW2(Hex)	Значение
90 00	Команда выполнена
67 00	Недопустимое значение поля Lc
6B 00	Параметры P1, P2 некорректны
6D 00	Код команды не идентифицирован
69 00	Команда не разрешена (не для текущей фазы жизни)
69 82	Условия доступа не удовлетворены
6A 82	Файл не найден
6F xx	Системная ошибка, xx – код ошибки
6F 3E	Неизвестная ошибка (возможно Lc не равно длине объекта данных ATR или его нет)

4.1.9. VERIFY

Данный раздел описывает минимальный набор требований к функционированию команды VERIFY.

Команда позволяет предъявить персональный идентификационный номер (ПИН).

Таблица 41 Формат команды VERIFY

CLA	00h
INS	20h
P1	00h
P2	01h
Lc	Длина поля данных
Поле данных	Значение ПИН
Le	Пусто

Ответное сообщение

Таблица 42. Формат ответа

Поле данных SW1-SW2	Пусто Байты статуса

Таблица 43. Возвращаемые байты статуса

SW1-SW2(Hex)	Значение
90 00	Команда выполнена
67 00	Недопустимое значение поля Lc
6A 88	Ссылочные данные не найдены
69 84	Ссылочные данные некорректны
63 CX	Предъявлен неверный пароль. Осталось X попыток.
63 83	Пароль заблокирован
6F 3E	Неизвестная ошибка

4.1.10. RESET RETRY COUNTER

Данный раздел описывает минимальный набор требований к функционированию команды RESET RETRY COUNTER.

Команда позволяет разблокировать персональный идентификационный номер, если он был заблокирован ранее, записать новое значение ПИН и установить его счетчик оставшихся попыток в исходное значение.

Таблица 44 Формат команды RESET RETRY COUNTER

CLA	00h
INS	2Ch
P1	00h
P2	01h
Lc	Длина поля данных
Поле данных	Код разблокировки ПИН + новое значение ПИН
Le	Пусто

Ответное сообщение

Таблица 45. Формат ответа

Поле данных SW1-SW2	Пусто Байты статуса

Таблица 46. Возвращаемые байты статуса

SW1-SW2(Hex)	Значение
90 00	Команда выполнена
67 00	Недопустимое значение поля Lc
6A 88	Ссылочные данные не найдены
69 84	Ссылочные данные некорректны
63 CX	Предъявлен неверный код разблокировки. Осталось X попыток.
63 83	Код разблокировки заблокирован
6F 3E	Неизвестная ошибка

4.1.11. SECURITY CHECK

Данный раздел описывает минимальный набор требований к функционированию команды SECURITY CHECK.

Команда выполняет вычисление криптографической контрольной суммы программного кода библиотеки криптографического ядра микроконтроллера карты. В качестве криптографической контрольной суммы используется имитовставка вычисленная по алгоритму ГОСТ28147-89.

Таблица 47 Формат команды SECURITY CHECK

CLA	00h
INS	52h
P1	00h
P2	00h
Lc	20h
Поле данных	Криптографический ключ для вычисления контрольной суммы
Le	4h

Ответное сообщение

Таблица 48. Формат ответа

Поле данных SW1-SW2	Криптографическая контрольная сумма Байты статуса

Таблица 49. Возвращаемые байты статуса

SW1-SW2(Hex)	Значение
90 00	Команда выполнена
67 00	Недопустимое значение поля Lc
6F 3E	Неизвестная ошибка

5. Механизмы системы безопасности

5.1. Защищенный обмен сообщениями

Данный раздел описывает минимальные требования к организации защищенного обмена сообщениями.

Защищенный обмен сообщениями (ЗОС) используется для обеспечения достоверности и конфиденциальности передаваемых данных.

Сообщения, передаваемые в режиме защищенного обмена, всегда удостоверяются криптографической контрольной суммой (ККС). Данные в сообщении могут передаваться как в зашифрованном, так и в открытом виде.

Режим обмена сообщениями, защищенный или незащищенный, и алгоритм формирования данных для вычисления крипто определяется классом команды.

Кодировка байта класса, при котором команда выполняется в режиме защищенного обмена сообщений, представлена в Таблице 39.

Таблица 50. Значение байта класса для команд, выполняемых в режиме ЗОС

CLA	Определение
xxxx 11xxb	Команда выполняется в режиме ЗОС Заголовок участвует в формировании MAC
xxxx 10xxb	Команда выполняется в режиме ЗОС. Заголовок не участвует в формировании MAC
xxxx 00xxb	Команда выполняются без ЗОС

5.1.1. Используемые криптографические алгоритмы

При формировании криптограммы в режиме защищенного обмена сообщениями используются следующие алгоритмы:

1. Для шифрования используется ГОСТ 28147-89 в режиме простой замены с зацеплением.
2. Для криптографической контрольной суммы используется алгоритм ГОСТ 28147-89 в режиме вычисления имитовставки.
3. Данные, участвующие в криптографических вычислениях, дополняются до кратности 8 в соответствии с ISO/IEC 9797-1 метод 2 (M2): “80 00 00…” (данные дополняются до кратности 8, даже если они уже кратны 8).

5.1.2. Структура защищенных сообщений

В режиме ЗОС сообщения передаются в особом формате. Поле данных сообщений состоит из BER-TLV объектов ЗОС. Состав BER-TLV объектов определяется политикой безопасности и направлением передачи данных.

5.1.3. Объекты поля данных защищенных сообщений

Для передачи данных в поле данных команды и ответа используются контекстно-зависимые BER-TLV объекты. Имя объекта соответствует его содержанию. Для передачи ККС используется объект с именем 8E, для передачи байтов статуса завершения команды объект с именем 99 и так далее. Объекты данных ЗОС, поддерживаемые ОС представлены в нижеследующей Таблица .

Таблица 51. Объекты поля данных ЗОС

Имя(Hex)	Длина(Hex)	Описание	Направление
81	XX	Открытые данные	Поле данных команды/ответа
87	XX	Зашифрованные данные вместе с индикатором типа дополнения	Поле данных команды/ответа
97	01	Le	Поле данных команды
99	02	SW1-2 команды	Поле данных ответа
8E	04/08	ККС	Поле данных команды/ответа

Объект данных 87 имеет формат, представленный на рисунке.

87	XX	01	Зашифрованные данные

Рисунок 4. Формат объекта данных 87

01– индикатор типа дополнения; дополнение в соответствии с ISO/IEC9797-1 метод 2: “80 00 00…”.

Структура модуля данных команды в режиме ЗОС

Если при незащищенном обмене возможны четыре варианта структуры Формат команды, то при защищенном обмене сообщениями все варианты преобразуются в один, с типом направления передачи данных case 4. При этом байт Le' всегда равен нулю. Поле данных команды при защищенном обмене сообщениями состоит из BER-TLV кодированных объектов модифицированного поля данных команды (Поле данных') и ККС. Объект ККС добавляется в конец модифицированного поля данных команды. Формат команды представлен на рисунке 5. Примеры структуры полей данных команды представлены в Таблице 38.

Общий формат команды

Заголовок команды				Данные команды		
CLA	INS	P1	P2	Lc	Поле данных	Le

Формат команды при защищенном обмене сообщениями

Заголовок команды				Данные команды			
CLA	INS	P1	P2	L'c	Поле данных'	ККС	L'e='00'

Пример структуры поля данных команды при защищенном обмене сообщениями:

$T_D \| L_D \|$ Данные $\| 97 \| 01 \| Le \| 8E \| 06 \|$ ККС

Поле данных'

L'c

Где

$T_D \| L_D \|$ Данные – объект данных 81/87

L_D = Lc, если передаются простые данные

L_D = Lc + Lp+1, если передаются зашифрованные данные, Lp – длина дополнения

Рисунок 5. Формат команды

Таблица 52. Примеры модифицированного поля данных команды в режиме ЗОС

Направление	Поле данных команды
CASE 1	8E‖06‖ ККС
CASE 2	97‖L‖Le‖8E‖06‖ ККС или 8E‖06‖ ККС, если Le=0
CASE 3	81‖Lc‖Данные‖8E‖06‖ ККС
	87‖Lc+Lp+1‖01‖ENC[Данные]‖8E‖06‖ ККС
CASE 4	81‖Lc‖Данные‖97‖L‖Le‖8E‖06‖ ККС
	87‖Lc+Lp+1‖01‖ENC[Данные]‖97‖L‖Le‖8E‖06‖ ККС

Структура модуля данных ответа в режиме ЗОС

Структура ответа при безопасном обмене сообщениями зависит от состояния завершения команды. В случае успешного завершения поле данных ответа состоит из BER-TLV кодированных объектов модифицированного поля данных ответа (Поле данных') и ККС. Формат ответа представлен на рисунке. Примеры структуры полей данных ответа представлены в Таблица 39.

Общий формат ответа

Поле данных	SW1	SW2

Формат ответа при защищенном обмене сообщениями

Поле данных'	ККС	90	00

Пример структуры поля данных ответа при защищенном обмене сообщениями:

$T_D \| L_D \|$ Данные $\| 99 \| 02 \|$ SW1,SW2 $\| 8E \| 06 \|$ ККС

Поле данных'

или если данные отсутствуют (при успешном завершении или ошибке исполнения команды):

$99 \| 02 \|$ SW1,SW2 $\| 8E \| 06 \|$ ККС

Поле данных'

Где

$T_D \| L_D \|$ Данные – объект данных 81/87, содержащий открытые или зашифрованные данные с индикатором дополнения

L_D = La, если передаются открытые данные(La- длина подготовленных данных)

L_D = La + Lp +1, если передаются зашифрованные данные, Lp – длина дополнения

Рисунок 6. Формат ответа

Таблица 53. Примеры структуры поля данных ответа в режиме ЗОС

Направление	Поле данных ответа
Case1	99 ‖ 02 ‖ SW1,2 ‖ 8E ‖ 06 ‖ ККС
Case2	81 ‖ Le ‖ Данные ‖ 99 ‖ 02 ‖ SW1,2 ‖ 8E ‖ 06 ‖ ККС
	87 ‖ Le+Lp +1 ‖ ENC[Данные] ‖ 99 ‖ 02 ‖ SW1,2 ‖ 8E ‖ 06 ‖ ККС
Case3	99 ‖ 02 ‖ SW1,2 ‖ 8E ‖ 06 ‖ ККС
Case4	81 ‖ Le ‖ Данные ‖ 99 ‖ 02 ‖ SW1,2 ‖ 8E ‖ 06 ‖ ККС
	87 ‖ Le+Lp +1 ‖ ENC[Данные] ‖ 99 ‖ 02 ‖ SW1,2 ‖ 8E ‖ 06 ‖ ККС

5.1.4. Вычисление криптографической контрольной суммы

Если команда передается в режиме ЗОС, то присутствие ККС под сообщениями команды и ответа обязательно. В подсчет ККС входят все нечетные объекты данных ЗОС присутствующие в поле данных команды или ответа.

ККС **под сообщением команды** вычисляется на следующих данных:

– инкрементированное случайное число-счетчик (SSC+1);

– четыре байта заголовка (CLA, INS, P1, P2), дополненные до кратности 8;
– модифицированное поле данных команды дополненное до кратности 8.

Участие заголовка в формировании ККС является не обязательным и определяется битом 3 класса команды. Установка этого бита в 1 говорит об участии заголовка в формировании ККС.

Состав данных, участвующих в вычислении ККС под сообщением команды CLA=XCh, представлен на Рисунок 7 .

Инкрементированное случайное число-счетчик (SSC+1)	+
Заголовок команды (CLA, INS, P1, P2)	+
Дополнение	+
Имя, длина и данные объекта данных 81 или 87	+
Имя, длина и данные объекта данных 97	+
Дополнение	

Рисунок 7. Данные для вычисления криптографической контрольной суммы под сообщением команды

Состав данных, участвующих в вычислении криптографической контрольной суммы под сообщением команды CLA=X8h, представлен на Рисунок 8 .

Инкрементированное случайное число-счетчик (SSC+1)	+
Имя, длина и данные объекта данных 81 или 87	+
Имя, длина и данные объекта данных 97	+
Дополнение	

Рисунок 8. Данные для вычисления криптографической контрольной суммы под сообщением команды

ККС обязательно должна быть вычислена на случайном числе. Участие объектов данных 81, 87 или 97 в вычислении ККС определяется их наличием в поле данных сообщения команды.

Криптографическая контрольная сумма под сообщением ответа вычисляется на следующих данных:

– инкрементированное случайное число-счетчик (SSC+1);

– модифицированное поле данных ответа дополненное до кратности 8.

Состав данных, участвующих в вычислении ККС под сообщением ответа, представлен на Рисунок 9.

Инкрементированное случайное число-счетчик (SSC+1)	||
Имя, длина и данные объекта данных 81 или 87	||
Имя, длина и данные объекта данных 99	||
Дополнение	

Рисунок 9. Данные для вычисления криптографической контрольной суммы под сообщением ответа

ККС обязательно вычисляется на случайное число и объект данных содержащий байты статуса завершения команды. Участие объекта данных 81 или 87 в вычислении криптографической контрольной суммы определяется их наличием в поле данных ответного сообщении.

5.1.5. Схема формирования сообщений команды и ответа

Реализованная схема модификации поля данных сообщений и формирования данных для вычисления криптографической ККС позволяет вести обмен защищенными сообщениями. Схемы формирования сообщений команды и ответа представлены на ниже следующих рисунках.

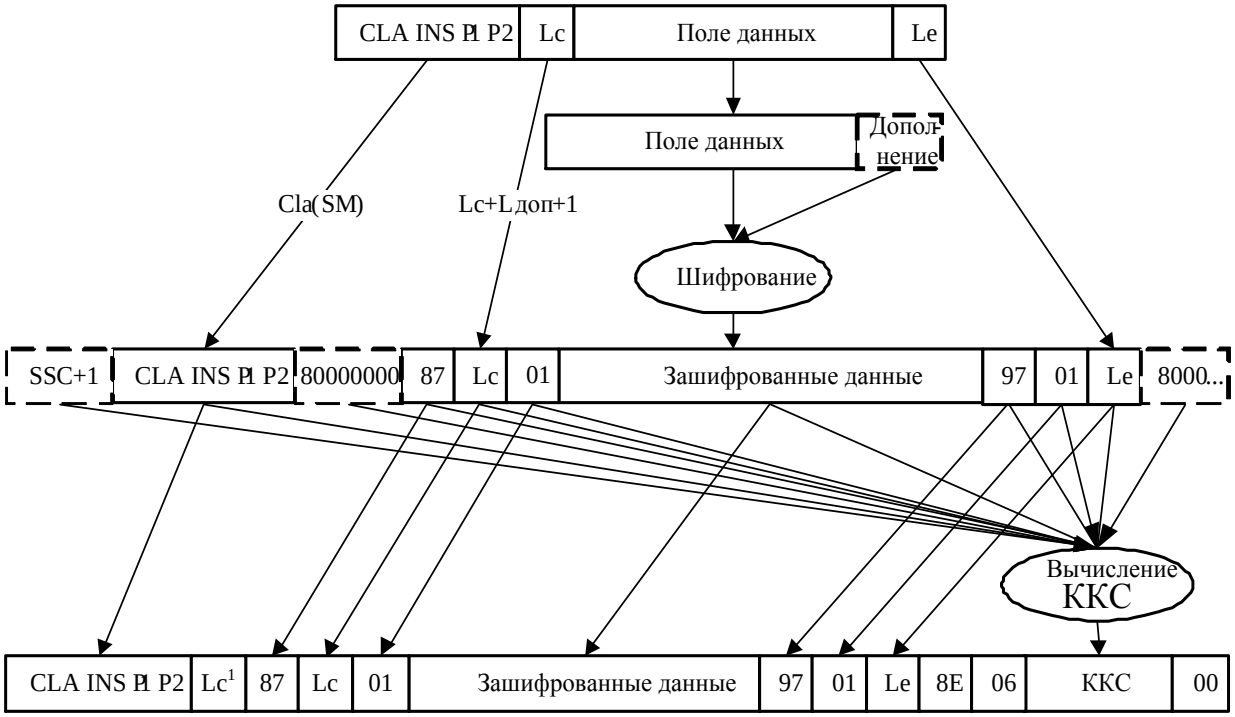

Рисунок 10. Схема формирования сообщения команды

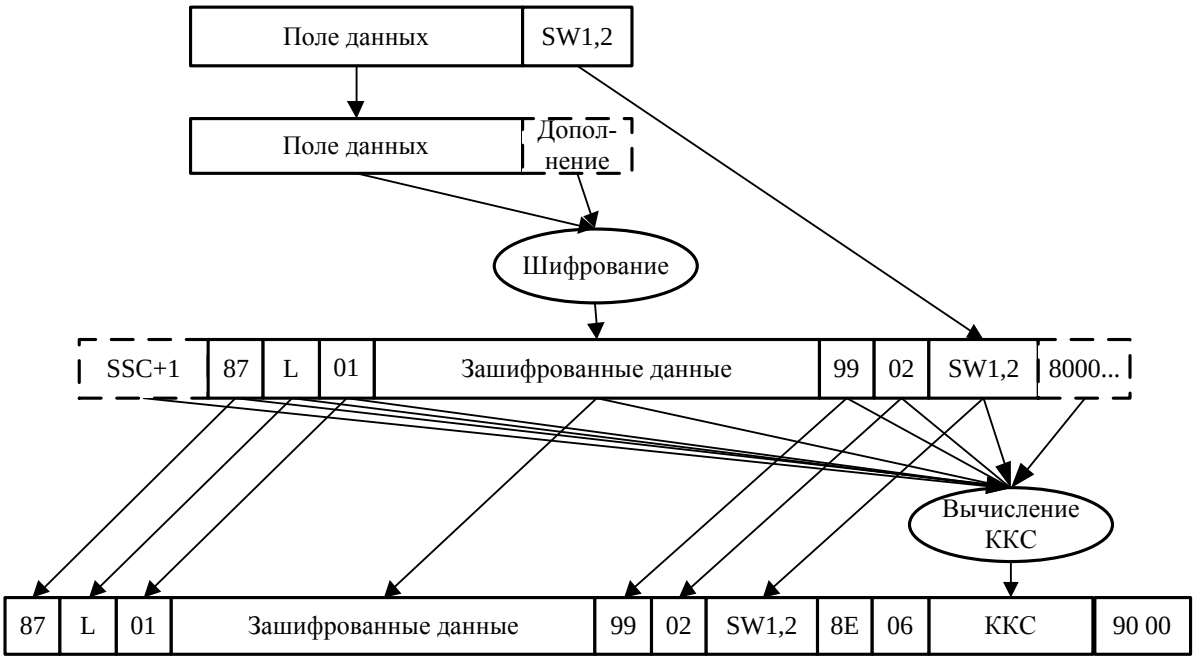

Рисунок 11. Схема формирования сообщения ответа

5.2. Аутентификация

В этом разделе описаны процедуры аутентификации, используемые в системе.

5.2.1. Внешняя аутентификация

Функция внешней аутентификации реализуется посредством выполнения следующего протокола:

Исходные данные: карты и терминала хранят ключ К.

Результат: карта принимается решение о том, что терминал признаётся или не признаётся корректно аутентифицированным.

Таблица 54. Протокол внешней аутентификации

Действия, выполняемые картой	Команды и данные в канале передачи данных	Действия, выполняемые терминалом
R=Ф2(1)	⇦ GET_CHALLENGE(Le=8)	
	⇨ R	
		Q= Ф1(K,R,1)
	⇦ EXTERNAL__AUTHENTICATE в режиме односторонней аутентификации (Lc=6)	
	⇦ младшие 6 байтов Q	
Q'= Ф1(K,R,1) результат: младшие 6 байтов Q равны младшим 6-ти байтам Q'		

Описание функций протокола:

- Ф1(K,T,L) — Функция зашифрования в режиме простой замены ГОСТ 28147-89 открытого текста T размером L блоков на ключе K.
- Ф2(L) — Функция выработки случайного числа размером L блоков

5.2.2. Внутренняя аутентификация

Функция внутренней аутентификации реализуется посредством выполнения следующего протокола.

Исходные данные: карта и терминал хранят ключ K.

Результат: ВУ принимается решение о том, что карта признаётся или не признаётся корректно аутентифицированной.

Таблица 55. Протокол внутренней аутентификации

Действия, выполняемые картой	Команды и данные в канале передачи данных	Действия, выполняемые терминалом
		R=Ф2(1)
	⇦ INTERNAL__AUTHENTICATE (Lc=8, Le=6)	
	⇦ R	
Q= Ф1(K,R,1)	⇨ младшие 6 байтов Q	Q'= Ф1(K,R,1) результат: младшие 6 байтов Q равны младшим 6-ти байтам Q'

Описание функций протокола:

- Ф1(K,T,L) — Функция зашифровывания в режиме простой замены ГОСТ 28147-89 открытого текста T размером L блоков на ключе K.
- Ф2(L) — Функция выработки случайного числа размером L блоков

5.2.3. Взаимная аутентификация

Исходными данными для протокола взаимной аутентификации между интеллектуальной картой (ИК) и терминалом (Т) является исходный ключ К.

Результат:

- Т принимается решение о том, что ИК признаётся или не признаётся корректно аутентифицированной;
- К принимается решение о том, что Т признаётся или не признаётся корректно аутентифицированным;
- ИК и Т вырабатывают сеансовый ключ К2 (256 бит).

Таблица 56. Протокол взаимной аутентификации

Действия, выполняемые картой	Команды и данные в канале передачи данных	Действия, выполняемые терминалом
(R’,G’)=Ф2(2)	⇦ GET_CHALLENGE(Le=16)	
	⇨ R’,G’	
		(R,G)=Ф2(2) V'=R'⊕G' V=R⊕G R1=Ф8(R,G, R’,G’) K1= Ф1(K,R1,4) K2= Ф1(K1,K,4) Q= Ф1(K1,V’,1) U'= Ф1(K1,V,1)
	⇦ MUTUAL AUTHENTICATE (Lc=22, Le=6)	
	⇦ R,G, младшие 6 байтов Q	
V=R⊕G V'=R'⊕G' R1=Ф7(R’,G’,R,G) K1= Ф1(K,R1,4) K2= Ф1(K1,K,4) U= Ф1(K1,V,1) Q'= Ф1(K1,V',1)		

результат: младшие 6 байтов Q равны младшим 6-ти байтам Q' и V≠V’		
	⇨ младшие 6 байтов U	
		результат: младшие 6 байтов U равны младшим 6-ти байтам U' и V≠V’

В приведенном протоколе аутентификации вместо ключа К одна из сторон может хранить мастер-ключ ИК, в этом случае ответная сторона должна хранить К=Ф5(МК,N), где N - серийный номер участника протокола (4 байта). При этом перед выполнением протокола, сторона, хранящая мастер-ключ должна получить серийный номер N от ответной стороны и вычислить К= Ф5(МК,N).

Описание функций протокола:

- Ф1(K,T,L) — Функция зашифровывания в режиме простой замены ГОСТ 28147-89 открытого текста Т размером L блоков на ключе К.
- Ф2(L) — Функция выработки случайного числа размером L блоков
- Ф5(МК,N) — Функция диверсификации ключа), где N- серийный номер участника протокола (4 байт) и МК — мастер-ключ.
- Ф7(R1,R2,R3,R4) — Подготовка исходных данных для выработки разового ключа в ИК. Входом алгоритма являются четыре 64-битных блока R1,R2,R3,R4.

  Выход: R (256 бит).

  Каждый блок разбивается на 2 32-битных подблока:

  R1=R11||R12, R2=R21||R22, R3=R31||R32, R4=R41||R42.

  Формируется R=R11||R31||R12||R32||R21||R41||R22||R42.

- Ф8(R1,R2,R3,R4) — Подготовка исходных данных для выработки разового ключа в T.

  Выход: R (256 бит).

  Входом алгоритма являются четыре 64-битных блока R1,R2,R3,R4.

  Каждый блок разбивается на 2 32-битных подблока:

  R1=R11||R12, R2=R21||R22, R3=R31||R32, R4=R41||R42.

  Формируется R=R31||R11||R32||R12||R41||R21||R42||R22.