ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

РЕКОМЕНДАЦИИ ПО СТАНДАРТИЗАЦИИ

Р 1323565.1.013—2017

Информационная технология

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

Использование режимов алгоритма блочного шифрования в протоколе защищенного обмена сообщениями в процессе эмиссии платежных карт

Издание официальное

Москва
Стандартинформ
2018

Предисловие

1 РАЗРАБОТАНЫ Открытым акционерным обществом «Информационные технологии и коммуникационные системы» (ОАО «ИнфоТеКС»)

2 ВНЕСЕНЫ Техническим комитетом по стандартизации ТК 26 «Криптографическая защита информации»

3 УТВЕРЖДЕНЫ И ВВЕДЕНЫ В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 28 декабря 2017 г. № 2116-ст

4 ВВЕДЕНЫ ВПЕРВЫЕ

Правила применения настоящих рекомендаций установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящим рекомендациям публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящих рекомендаций 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 Федерального агентства по техническому регулированию и метрологии в сети Интернет (www.gost.ru)

Содержание

Введение

В настоящих рекомендациях рассматривается протокол SCP-F2, предназначенный для защищенного обмена сообщениями между интегральной схемой смарт-карты и терминалом. Протокол SCP-F2 основывается на протоколе SCP02 спецификаций открытого стандарта GlobalPlatform [1]. В отличие от SCP02 в SCP-F2 используются российские криптографические алгоритмы ГОСТ 28147—89, ГОСТ Р 34.10—2012 и ГОСТ Р 34.11—2012.

Конструкции протокола SCP-F2 представляют собой защищенный канал связи между картой и внешним объектом, основанный на криптографических алгоритмах аутентификации, выработки ключей, конфиденциальности и целостности сообщений и предназначенный для использования системами при загрузке, установке и удалении приложений (апплетов) в интегральную схему смарт-карты.

Примечание — Настоящие рекомендации дополнены приложением А.

РЕКОМЕНДАЦИИ ПО СТАНДАРТИЗАЦИИ

Информационная технология

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

Использование режимов алгоритма блочного шифрования в протоколе защищенного обмена сообщениями в процессе эмиссии платежных карт

Information technology. Cryptographic data security.
Using of block encryption algorithm modes in secure messaging protocol for issuance of payment card

Дата введения — 2018—06—01

1 Область применения

Описанный в настоящих рекомендациях протокол рекомендуется применять при удаленном управлении контентом смарт-карты.

2 Нормативные ссылки

В настоящих рекомендациях использованы нормативные ссылки на следующие документы:

ГОСТ 28147—89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования

ГОСТ Р 34.10—2012 Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи

ГОСТ Р 34.11—2012 Информационная технология. Криптографическая защита информации. Функция хэширования

Р 50.1.113—2016 Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования

Р 1323565.1.010—2017 Информационная технология. Криптографическая защита информации. Использование функции диверсификации для формирования производных ключей платежного приложения

Примечание — При пользовании настоящими рекомендациями целесообразно проверить действие ссылочных документов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный документ, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого документа с учетом всех внесенных в данную версию изменений. Если заменен ссылочный документ, на который дана датированная ссылка, то рекомендуется использовать версию этого документа с указанным выше годом утверждения (принятия). Если после утверждения настоящих рекомендаций в ссылочный документ, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный документ отменен без замены, то положение, в котором дана ссылка на него, применяется в части, не затрагивающей эту ссылку.

Издание официальное

3 Обозначения

В настоящих рекомендациях использованы следующие обозначения:

V_n — Конечномерное векторное пространство размерности n;

$A||B$ — Конкатенация строк, то есть если $A \in V_{n_1}$, $B \in V_{n_2}$, $A = (a_{n1-1}, a_{n1-2}, \ldots, a_0)$, $B = (b_{n2-1}, b_{n2-2}, \ldots, b_0)$, то $A||B = (a_{n1-1}, a_{n1-2}, \ldots, a_0, b_{n2-1}, b_{n2-2}, \ldots, b_0) \in V_{n1+n2}$;

$\oplus$ — Операция покомпонентного сложения по модулю 2 двух двоичных строк одинаковой длины;

0^r — Строка, состоящая из r нулей;

ATC — Счетчик сессий (Application Transaction Counter). Длина счетчика равна 2 байтам;

APDU — Протокольный Блок Данных Приложения;

CIN — Номер Образа Карты/Идентификационный Номер Карты;

CLA — Класс Командного Сообщения. Длина значения равна 1 байту;

C-MAC — Имитовставка запроса; формируется для всех команд, начиная с EXTERNAL AUTHENTICATE;

DAP — Шаблон Аутентификации Данных;

D_K — Функция расшифрования;

E_K — Функция шифрования;

$HMAC_{256}$ — Функция вычисления кода аутентификации сообщения HMAC, использующей алгоритм HMAC_GOSTR3411_2012_256, описанный в Р 50.1.113—2016. Длина значения равна 32 байтам;

ICC — Смарт-карта;

ICV — Начальный Вектор Формирования Цепи;

INS — Указательный байт командного сообщения. Длина значения равна 1 байту;

ISD — Домен Безопасности Эмитента;

KDF — Функция диверсификации, использующая алгоритм KDF_GOSTR3411_2012_256, описанный в Р 50.1.113—2016. Длина значения равна 32 байтам;

K_{ENC} — Мастер-ключ, используемый для генерации сессионного ключа (S_{ENC}) для аутентификации и шифрования (формирования/проверки криптограммы аутентификации); формируется в соответствии с Р 1323565.1.010—2017. Длина значения равна 32 байтам;

K_{MAC} — Мастер-ключ, используемый для генерации сессионного ключа (S_{MAC}) для обеспечения целостности (генерации и проверки имитовставки); формируется в соответствии с Р 1323565.1.010—2017. Длина значения равна 32 байтам;

K_{DEC} — Мастер-ключ, используемый для генерации сессионного ключа (S_{DEC}) для шифрования и расшифрования критичных данных; формируется в соответствии с Р 1323565.1.010—2017. Длина значения равна 32 байтам;

Lc — Точная длина данных команды. Длина значения равна 1 байту;

Le — Максимальная длина данных, ожидаемых в ответ на команды. Длина значения равна 1 байту;

MAC — Код Аутентификации Сообщения;

P1 — Параметр 1 контроля указателя. Длина значения равна 1 байту;

P2 — Параметр 2 контроля указателя. Длина значения равна 1 байту;

RFU — Зарезервировано для последующего использования;

R-MAC — Имитовставка для контроля целостности ответных сообщений; вычисляется, начиная со следующей после EXTERNAL AUTHENTICATE команды;

SCP — Протокол Защищенного Канала;

SW — Слово Состояния. Длина значения равна 2 байтам;

SW1 — Первый байт Слова состояния;

SW2 — Второй байт Слова состояния;

S_{ENC} — Сессионный ключ для аутентификации и шифрования (формирования/проверки криптограммы аутентификации). Длина значения равна 32 байтам;

S^C_{MAC} — Сессионный ключ для обеспечения целостности (генерации и проверки имитовставки C-MAC). Длина значения равна 32 байтам;

S^{R}_{MAC} — Сессионный ключ для обеспечения целостности (генерации и проверки имитовставки R-MAC). Длина значения равна 32 байтам;

S_{DEC} — Сессионный ключ для шифрования и расшифрования критичных данных. Длина значения равна 32 байтам.

4 Описание

Возможные значения аргументов функций в представленных алгоритмах ограничены допустимостью их использования в качестве входных параметров преобразований.

4.1 Криптографический протокол

Для аутентификации между интегральной схемой карты и терминалом осуществляется следующее взаимодействие:

Терминал	Домен безопасности (интегральная схема карты)
Генерирует случайное число хоста	
Передает карте случайное число хоста при помощи команды INITIALIZE UPDATE	
	Генерирует случайное число карты
	Используя мастер-ключи, диверсифицирующую константу и счетчик сессий, вырабатывает сессионные ключи по ГОСТ 28147—89
	Используя случайные числа хоста и карты, счетчик сессий, синхропосылку и сессионные ключи, генерирует криптограмму аутентификации карты
	Передает ответное сообщение, содержащее информацию о ключе, счетчик сессий, случайное число карты и криптограмму карты. Инкрементирует счетчик сессий
Используя мастер-ключи, диверсифицирующую константу и счетчик сессий, вырабатывает сессионные ключи по ГОСТ 28147—89	
Используя случайные числа хоста и карты, счетчик сессий, синхропосылку и сессионные ключи, генерирует свою криптограмму аутентификации карты и сравнивает ее с возвращенной картой. Если криптограммы не совпали, процесс прерывается, все возвращается в исходное состояние	
Используя случайные числа хоста и карты, счетчик сессий, синхропосылку и сессионные ключи, генерирует криптограмму аутентификации хоста	
Формирует команду EXTERNAL AUTHENTICATE и рассчитывает имитовставку APDU-команды	
Передает карте криптограмму аутентификации хоста и имитовставку при помощи команды EXTERNAL AUTHENTICATE	
	Карта вычисляет имитовставку и проверяет имитовставку, переданную в команде, в случае неудачи возвращается статус ошибки, процесс установления защищенного соединения прерывается

Окончание таблицы

Терминал	Домен безопасности (интегральная схема карты)
	Используя случайные числа хоста и карты, счетчик сессий, синхропосылку и сессионные ключи, генерирует криптограмму аутентификации хоста и сравнивает ее с переданной хостом. Если криптограммы не совпали, процесс прерывается, все возвращается в исходное состояние, возвращается статус ошибки
	Байты статуса имеют значение '0x90' '0x00'

4.2 APDU-команды SCP-F2

В таблице 1 собраны минимальные требования безопасности для APDU-команд.

Таблица 1

Команда	Минимальная защита
INITIALIZE UPDATE	Нет
EXTERNAL AUTHENTICATE	Имитовставка

4.2.1 Команда INITIALIZE UPDATE

4.2.1.1 Назначение и применение

Команда INITIALIZE UPDATE используется при установке защищенного соединения для передачи данных карты и сессионных данных между картой и хостом. Эта команда инициирует установку сессии защищенного соединения.

В любой момент в течение текущего защищенного соединения команда INITIALIZE UPDATE может быть выдана карте для инициирования новой сессии защищенного соединения.

4.2.1.2 Командное сообщение

В таблице 2 показан формат командного сообщения INITIALIZE UPDATE.

Таблица 2

Код	Значение	Назначение
CLA	'80'	Открытая передача данных с использованием основного логического канала
	'81' — '83' или 'C0' — 'CF'	Открытая передача данных с использованием дополнительного логического канала
INS	'50'	INITIALIZE UPDATE
P1	'xx'	Номер версии ключа (Key Version Number) внутри домена безопасности, который должен быть использован для инициирования сессии защищенного соединения. Если это значение равно нулю, будет использован первый доступный ключ, выбранный доменом безопасности
P2	'00'	Управляющий параметр *P2*
Lc	'08'	Длина случайного числа хоста
Data	'xx xx...'	Случайное число хоста, выбранное терминальным устройством, должно быть уникальным для текущей сессии
Le	'00'	

4.2.1.3 Ответное сообщение

Поле данных в ответном сообщении содержит конкатенацию без разделителей элементов данных, описанных в таблице 3.

Таблица 3

Название		Длина
CIN карты (опционально)		10 байт
Данные диверсификации ключа	Информация о ключе, включающая номер версии ключа (Key Version Number) и идентификатор протокола SCP-F2 (здесь 'F2'), используемый для инициирования сессии защищенного соединения	2 байта
	Счетчик сессий, представляющий собой внутренний суммирующий счетчик, применяемый при создании сессионных ключей	2 байта
	Случайное число карты	6 байт
Криптограмма карты		6 байт

4.2.1.4 Обработка состояния, возвращаемого в ответном сообщении

При успешном выполнении команды байты статуса имеют значение '0x90' '0x00'.

Данная команда может возвращать одну из ошибок состояния INITIALIZE UPDATE, определенных в таблице 4.

Таблица 4

SW1	*SW2*	Значение
'64'	'00'	Неизвестная ошибка
'67'	'00'	Неверная длина *Lc*
'68'	'81'	Логический канал не поддерживается или не активен
'69'	'82'	Неверный статус безопасности
'69'	'85'	Неверные условия использования команды
'6A'	'86'	Неверные значения *P1 P2*
'6A'	'88'	Исходные данные не обнаружены
'6D'	'00'	Недопустимая команда
'6E'	'00'	Недопустимый класс

4.2.2 Команда EXTERNAL AUTHENTICATE

4.2.2.1 Назначение и применение

Команда EXTERNAL AUTHENTICATE используется картой во время инициирования защищенного соединения для аутентификации хоста и определения уровня безопасности, требуемого для последующих команд.

Этой команде должно предшествовать успешное выполнение команды INITIALIZE UPDATE.

4.2.2.2 Командное сообщение

В таблице 5 показан формат командного сообщения EXTERNAL AUTHENTICATE.

Таблица 5

Код	Значение	Обозначение
CLA	'84'	Защищенная передача данных с использованием основного логического канала
	'85' — '87' или 'E0' — 'EF'	Защищенная передача данных с использованием дополнительного логического канала
INS	'82'	EXTERNAL AUTHENTICATE

Окончание таблицы 5

Код	Значение	Обозначение
P1	'xx'	Уровень безопасности для всех команд с защитой сообщений, следующих за командой EXTERNAL AUTHENTICATE (он не применяется к данной команде) в течение данной сессии защищенного соединения. Управляющий параметр *P1* должен быть установлен согласно таблице 6
P2	'00'	Управляющий параметр *P2*
Lc	'0A'	Длина криптограммы хоста и имитовставки
Data	'xx xx…'	Криптограмма хоста и имитовставка
Le		Отсутствует

Возможные значения управляющего параметра *P1* команды EXTERNAL AUTHENTICATE описаны в таблице 6.

Таблица 6

b8	b7	b6	b5	b4	b3	b2	b1	Описание
0	0	1	1	0	0	1	1	RFU
0	0	1	1	0	0	0	1	RFU
0	0	1	1	0	0	0	0	RFU
0	0	0	1	0	0	1	1	C-DECRYPTION, C-MAC и R-MAC
0	0	0	1	0	0	0	1	C-MAC и R-MAC
0	0	0	1	0	0	0	0	R-MAC
0	0	0	0	0	0	0	1	C-MAC
0	0	0	0	0	0	0	0	Незащищенный режим

4.2.2.3 Обработка состояния, возвращаемого в ответном сообщении

Об успешном выполнении команды свидетельствуют значения байт статуса '0x90' '0x00'.

Эта команда может возвращать один из кодов предупреждения команды EXTERNAL AUTHENTICATE, определенных в таблице 7.

Таблица 7

SW1	*SW2*	Значение
'63'	'00'	Ошибка аутентификации криптограммы хоста
'64'	'00'	Неизвестная ошибка
'67'	'00'	Неверная длина *Lc*
'68'	'81'	Логический канал не поддерживается или не активен
'68'	'82'	Неверное значение криптограммы
'69'	'82'	Неверный статус безопасности
'69'	'85'	Неверные условия использования команды
'6A'	'86'	Неверные значения *P1 P2*
'6A'	'88'	Исходные данные не обнаружены
'6D'	'00'	Недопустимая команда
'6E'	'00'	Недопустимый класс

4.3 Уровень безопасности (Security Level)

Текущий уровень безопасности (Current Security Level) какого-либо обмена сообщениями, не включенного в сессию защищенного соединения, должен быть установлен в значение NO_SECURITY_LEVEL.

Текущий уровень безопасности (Current Security Level) для сессии защищенного соединения SCP-F2 представляет собой комбинацию следующих битов: AUTHENTICATED, C_MAC, R_MAC и C_DECRYPTION. Текущий уровень безопасности (Current Security Level) должен формироваться следующим образом:

- NO_SECURITY_LEVEL, если сессия защищенного соединения завершена или еще не полностью установлена;
- AUTHENTICATED устанавливается после успешной обработки команды EXTERNAL AUTHENTICATE. Бит AUTHENTICATED должен быть сброшен, как только сессия защищенного соединения завершена;
- C_MAC устанавливается после успешной обработки команды EXTERNAL AUTHENTICATE с управляющим параметром *P1*, указывающим на применение C-MAC (*P1* = 'x1' или 'x3'). Бит C_MAC должен быть сброшен, как только сессия защищенного соединения завершена. C_MAC синхронизирован с AUTHENTICATED и одновременно с ним выставляется и сбрасывается;
- C_DECRYPTION устанавливается после успешной обработки команды EXTERNAL AUTHENTICATE с управляющим параметром *P1*, указывающим на шифрование команды (*P1* = 'x3'). Бит C_DECRYPTION должен быть сброшен, как только сессия ЗОС завершена. C_DECRYPTION всегда синхронизирован с AUTHENTICATED и C_MAC и одновременно с ними выставляется и сбрасывается;
- R_MAC устанавливается после успешной обработки команды EXTERNAL AUTHENTICATE со значением управляющего параметра *P1*, указывающим на R-MAC (*P1* = '1x'). Бит R_MAC должен быть сброшен, как только сессия защищенного соединения завершена. R_MAC всегда используется вместе с AUTHENTICATED и одновременно с ним выставляется и сбрасывается. R_MAC может также сочетаться с C_MAC или C_DECRYPTION (в соответствии со значением управляющего параметра *P1* команды EXTERNAL AUTHENTICATE) и одновременно с ними устанавливаться и сбрасываться.

4.4 Правила протокола

К протоколу SCP-F2 применяются следующие правила:

- при успешной установке сессии защищенного соединения текущий уровень безопасности (Current Security Level) устанавливается в значение, указанное в команде EXTERNAL AUTHENTICATE;
- текущий уровень безопасности (Current Security Level) сохраняется в течение всей сессии защищенного соединения до тех пор, пока не будет успешно модифицирован по запросу приложения карты, открывшего эту сессию;
- когда текущий уровень безопасности (Current Security Level) не установлен в значение AUTHENTICTED (имеет значение NO_SECURITY_LEVEL), то:
- если сессия защищенного соединения была прервана во время той же самой сессии приложения, входящая команда должна быть отклонена с ошибкой безопасности;
- в противном случае верификация безопасности входящей команды не производится;
- если сессия защищенного соединения задействована для входящих команд (то есть Current Security Level установлен как минимум в AUTHENTICATED), безопасность входящей команды проверяется в соответствии с текущим уровнем безопасности (Current Security Level) независимо от признаков ЗОС в байте *CLA* заголовка команды:
- когда безопасность команды не соответствует текущему уровню безопасности (Current Security Level) (в том числе превышает), команда должна быть отвергнута с ошибкой безопасности, сессия защищенного соединения прервана, а текущий уровень безопасности (Current Security Level) сброшен в NO_SECURITY_LEVEL;
- если обнаружена ошибка протокола (некорректная криптограмма, неверный формат команды), команда должна быть отвергнута с ошибкой безопасности, сессия защищенного соединения прервана, а текущий уровень безопасности (Current Security Level) сброшен в значение NO_SECURITY_LEVEL;

- во всех остальных случаях сессия защищенного соединения остается активной, а значение текущего уровня безопасности (Current Security Level) неизменным. За дальнейшую обработку команды отвечает приложение;
- если сессия защищенного соединения прервана (ABORTED), она еще не считается завершенной (TERMINATED);
- текущая сессия защищенного соединения (прерванная или еще открытая) должна быть завершена, а текущий уровень безопасности (Current Security Level) сброшен в значение NO_SECURITY_LEVEL в следующих случаях:
- попытка инициирования новой сессии защищенного соединения (новая команда INITIALIZE UPDATE);
- завершение сессии приложения (например, выбор нового приложения);
- завершение сессии карты (карта сброшена или выключено ее питание).

4.5 Криптографические алгоритмы

Для SCP-F2 применяются криптографические алгоритмы, описанные в ГОСТ 28147—89, и алгоритмы хэширования, описанные в ГОСТ Р 34.11—2012.

При использовании алгоритмов шифрования ГОСТ 28147—89 должны использоваться следующие параметры алгоритма (узлы замены):

id-tc26-gost-28147-param-Z, «1.2.643.7.1.2.5.1.1».

4.5.1 Режим сцепления блоков для ГОСТ 28147—89

4.5.1.1 Зашифрование

Открытый и при необходимости дополненный текст $P \in V^*$, $|P| = 64 \cdot q$, представляется в виде: $P = P_1 || P_2 || \ldots || P_q$, $P_i \in V_{64}$, $i = 1, 2, \ldots, q$.

Блоки шифртекста вычисляются по следующему правилу:

$$C_0 = 0^{64},$$

$$C_i = E_K\left(P_i \oplus C_{i-1}\right), \ i = 1, 2, \ldots, q.$$

Результирующий шифртекст имеет вид: $C = C_1 || C_2 || \ldots || C_q$.

4.5.1.2 Расшифрование

Шифртекст представляется в виде: $C = C_1 || C_2 || \ldots || C_q$, $C_i \in V_{64}$, $i = 1, 2, \ldots, q$.

Блоки открытого текста вычисляются по следующему правилу:

$$C_0 = 0^{64},$$

$$P_i = D_K(C_i) \oplus C_{i-1}.$$

Исходный (дополненный) открытый текст имеет вид: $P = P_1 || P_2 || \ldots || P_q$.

4.5.2 Сессионные ключи ГОСТ 28147—89

Сессионные ключи ГОСТ 28147—89 генерируются каждый раз при установлении защищенного соединения. Эти сессионные ключи должны быть использованы для защиты всех последующих команд.

4.5.2.1 Сессионный ключ для вычисления имитовставки запроса C-MAC

Формирование сессионного ключа S_{MAC}^{C} осуществляется с использованием функции диверсификации KDF_GOSTR3411_2012_256 на основе $HMAC_{256}$ и определяется выражением:

$$S_{MAC}^{C} = KDF(K_{in}, label, seed) = HMAC_{256}(K_{in}, 0x01 || label || 0x00 || seed || 0x01 || 0x00),$$

где:

$$K_{in} = K_{MAC}$$

$$label = (0x01 || 0x01)$$

$$seed = ATC,$$

то есть

$$S_{MAC}^{C} = HMAC_{256}(K_{MAC}, 0x01 || label || 0x00 || ATC || 0x01 || 0x00).$$

4.5.2.2 Сессионный ключ для вычисления имитовставки ответа R-MAC

Формирование сессионного ключа S^R_{MAC} осуществляется с использованием функции диверсификации KDF_GOSTR3411_2012_256 на основе $HMAC_{256}$ и определяется выражением:

$$S^R_{MAC} = KDF(K_{in}, label, seed) = HMAC_{256}(K_{in}, 0x01||label||0x00||seed||0x01||0x00),$$

где:

$K_{in} = K_{MAC}$

$label = (0x01||0x02)$

$seed = ATC$,

то есть

$$S^R_{MAC} = HMAC_{256}(K_{MAC}, 0x01||label||0x00||ATC||0x01||0x00).$$

4.5.2.3 Сессионный ключ для формирования/проверки криптограммы аутентификации

Формирование сессионного ключа S^C_{MAC} осуществляется с использованием функции диверсификации KDF_GOSTR3411_2012_256 на основе $HMAC_{256}$ и определяется выражением:

$$S_{ENC} = KDF(K_{in}, label, seed) = HMAC_{256}(K_{in}, 0x01||label||0x00||seed||0x01||0x00),$$

где:

$K_{in} = K_{ENC}$

$label = (0x01||0x82)$

$seed = ATC$,

то есть

$$S_{ENC} = HMAC_{256}(K_{ENC}, 0x01||label||0x00||ATC||0x01||0x00).$$

4.5.2.4 Сессионный ключ для шифрования/расшифрования критичных данных

Формирование сессионного ключа S^C_{MAC} осуществляется с использованием функции диверсификации KDF_GOSTR3411_2012_256 на основе $HMAC_{256}$ и определяется выражением:

$$S_{DEC} = KDF(K_{in}, label, seed) = HMAC_{256}(K_{in}, 0x01||label||0x00||seed||0x01||0x00),$$

где:

$K_{in} = K_{DEC}$

$label = (0x01||0x81)$

$seed = ATC$,

то есть

$$S_{ENC} = HMAC_{256}(K_{DEC}, 0x01||label||0x00||ATC||0x01||0x00).$$

4.5.3 Криптограмма аутентификации при инициировании защищенного соединения

Как карта, так и терминальное устройство (хост) генерируют криптограммы аутентификации. Терминальное устройство верифицирует криптограмму карты, а карта верифицирует криптограмму хоста. При генерации и верификации криптограмм аутентификации используется сессионный ключ S_{ENC} в режиме сцепления блоков. Криптограмма аутентификации представляет собой старшие (первые) 6 байтов последнего блока.

4.5.3.1 Криптограмма аутентификации карты

Генерация и верификация криптограммы карты выполняется посредством конкатенации 8 байтов случайного числа хоста (терминального устройства), 2 байтов счетчика сессий и 6 байтов случайного числа карты в один 16-байтовый блок.

Эти данные должны быть дополнены блоком из 8 байтов — ('0x80'||'0x00'||'0x00'||'0x00'||'0x00'||'0x00'||'0x00'||'0x00').

Далее к этому 24-байтовому блоку применяется алгоритм зашифрования ГОСТ 28147—89 в режиме сцепления блоков с использованием сессионного ключа S_{ENC} и синхропосылки ICV = ('0x00'||'0x00'||'0x00'||'0x00'||'0x00'||'0x00'||'0x00'||'0x00').

Криптограмма аутентификации представляет собой старшие (первые) 6 байтов последнего блока.

4.5.3.2 Криптограмма аутентификации хоста

Генерация и верификация криптограммы хоста выполняется посредством конкатенации 2 байтов последовательного счетчика, 6 байтов случайного числа карты и 8 байтов случайного числа хоста в один 16-байтовый блок.

Эти данные должны быть дополнены блоком из 8 байтов — ('0x80'||'0x00'||'0x00'||'0x00'||'0x00'||'0x00'||'0x00'||'0x00').

Далее к этому 24-байтовому блоку применяется алгоритм шифрования ГОСТ 28147—89 в режиме сцепления блоков с использованием сессионного ключа S_{ENC} и синхропосылки ICV = ('0x00'||'0x00'||'0x00'||'0x00'||'0x00'||'0x00'||'0x00'||'0x00').

Криптограмма аутентификации представляет собой старшие (первые) 6 байтов последнего блока.

4.5.4 Генерация и верификация имитовставки C-MAC APDU-команды

Имитовставка запроса (C-MAC) генерируется терминальным устройством по всем полям APDU запроса, кроме *Le*.

Имитовставка C-MAC имеет размер 4 байта и вычисляется по следующей схеме:

1 C-MAC формируется для всех команд, начиная с EXTERNAL AUTHENTICATE. Для вычисления первого C-MAC$_0$ команды EXTERNAL AUTHENTICATE используется синхропосылка, состоящая из двоичных нулей;

2 Для всех последующих команд в качестве синхропосылки ICV^C используется имитовставка предыдущей команды, выровненная стандартным способом [добавление справа ('0x80'||'0x00'||'0x00'||'0x00')] и зашифрованная по ГОСТ 28147—89 в режиме простой замены с сессионным ключом для расчета имитовставки (S^C_{MAC}), то есть

$$ICV_i^C = E_{S^C_{MAC}}[\text{C-MAC}_{i-1}||'0x80'||'0x00'||'0x00'||'0x00'], i = 1, \ldots;$$

3 Корректируется *Lc* с учетом добавления C-MAC к полю данных запроса, то есть

$Lc' = Lc + 4$ (Lc' — модифицированный Lc);

4 Модифицируется *CLA*: устанавливается бит защиты сообщений (в 3-й бит устанавливается значение '1'). *CLA*' — модифицированный *CLA*;

5 Подготавливаются данные для вычисления C-MAC:

$$M = (CLA'||INS||P1||P2||Lc'||[Data]);$$

6 Выполняется вычисление имитовставки по алгоритму ГОСТ 28147—89 на сессионном ключе S^C_{MAC}:

$$\text{C-MAC} = MAC(S^C_{MAC})\,[ICV^C||M];$$

7 C-MAC добавляется в конец командного сообщения APDU к полю данных.

После вычисления C-MAC APDU команда имеет вид:

$$\text{APDU} = CLA'||INS||P1||P2||Lc'||Data||\text{C-MAC}.$$

Чтобы верифицировать имитовставку C-MAC, карта использует синхропосылку ICV^C и сессионный ключ S^C_{MAC}, аналогичные тем, что использовались терминальным устройством для генерации имитовставки C-MAC. Верифицированная имитовставка C-MAC запоминается и используется в качестве синхропосылки ICV^C для следующей имитовставки. Это справедливо независимо от того, выполнена команда APDU успешно или нет, то есть верифицированная имитовставка C-MAC никогда не сбрасывается (в случае ошибки команда может быть повторена, но имитовставка должна быть другой).

Если сессия защищенного соединения происходит на вспомогательном логическом канале (Supplementary Logical Channel), номер логического канала добавляется к байту класса сообщения после генерации имитовставки C-MAC и до передачи сообщения карте. Карта должна удалять любую информацию о логическом канале из байта класса (то есть считать, что номер логического канала равен нулю) перед верификацией имитовставки C-MAC.

4.5.5 Генерация и верификация имитовставки R-MAC APDU-ответа

Протокол SCP-F2 предусматривает возможность контроля целостности ответных сообщений при помощи имитовставки R-MAC. Если при инициировании сессии защищенного соединения указан уровень безопасности, предполагающий вычисление R-MAC, то имитовставка R-MAC автоматически присоединяется ко всем ответным сообщениям, не считая первого (на команду EXTERNAL AUTHENTICATE).

Имитовставка R-MAC имеет размер 4 байта и вычисляется по следующей схеме:

1 R-MAC вычисляется, начиная со следующей после EXTERNAL AUTHENTICATE команды. В качестве ICV^R для вычисления первого R-MAC берется C-MAC от команды EXTERTAL AUTHNTICATE, дополненный до 8 байтов нулями, то есть

$$ICV_1^R = (\text{C-MAC}_0 || '0x00' || '0x00' || '0x00' || '0x00');$$

2 Для всех последующих команд в качестве ICV^R берется предыдущий R-MAC, дополненный до 8 байтов нулями, то есть

$$ICV_i^R = (\text{R-MAC}_{i-1} || '0x00' || '0x00' || '0x00' || '0x00'), i = 2, \ldots;$$

3 В байте *CLA* команды сбрасываются биты защищенного обмена и номер логического канала (*CLA*' — модифицированный *CLA*);

4 Выставляется байт *Li*, в котором закодирована длина данных ответа по модулю 256. *Li* генерируется картой. Если данных ответа нет, этот байт все равно присутствует и установлен в ноль. В случае ошибки байт *Li* выставляется в '0x00', указывая на отсутствие данных ответа;

5 Подготавливаются данные для вычисления R-MAC:

$M = (CLA' || INS || P1 || P2 || Lc$ (без учета шифрования и C-MAC)||

Data (без шифрования и C-MAC)||*Li*||*R-Data* (данные ответа)||*SW*);

6 Выполняется вычисление имитовставки по ГОСТ 28147—89 на сессионном ключе S_{MAC}^R (с надлежащим выравниванием данных) и использованием синхропосылки ICV^R:

$$\text{R-MAC} = MAC(S_{MAC}^R)\,[ICV^R || M];$$

7 R-MAC добавляется к полю ответа.

После вычисления R-MAC APDU команда имеет вид:

$$\text{APDU} = CLA' || INS || P1 || P2 || Lc || Data || Li || R\text{-}Data || \text{R-MAC} || SW.$$

Сгенерированная имитовставка R-MAC должна сохраняться и использоваться как синхропосылка ICV^R для вычисления следующего значения R-MAC, независимо от того, выполнена APDU-команда успешно или нет. В случае обработки команд, в которых не предусмотрено поле данных ответа, добавляется поле *Le*, равное нулю, что означает готовность терминала принять до 256 байтов данных в ответе.

4.5.6 Шифрование поля данных APDU-команды

В SCP-F2 для всех APDU-команд может применяться шифрование сообщений.

Терминальное устройство зашифровывает поле данных командного сообщения, передаваемого карте. Любая ключевая информация внутри поля данных, которая шифруется на сессионном ключе шифрования критичных данных S_{DEC}, например секретные или закрытые ключи, также шифруется вместе с другими данными. Заголовок и имитовставка при этом не шифруются.

Шифрование и расшифрование командного сообщения производится с использованием ГОСТ 28147—89 в режиме сцепления блоков, с использованием сессионного ключа S_{ENC} и синхропосылки ICV^E, которая вычисляется следующим образом:

Синхропосылка ICV^E для команды-запроса вычисляется по схеме:

$$ICV_i^E = E_{S_{MAC}^C}[\text{C-MAC}_i || '0x80' || '0x00' || '0x00' || '0x00'], i = 1, \ldots$$

Синхропосылка ICV^E для команды-ответа вычисляется по схеме:

$$ICV_i^E = E_{S_{MAC}^R}[\text{R-MAC}_i || '0x80' || '0x00' || '0x00' || '0x00'], i = 1, \ldots$$

Результат операции становится частью поля данных в командном сообщении.

Перед шифрованием данных они должны быть выровнены согласно следующей процедуре:

- добавить '0x80' справа от блока данных;
- если полученная длина блока данных кратна 64 битам, то дальнейшее заполнение не требуется;
- добавлять двоичные нули в правой части блока данных, пока длина блока данных не станет кратной 64 битам.

В отличие от имитовставки байты выравнивания теперь становятся частью поля данных, в результате чего снова требуется изменить значение *Lc*. Шифрование данных выполняется после вычисления имитовставок C-MAC и R-MAC, если они предусмотрены действующим уровнем безопасности.

Сообщение передается карте. Карта должна сначала расшифровать командное сообщение и удалить байты выравнивания, прежде чем анализировать имитовставку C-MAC. При расшифровании используется синхропосылка ICV^E, вычисленная по той же схеме, что и для шифрования, и тот же сессионный ключ S_{ENC}. Байты выравнивания должны быть удалены, а длина *Lc* модифицирована. В результате длина *Lc* будет равна сумме длины оригинального текста и длины имитовставки C-MAC.

4.5.7 Шифрование и расшифрование критичных данных

Шифрование данных используется при передаче карте критичных данных. Например, в команде PUT KEY необходимо шифровать все передаваемые карте ключи.

Процесс шифрования данных производится с использованием ГОСТ 28147—89 в режиме сцепления блоков, с использованием сессионного ключа S_{DEC} и синхропосылки ICV^E, которая вычисляется следующим образом:

Синхропосылка ICV^E для команды-запроса вычисляется по схеме:

$$ICV_i^E = E_{S_{MAC}^C}[\text{C-MAC}_i \,||\, '0x80' || '0x00' || '0x00' || '0x00'],\ i = 1, \ldots$$

Синхропосылка ICV^E для команды-ответа вычисляется по схеме:

$$ICV_i^E = E_{S_{MAC}^R}[\text{R-MAC}_i \,||\, '0x80' || '0x00' || '0x00' || '0x00'],\ i = 1, \ldots$$

Поскольку все стандартные ключи имеют длину 32 байта (кратны размеру блока — 8 байтов), выравнивание для операций шифрования не требуется.

К зашифрованным критичным данным применяются все криптографические операции защищенного обмена сообщениями (шифрование, вычисление C-MAC, R-MAC), как к обыкновенным данным команды.

Процесс расшифрования критичных данных картой прямо противоположен описанной выше операции: в частности, при операции расшифрования не удаляются выравнивающие байты, так как выравнивание не используется.

4.6 Управление контентом карты

Введем коды для криптографических алгоритмов на эллиптических кривых, которые можно использовать для ГОСТ Р34.10—2012:

'B0' — ECC public key;

'B1' — ECC private key.

Управление контентом карты осуществляется согласно базовой спецификации с учетом следующих особенностей.

4.6.1 Проверка DAP-блока

Провайдер приложения или контролирующий орган могут потребовать проверить целостность кода приложения и аутентификацию до того, как код приложения загружен, установлен или доступен держателю карты. Привилегии обязательной DAP-верификации предоставляют данные сервисы от имени контролирующего органа.

Проверка DAP-блока осуществляется с использованием алгоритма ГОСТ 28147—89 в режиме вычисления имитовставки. Для генерации Load File Data Block Hash должен использоваться алгоритм ГОСТ Р34.11—2012 (длина выходного значения хэш-функции равна 256 битов). Для вычисления имитовставки от хэш-значения используется отдельный ключ K-DAP. Длина имитовставки — 4 байта.

4.6.2 Токены

Токены делегированного управления являются подписями одной или нескольких функций делегированного управления (загрузка, установка и выгрузка), сформированных эмитентом карты и использующихся для обеспечения контроля эмитентом карты изменений ее контента. Токены требуются, когда Домен Безопасности Эмитента (ISD) не может сам отслеживать изменения. ISD должен проверять токены.

Для вычисления токенов используется алгоритм вычисления ЭП ГОСТ Р34.10—2012 вместе с алгоритмом хэширования ГОСТ Р34.11—2012. Управление ключами производится согласно документу [2]. Размер подписи — 64 байта.

4.6.3 Квитанции (receipts)

ISD может формировать квитанции в процессе делегированного управления. Квитанции подтверждают эмитенту карты, что диспетчер приложения изменил контент карты.

Для вычисления квитанций используется алгоритм ГОСТ 28147—89 в режиме вычисления имитовставки. Длина квитанции соответствует длине имитовставки — 4 байта.

Приложение А
(справочное)

Контрольные примеры

Приводимые ниже значения счетчика АТС, случайных чисел карты и хоста, данных APDU-команд, а также значения мастер-ключей K_{MAC}, K_{ENC}, K_{DEC} рекомендуется использовать только для проверки корректной работы конкретной реализации алгоритмов, описанных в настоящих рекомендациях.

Все числовые значения приведены в десятичной или шестнадцатеричной записи. Нижний индекс в записи числа обозначает основание системы счисления.

В данном приложении двоичные строки из V^*, длина которых кратна 4, записываются в шестнадцатеричном виде, а символ конкатенации ("||") опускается. То есть строка $a \in V_{4r}$ будет представлена в виде $a_{r-1}\, a_{r-2} \ldots a_0$, где $a_i \in \{0, 1, \ldots, 9, a, b, c, d, e, f\}$, $i = 0, 1, \ldots, r-1$. Соответствие между двоичными строками длины 4 и шестнадцатеричными строками длины 1 задается естественным образом (таблица А.1).

Преобразование, ставящее в соответствие двоичной строке длины $4r$ шестнадцатеричную строку длины r, и соответствующее обратное преобразование для простоты записи опускаются.

Таблица А.1 — Соответствие между двоичными и шестнадцатеричными строками

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	a
1011	b
1100	c
1101	d
1110	e
1111	f

Символ «\\» обозначает перенос числа на новую строку.

А.1 Исходные данные

АТС = 0010_{16}

Случайное число карты: 010203040506_{16}

Случайное чисто хоста: 0102030405060708_{16}

А.1.1 Сессионные ключи

Мастер-ключи:

K_{MAC} = 3*d*292*eecd*26*b*7963*b*4*c*980*d*5*fcd*3068*f* \\
624*b*6*d*56*b*434326*d*89*cdf*5842*b*193006$_{16}$

K_{ENC} = 239*ae*6*ef*90*a*1*ebd*1*fbc*2*a*3*cf*695*e*6*f*10 \\
*bfd*1*b*2*da*6*e*73*e*04*dc*5*b*76*de*4*aa*7*ac*544$_{16}$

K_{DEC} = ce9ec8c79b8a679b2b12bf5514143b5a \\
9a805fd615f801b2b856921ddd216130$_{16}$

А.1.1.1 Сессионный ключ для вычисления имитовставки запроса C-MAC

Для вычисления сессионного ключа S^C_{MAC} используются мастер-ключ K_{MAC} и константа $label = 0101_{16}$. На основе этих исходных данных получаем следующее значение сессионного ключа S^C_{MAC}:

S^C_{MAC} = e7a72288c845ec6549377b1b30813f05 \\
05f1846195fbfedf750ca8918a857d7e$_{16}$

А.1.1.2 Сессионный ключ для вычисления имитовставки ответа R-MAC

Для вычисления сессионного ключа S^R_{MAC} используются мастер-ключ K_{MAC} и константа $label = 0102_{16}$. На основе этих исходных данных получаем следующее значение сессионного ключа S^R_{MAC}:

S^R_{MAC} = 3e763841b860ec2189c91949db50fc30 \\
6ff907d3f9030f51bd20f9e46342f1c6$_{16}$

А.1.1.3 Сессионный ключ для формирования/проверки криптограммы аутентификации

Для вычисления сессионного ключа S_{ENC} используются мастер-ключ K_{ENC} и константа $label = 0182_{16}$. На основе этих исходных данных получаем следующее значение сессионного ключа S_{ENC}:

S_{ENC} = a511f2d7a74f7f2aad9fa068b79d1c42 \\
cb11f4bcdb6191d6ca881566de06ea52$_{16}$

А.1.1.4 Сессионный ключ для шифрования/расшифрования критичных данных

Для вычисления сессионного ключа S_{DEC} используются мастер-ключ K_{DEC} и константа $label = 0181_{16}$. На основе этих исходных данных получаем следующее значение сессионного ключа S_{DEC}:

S_{DEC} = 7bcd37b59f11c203622ce4df853fab72 \\
49d351d67a19da47f0cc65b4d99185b1$_{16}$

А.1.2 Криптограмма аутентификации карты

Для вычисления криптограммы аутентификации карты используются сессионный ключ S_{ENC}, случайное число карты, случайное число хоста и счетчик сессий. На основе этих исходных данных получаем следующее значение криптограммы карты:

$cardCryptogram = ab404dd3a931_{16}$

А.1.3 Криптограмма аутентификации хоста

Для вычисления криптограммы аутентификации хоста используются сессионный ключ S_{ENC}, случайное число карты, случайное число хоста и счетчик сессий. На основе этих исходных данных получаем следующее значение криптограммы хоста:

$terminalCryptogram = 2b9b124505c0_{16}$

А.1.4 Генерация имитовставки C-MAC APDU-команды

Для вычисления имитовставки C-MAC для команды EXTERNAL AUTHENTICATE используются сессионный ключ S^C_{MAC} и следующие исходные данные:

$$(CLA||INS||P_1||P_2||Lc||Data) = (84||82||13||00||06||2b9b124505c0)_{16}$$

$$ICV = 00000000_{16}$$

На основе этих исходных данных получаем следующее значение C-MAC:

C-MAC = 98434854$_{16}$

Тогда APDU-команда будет иметь вид:

EXTERNAL AUTHENTICATE = 848213000a2b9b124505c098434854$_{16}$

А.1.5 Генерация имитовставки R-MAC APDU-команды

Для вычисления имитовставки R-MAC для команды, следующей за EXTERNAL AUTHENTICATE, используются сессионный ключ S^R_{MAC} и следующие исходные данные:

$$(CLA||INS||P_1||P_2||Lc||Data) = (84||ca||13||00||03||119a10)_{16}$$

Данные ответа отсутствуют, то есть: $(Li||R\text{-}Data) = (00)_{16}$

$SW = 9000_{16}$

$ICV = 98434854_{16}$

На основе этих исходных данных получаем следующее значение R-MAC:

R-MAC = $3d824337_{16}$

Тогда APDU-команда будет иметь вид:

APDU = $84ca130003119a10003d8243379000_{16}$

A.1.6 Шифрование поля данных APDU-команды

Команда APDU с имитовставкой C-MAC:

APDU = $84ca130007119a1014ac12dc_{16}$

Следовательно, шифруются данные: $Data = 119a10_{16}$

После шифрования поля данных на сессионном ключе S_{ENC} команда принимает вид:

APDU = $84ca13000c7b91cf97ccc6a3d014ac12dc_{16}$

A.1.7 Шифрование критичных данных APDU-команды

Команда APDU с критичными данными и имитовставкой C-MAC:

APDU = 84*ca*130024 \\
590*a*133*c*6*bf*0*de*92209*d*18*f*804*c*754*db* \\
4*c*02*a*8672*efb*984*a*417*eb*5179*b*401289 \\
$a2cc4ed5_{16}$

Следовательно, шифруются критичные данные:

Data = 590*a*133*c*6*bf*0*de*92209*d*18*f*804*c*754*db* \\
$4c02a8672efb984a417eb5179b401289_{16}$

После шифрования критичных данных на сессионном ключе S_{DEC} команда принимает вид:

APDU = 84*ca*130024 \\
*e*065*ed*007148*c*2*ede*3*eccf*328318*ef*73 \\
16342*a*5*ad*1*acefb*0*eb*6*be*05*dc*43184*a*4 \\
$a2cc4ed5_{16}$

A.2 Исходные данные

ATC = 0003_{16}

Случайное число карты: 110213041516_{16}

Случайное чисто хоста: 6122335405062938_{16}

A.2.1 Сессионные ключи

Мастер-ключи:

K_{MAC} = *d*5*f*40*f*395712*ec*4*e*47540318*b*5*b*718*eb* \\
$8bb195994ff10e7c6e4a896760f443f7_{16}$

K_{ENC} = 63*b*47*cd*8*e*6*b*3743946*f*279*be*412*e*9*f*87 \\
$19013ee919ab99ee0b253cd5f5c43978_{16}$

K_{DEC} = 0*f*17*df*77467*bcc*4*deef*2*c*016*eed*30753 \\
$2d337d21f5ed1295234528a4c9fe1fc7_{16}$

A.2.1.1 Сессионный ключ для вычисления имитовставки запроса C-MAC

Для вычисления сессионного ключа S^C_{MAC} используются мастер-ключ K_{MAC} и константа $label = 0101_{16}$. На основе этих исходных данных получаем следующее значение сессионного ключа S^C_{MAC}:

S^C_{MAC} = 428*d*1*aa*8893*b*2*bb*797*e*71*e*87612*b*6548 \\
$4014e81870c1e0ac7f7377a12fb4a621_{16}$

А.2.1.2 Сессионный ключ для вычисления имитовставки ответа R-MAC

Для вычисления сессионного ключа S^R_{MAC} используются мастер-ключ K_{MAC} и константа $label = 0102_{16}$. На основе этих исходных данных получаем следующее значение сессионного ключа S^R_{MAC}:

$$S^R_{MAC} = 6d2db8b5a508694baec0ce6e1276a3b4 \backslash\backslash$$
$$8ef84b5744452ce6ad5fd9595651d40a_{16}$$

А.2.1.3 Сессионный ключ для формирования/проверки криптограммы аутентификации

Для вычисления сессионного ключа S_{ENC} используются мастер-ключ K_{ENC} и константа $label = 0182_{16}$. На основе этих исходных данных получаем следующее значение сессионного ключа S_{ENC}:

$$S_{ENC} = 7549c87538736a8237f339ce872a34ed \backslash\backslash$$
$$d833bc02318e46d6086df8f84b0b1550_{16}$$

А.2.1.4 Сессионный ключ для шифрования/расшифрования критичных данных

Для вычисления сессионного ключа S_{DEC} используются мастер-ключ K_{DEC} и константа $label = 0181_{16}$. На основе этих исходных данных получаем следующее значение сессионного ключа S_{DEC}:

$$S_{DEC} = 5ccffaaf038c5dbc023b077c13d43c45 \backslash\backslash$$
$$e98ec17b628b29709ba99075bf9ec60a_{16}$$

А.2.2 Криптограмма аутентификации карты

Для вычисления криптограммы аутентификации карты используются сессионный ключ S_{ENC}, случайное число карты, случайное число хоста и счетчик сессий. На основе этих исходных данных получаем следующее значение криптограммы карты:

$$cardCryptogram = 9fe76e33976b_{16}$$

А.2.3 Криптограмма аутентификации хоста

Для вычисления криптограммы аутентификации хоста используются сессионный ключ S_{ENC}, случайное число карты, случайное число хоста и счетчик сессий. На основе этих исходных данных получаем следующее значение криптограммы хоста:

$$terminalCryptogram = 1be4f4ae3e03_{16}$$

А.2.4 Генерация имитовставки C-MAC APDU-команды

Для вычисления имитовставки C-MAC для команды EXTERNAL AUTHENTICATE используются сессионный ключ S^C_{MAC} и следующие исходные данные:

$$(CLA||INS||P_1||P_2||Lc||Data) = (84||82||13||00||06||1be4f4ae3e03)_{16}$$

$$ICV = 00000000_{16}$$

На основе этих исходных данных получаем следующее значение C-MAC:

$$C\text{-}MAC = f43be2fb_{16}$$

Тогда APDU-команда будет иметь вид:

$$EXTERNAL\ AUTHENTICATE = 848213000a1be4f4ae3e03f43be2fb_{16}$$

А.2.5 Генерация имитовставки R-MAC APDU-команды

Для вычисления имитовставки R-MAC для APDU-команды используются сессионный ключ S^R_{MAC} и следующие исходные данные:

$$(CLA||INS||P_1||P_2) = (84||ca||13||00)_{16}$$

$$Lc = 20_{16}$$

$$Data = 590a133c6bf0de92209d18f804c754db \backslash\backslash$$
$$4c02a8672efb984a417eb5179b401289_{16}$$

Данные ответа:

$$Li = 20_{16}$$

$$R\text{-}Data = b1728c371137d95f388fb8fe4773e2a8 \backslash\backslash$$
$$811725d4cc05801e1e0a0230990b5803_{16}$$

$SW = 9000_{16}$

$ICV = 3d824337_{16}$

На основе этих исходных данных получаем следующее значение R-MAC:

R-MAC = $4aca4f14_{16}$

Тогда APDU-команда будет иметь вид:

APDU = 84*ca*130020 \ \
590*a*133*c*6*bf*0*de*92209*d*18*f*804*c*754*db* \ \
4*c*02*a*8672*efb*984*a*417*eb*5179*b*401289 \ \
20 \ \
*b*1728*c*371137*d*95*f*388*fb*8*fe*4773*e*2*a*8 \ \
811725*d*4*cc*05801*e*1*e*0*a*0230990*b*5803 \ \
$4aca4f149000_{16}$

A.2.6 Шифрование поля данных APDU-команды

Команда EXTERNAL AUTHENTICATE с имитовставкой C-MAC:

EXTERNAL AUTHENTICATE = $848213000a1be4f4ae3e03f43be2fb_{16}$

Следовательно, шифруются данные: *Data* = $1be4f4ae3e03_{16}$

После шифрования поля данных на сессионном ключе S_{ENC} команда принимает вид:

EXTERNAL AUTHENTICATE = $848213000c8b33b1ddf6dcfabaf43be2fb_{16}$

A.2.7 Шифрование критичных данных APDU-команды

Команда APDU с критичными данными:

APDU = 84*ca*130020 \ \
590*a*133*c*6*bf*0*de*92209*d*18*f*804*c*754*db* \ \
4*c*02*a*8672*efb*984*a*417*eb*5179*b*401289 \ \
20 \ \
*b*1728*c*371137*d*95*f*388*fb*8*fe*4773*e*2*a*8 \ \
811725*d*4*cc*05801*e*1*e*0*a*0230990*b*5803 \ \
$4aca4f149000_{16}$

Следовательно, шифруются критичные данные:

Data = 590*a*133*c*6*bf*0*de*92209*d*18*f*804*c*754*db* \ \
$4c02a8672efb984a417eb5179b401289_{16}$

R-Data = *b*1728*c*371137*d*95*f*388*fb*8*fe*4773*e*2*a*8 \ \
$811725d4cc05801e1e0a0230990b5803_{16}$

После шифрования критичных данных на сессионном ключе S_{DEC} команда принимает вид:

APDU = 84*ca*130020 \ \
*f*31*a*7*b*24*c*3*a*2354*dda*9*ceeb*3822*f*1*a*15 \ \
837*f*527*a*1*a*1*fb*6*fad*577*b*43*b*8*d*69907*e* \ \
20 \ \
*bec*205*c*6*a*135*a*74312*d*06*dfa*7*ac*59722 \ \
5*cfe*3*e*2*b*16893*f*066231*d*44271*c*83931 \ \
$4aca4f149000_{16}$

А.3 Исходные данные

ATC = 0001_{16}

Случайное число карты: 112213562389_{16}

Случайное чисто хоста: 7832336312062934_{16}

А.3.1 Сессионные ключи

Мастер-ключи:

K_{MAC} = 9*ce*94350*c*5*e*9*b*9*f*835888*f*6065956*efb* \ \
*a*6133*ad*1*fba*2*fc*31303*caae*56*e*6*ea*6*ea*$_{16}$

K_{ENC} = 8*f*6*fe*73189*b*70614*d*518*d*8*bc*56759578 \ \
58*da*3*b*9825*ddb*705787*cff*81*d*57*ec*81*d*$_{16}$

K_{DEC} = *cadf*60*b*985*e*8*ca*702*a*98*e*49*ab*4*ed*53*b*5 \ \
5*ed*1*e*7*d*2*adaeae*46*cb*1*c*3*e*2*efb*7607*bb*$_{16}$

А.3.1.1 Сессионный ключ для вычисления имитовставки запроса C-MAC

Для вычисления сессионного ключа S^{C}_{MAC} используются мастер-ключ K_{MAC} и константа *label* = 0101_{16}. На основе этих исходных данных получаем следующее значение сессионного ключа S^{C}_{MAC}:

S^{C}_{MAC} = *aa*6*bde*5543*a*6*f*9*e*8*e*0*f*74*b*5*aa*8*a*985*b*7 \ \
56*adb*9*e*0*caf*1569*f*17*d*5937*ca*2*c*54*dd*7$_{16}$

А.3.1.2 Сессионный ключ для вычисления имитовставки ответа R-MAC

Для вычисления сессионного ключа S^{R}_{MAC} используются мастер-ключ K_{MAC} и константа *label* = 0102_{16}. На основе этих исходных данных получаем следующее значение сессионного ключа S^{R}_{MAC}:

S^{R}_{MAC} = 7873*acad*0*c*15*dc*01*b*2*bf*89896*a*7*f*5*c*81 \ \
*cbc*12*fea*2*e*72*a*89*f*5*b*898*a*774*d*4*d*9*c*11$_{16}$

А.3.1.3 Сессионный ключ для формирования/проверки криптограммы аутентификации

Для вычисления сессионного ключа S_{ENC} используются мастер-ключ K_{ENC} и константа *label* = 0182_{16}. На основе этих исходных данных получаем следующее значение сессионного ключа S_{ENC}:

S_{ENC} = *bcfbcc*813*b*7020*b*5*a*903722*cfb*4516*bf* \ \
0*b*96*b*9*dd*914828046*ffea*204318*c*2*f*56$_{16}$

А.3.1.4 Сессионный ключ для шифрования/расшифрования критичных данных

Для вычисления сессионного ключа S_{DEC} используются мастер-ключ K_{DEC} и константа *label* = 0181_{16}. На основе этих исходных данных получаем следующее значение сессионного ключа S_{DEC}:

S_{DEC} = 8*f*739*b*771*af*97*d*4294*cca*17338*b*2*ccc*5 \ \
9*a*14*d*4*cd*5930*fce*716*afa*0694*e*269053$_{16}$

А.3.2 Криптограмма аутентификации карты

Для вычисления криптограммы аутентификации карты используются сессионный ключ S_{ENC}, случайное число карты, случайное число хоста и счетчик сессий. На основе этих исходных данных получаем следующее значение криптограммы карты:

cardCryptogram = *b*845*e*5*f*95*f*37$_{16}$

А.3.3 Криптограмма аутентификации хоста

Для вычисления криптограммы аутентификации хоста используются сессионный ключ S_{ENC}, случайное число карты, случайное число хоста и счетчик сессий. На основе этих исходных данных получаем следующее значение криптограммы хоста:

terminalCryptogram = *eb*3203*fc*84*ab*$_{16}$

А.3.4 Генерация имитовставки C-MAC APDU-команды

Для вычисления имитовставки C-MAC для команды EXTERNAL AUTHENTICATE используются сессионный ключ S^{C}_{MAC} и следующие исходные данные:

$(CLA||INS||P_1||P_2||Lc||Data) = (84||82||13||00||06||eb3203fc84ab)_{16}$

$ICV = 00000000_{16}$

На основе этих исходных данных получаем следующее значение C-MAC:

C-MAC = $3b6ccdb4_{16}$

Тогда APDU-команда будет иметь вид:

EXTERNAL AUTHENTICATE = 848213000*aeb*3203*fc*84*ab*3*b*6*ccdb*4_{16}

А.3.5 Генерация имитовставки R-MAC APDU-команды

Для вычисления имитовставки R-MAC для APDU-команды используются сессионный ключ S^R_{MAC} и следующие исходные данные:

$(CLA||INS||P_1||P_2||Lc||Data) = (84||ca||13||00||06||119aba122190)_{16}$

Данные ответа: $(Li||R\text{-}Data) = (07||000120aa809012)_{16}$

$SW = 9000_{16}$

$ICV = 4aca4f14_{16}$

На основе этих исходных данных получаем следующее значение R-MAC:

R-MAC = $814dbd0f_{16}$

Тогда APDU-команда будет иметь вид:

APDU = 84*ca*130006119*aba*122190 \ \
07000120*aa*809012 \ \
$814dbd0f9000_{16}$

А.3.6 Шифрование поля данных APDU-команды

Команда APDU с имитовставкой R-MAC:

APDU = $84ca130006119aba12219007000120aa809012814dbd0f9000_{16}$

Следовательно, шифруются данные:

Data = $119aba122190_{16}$

R-Data = $000120aa809012_{16}$

После шифрования поля данных на сессионном ключе S_{ENC} команда принимает вид:

APDU = $84ca13000833dfc3b82e3bd06c0864fc317abf1062aa814dbd0f9000_{16}$

А.3.7 Шифрование критичных данных APDU-команды

Команда APDU с имитовставкой C-MAC и критичными данными:

APDU = 84*ca*130024 \ \
833*c*9066*e*2*e*037*db*9*c*089*a*1*f*4*c*64460*d* \ \
7*e*320*e*436230*f*8*a*005*db*4*fbdb*8*ef*24*c*8 \ \
$c93a286f_{16}$

Следовательно, шифруются критичные данные:

Data = 833*c*9066*e*2*e*037*db*9*c*089*a*1*f*4*c*64460*d* \ \
$7e320e436230f8a005db4fbdb8ef24c8_{16}$

После шифрования критичных данных на сессионном ключе S_{DEC} команда принимает вид:

APDU = 84*ca*130024 \ \
30*f*444*fca*2*aeb*993*fc*1*f*134*d*7*a*180*ad*5 \ \
*b*8*d*76*d*5*abd*22*b*7*d*7*e*096*d*1*bf*1*e*492*e*0*f* \ \
$c93a286f_{16}$

Библиография

[1] GlobalPlatform Card Specification Version 2.2.1 (Public Release January 2011 Document Reference: GPC_SPE_034)
[2] Security Upgrade for Card Content Management (GlobalPlatform Card Specification v2.2 — Amendment E)

УДК 681.3.06:006.354 ОКС 35.040

Ключевые слова: информационная технология, криптографическая защита информации, аутентификация, ключ, криптографический протокол, платежная карта, платежное приложение, защищенный обмен сообщениями, защищенный канал

БЗ 1—2018/153

Редактор *Н.А. Аргунова*
Технический редактор *И.Е. Черепкова*
Корректор *Е.Р. Ароян*
Компьютерная верстка *Л.В. Софейчук*

Сдано в набор 10.01.2018. Подписано в печать 13.02.2018. Формат 60×84 1/8. Гарнитура Ариал.
Усл. печ. л. 3,26. Уч.-изд. л. 2,95. Тираж 18 экз. Зак. 188.

Подготовлено на основе электронной версии, предоставленной разработчиком стандартов

ИД «Юриспруденция», 115419, Москва, ул. Орджоникидзе, 11.
www.jurisizdat.ru y-book@mail.ru

Издано и отпечатано во ФГУП «СТАНДАРТИНФОРМ», 123001, Москва, Гранатный пер., 4.
www.gostinfo.ru info@gostinfo.ru