ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

СКАНТЕК

Платежный аплет PayCORN

Версия 18.10

Сентябрь 2015

Россия, 119049, г. Москва, Донская ул., д. 15

Телефон: (499) 271-9661 • e-mail: 2b@scantech.ru

Список изменений.

12.08.2014

1. Введено понятие «Детерминированная особая ситуация». Обработка таких ситуаций определяется не эмитентом, а заранее определена в платежном приложении. В документ добавлен соответствующий раздел.

2. Переименован элемент данных IAF Issuer Policy (IAFIP), поскольку принято решение расширить его использование (не только для случая, когда аутентификация эмитента провалилась). Теперь он называется Issuer Policy (IP).

3. Добавлено описание особой ситуации «новая карта». В элементе данных Issuer Policy (IP) определен новый признак, который позволяет эмитенту указать, что особая ситуация «новая карта» должна считаться детерминированной.

4. Реализован другой способ индикации события, которое привело платежное приложение к изменению решения терминала об обработке транзакции. В результате обновлено описание элемента данных Issuer Application Data.

5. Уточнена логика обработки детерминированных особых ситуаций, что привело к достаточно существенному изменению описания команд платежного приложения GENERATE AC и GET PROCESSING OPTIONS.

6. Изменено назначение признака «PIN-код не установлен» в CVR. Теперь он используется только для информирования эмитента при онлайновой обработке транзакции и не может использоваться в качестве признака в CIAC.

7. Уточнено описание некоторых элементов данных в главе «Кодирование элементов данных».

8. Расширен список дополнительной документации.

9. Выполнена стилистическая правка документа, исправлены замеченные грамматические ошибки, сделана попытка улучшить отображение документа продуктами Adobe (но конвертирование документа в формат PDF пока не приводит к удовлетворительным результатам).

14.08.2014

1. Аплету присвоен другой AID. Поскольку RID аплета не зарегистрирован в ISO/IEC 7816-5 Registration Authority (TDC Services A/S), AID аплета должен начинаться с шестнадцатеричной цифры F, которая указывает, что это проприетарный, незарегистрированный RID.

19.08.2014

1. Уточнён алгоритм сброса признаков в CVR при выполнении второй команды GENERATE AC в ситуациях, когда аутентификация эмитента провалилась и когда аутентификация эмитента выполнена успешно.

04.10.2014

1. В список объектов персонализации платежного приложения добавлен новый необязательный объект персонализации Track 2 Equivalent Data.

15.10.2014

1. Сертификаты публичных ключей эмитента и карты, которые раньше были фиксированной длины, теперь могут иметь любую длину, определяемую эмитентом в соответствии с правилами EMV.

2. Секретный ключ карты (ICC Private Key) теперь может иметь любую длину, определяемую эмитентом в соответствии с правилами EMV, и представляется в обычном виде, а не в CRT-форме (т. е. в форме представления ключа в соответствии с теоремой о китайских остатках).

3. Для большинства объектов персонализации платежного приложения больше не используются значения по умолчанию.

4. Ликвидирована команда персонализации для получения длины буфера APDU.

17.10.2014

1. Уточнена логика обработки команды SELECT, которая доводится до платежного приложения, когда аплета с AID, указанным в команде, нет на карте. Теперь в этой ситуации возвращаются байты состояния 6A82.

23.11.2014

1. Объект персонализации Application Version Number теперь не является обязательным. Если он не определен при персонализации, то не предоставляется терминалу при чтении данных платежного приложения.

2. В список объектов персонализации платежного приложения добавлен новый необязательный объект персонализации PIN-блок в ISO Format 0.

29.11.2014

1. Платежное приложение теперь поддерживает глобальный PIN-код, для чего используются интерфейс CVM и Global PIN, определённые в спецификациях GlobalPlatform. В связи с этим унифицированы объекты персонализации, определяющие PIN-код. Во-первых, введен объект

«Признаки PIN-кода», который задает формат PIN-блока, если он определен, и область действия PIN-кода (глобальный или локальный PIN-код). Во-вторых, в объекте персонализации «Значение PIN-кода» теперь может быть определен PIN-блок в форматах ISO Format 1 или ISO Format 2. PIN-блок в ISO Format 0 временно не поддерживается, поскольку идет непростая борьба за размер аплета на карте.

2. Из-за особенностей реализации глобального PIN-кода объект персонализации «Максимальное количество предъявлений PIN-кода» стал необязательным объектом, для которого определено значение по умолчанию в платежном приложении.

04.12.2014

1. В разделе «Генерация сессионного ключа» главы «Криптографические алгоритмы» описан процесс получения сессионного ключа для Secure Messaging.

2. В главу «Криптографические алгоритмы» добавлен раздел «Secure Messaging», который содержит описание Secure Messaging, применяемого для команд критичного скрипт-процессинга.

3. Исправлено описание команды PIN CHANGE / UNBLOCK.

12.12.2014

1. Изменен формат элемента данных Application Interchange Profile (AIP), который определяет возможности карты по выполнению транзакции и требования карты к терминалу. Теперь во втором байте AIP определяется, что приложение использует EMV Mode для обработки бесконтактной транзакции.

2. Исправлены неточности в приложении «Объекты данных».

16.12.2014

1. Объект персонализации Application Effective Date теперь не является обязательным. Если он не определен при персонализации, то не предоставляется терминалу при чтении данных платежного приложения.

04.03.2015

1. Изменено количество записей, считываемых по команде READ RECORD, и порядок элементов данных в записях. Первые записи файла теперь содержит статические данные, которые должны быть аутентифицированы при выполнении транзакции в контактном режиме.

2. В соответствии с рекомендациями по составу статических данных, которые должны быть аутентифицированы, добавлен элемент данных Static Data Authentication Tag List, в который включен тэг 82 для Application Interchange Profile (AIP).

3. Изменена логика обработки детерминированной особой ситуации «PIN-код не установлен». Теперь не используется специальный CVM List, который содержит единственный метод верификации владельца карты – ввод PIN-кода в online. Вместо этого рекомендуется во время персонализации установить такой CVM List, чтобы терминал осуществлял ввод PIN-кода в online, когда ему предоставляется информация о том, что проверка PIN-кода в офлайновом режиме невозможна, поскольку исчерпан счетчик предъявлений PIN-кода.

4. В раздел «Восстановление публичного ключа карты» главы «Криптографические алгоритмы» добавлено описание формирования перечня статических данных, которые должны быть аутентифицированы, при вычислении значения хэш-функции сертификата публичного ключа карты.

14.03.2015

1. Платежное приложение теперь обеспечивает обработку команд с Secure Messaging Format 1. В связи с этим переработан раздел «Secure Messaging» главы «Криптографические алгоритмы» и обновлено описание команды PIN CHANGE/ UNBLOCK.

2. Полностью переработана команда GET DATA, которая теперь предоставляет информацию о составных объектах данных, содержащих элементарные объекты со значениями внутренних переменных платежного приложения. Соответствующие изменения внесены в приложение «Объекты данных».

3. Описаны новые элементы данных платежного приложения Card Decision Reason, Card Decision Explanation, Proprietary Version и Proprietary Features.

4. Вместо 4-х байтных лимитов и аккумуляторов сумм платежных операций, определенных в двоичном формате, теперь используются 6-ти байтные лимиты и аккумуляторы в двоично-десятичном формате n12. В связи с этим изменены длины значений соответствующих объектов персонализации и из списка CDOL1 удален объект Amount, Authorized (Binary).

5. Изменен формат области счетчиков и области с данными на усмотрение эмитента в элементе данных Issuer Application Data.

6. Изменен формат элемента данных Terminal Transaction Processing Information (TTPI). Из данных этого элемента удалены три последних зарезервированных байта и теперь он имеет длину 2 байта. В связи с этим изменен список PDOL.

7. Расширен перечень кодов возврата, возвращаемых командой STORE DATA, чтобы облегчить анализ ошибочных ситуаций, возникающих при персонализации платежного приложения.

8. Теперь признак «Заблокировать приложение», устанавливаемый эмитентом в CSU, обрабатывается платежным приложением точно так же, как и признак «Заблокировать карту».

03.04.2015

1. Жестокая реальность опровергла все надежды на то, что эмитент будет формировать только команды критичного скрипт-процессинга. Поэтому в текст всего документа внесены изменения, связанные с тем, что платежное приложение обрабатывает не только команды критичного скрипт-процессинга, но и команды некритичного скрипт-процессинга.

2. Изменена логика формирования признаков в Card Verification Results (CVR). Теперь в CVR не формируются признаки, которые предназначены только для эмитента, а CVR всегда отражает текущее состояние транзакции. В связи с этим изменился принцип формирования CVR, что нашло своё отражение в описании команд GENERATE AC и GET PROCESSING OPTIONS (в бесконтактном режиме).

3. В Card Issuer Action Code (CIAC) появились новые признаки, т. к. CVR теперь отражает текущее состояние транзакции и нет признаков, которые предназначены только для эмитента.

4. Из документа удалена глава «Особые ситуации». Информация из этой главы перенесена в отдельные разделы главы «Команды аплета».

5. В главу «Команды аплета» добавлен раздел «Отображение текущего состояния в CVR», который описывает, каким образом в CVR устанавливаются признаки текущего состояния транзакции.

6. Во второй команде GENERATE AC теперь разграничиваются ситуации Unable to Go Online и не получены данные аутентификации эмитента. В случае, когда не получены данные аутентификации эмитента, транзакция отклоняется и возвращается криптограмма AAC. В связи с этим уточнено описание команды GENERATE AC, изменено описание элемента данных Authorization Response Code, и расширен перечень кодов, определяющих причину, по которой карта изменила решение терминала об обработке транзакции (введен код «Не предоставлены данные аутентификации эмитента»).

10.04.2015

1. Теперь используется другое представление лимитов в составных объектах с тэгами BF30 и BF35. В связи с этим соответствующие изменения внесены в команду GET DATA и приложение «Объекты данных».

2. Разработана новая команда PUT DATA, позволяющая модифицировать определенные данные платежного приложения, которые идентифицируются тэгами (эти теги используются в команде GET DATA для получения данных платежного приложения, которые не считываются с помощью команды READ RECORD).

3. В раздел «Secure Messaging» главы «Криптографические алгоритмы» добавлено описание объекта с тэгом 81, в который содержит не конфиденциальные данные, передаваемые вместе с командой PUT DATA.

4. Исправлены неточности в реализации некритического скрипт-процессинга.

5. В объекте персонализации «Значение PIN-кода» теперь может быть определен PIN-блок в любом формате ISO (0, 1, 2 или 3), а в объекте персонализации «Признаки PIN-кода» не требуется определять формат PIN-блока.

16.04.2015

1. Исправлена ошибка при обработке команд некритичного скрипт-процессинга. В связи с этим немного откорректировано описание команды GENERATE AC.

23.04.2015

1. Исправлены отдельные неточности, которые были замечены при чтении документа.

01.05.2015

1. В платежном приложении теперь контролируется кратность двум длины модуля ключа карты. В связи с этим внесены изменения в главу «Персонализация».

2. Поскольку изменена длина и определены новые признаки в Issuer Policy (IP), переработан раздел «Issuer Policy (IP)» в главе «Кодирование элементов данных».

3. Уточнена логика обработки детерминированных ситуаций, описанная в разделе «Детерминированные особые ситуации».

4. Изменена длина объекта персонализации Issuer Policy (IP) с DGI23. Теперь длина значения этого объекта персонализации может находиться в диапазоне от 1 до 3 (байты, которые не определены в значении объекта, заполняются нулями).

5. В команде GET PROCESSING OPTIONS в бесконтактном режиме ситуации «новая карта» и «PIN-код не установлен» теперь могут быть проигнорированы и не будут являться препятствием для выполнения бесконтактной обработки, если установлены соответствующие признаки в Issuer Policy.

6. Изменилась длина составного объекта с тэгом BF3B, который возвращается командой GET DATA и используется для информирования об опциях эмитента, так как длина Issuer Policy (IP) теперь равна 3.

7. Определено новое значение по умолчанию для объекта персонализации Terminal Transaction Processing Information (TTPI). Теперь значение по умолчанию не содержит признака «Терминал считает, что требуется верификация владельца карты».

28.05.2015

1. В Issuer Policy (IP) имплементированы все опции эмитента, обязательные для Common Core Definitions, и реализована обработка транзакции в соответствии с опциями эмитента.

2. Уточнена логика установки и сброса признаков исключительных ситуаций, а также признаков и полей CVR.

3. Реализован новый алгоритм установки состояний «Card Block» и «Application Block», а также уточнена логика обработки команд после установки этих состояний. В документ добавлены разделы «Заблокированное платежное приложение» и «Заблокированная карта».

4. Реализована команда APPLICATION UNBLOCK, которая используется для снятия ограничения на использование платежного приложения, когда платежное приложение было заблокировано эмитентом при установке соответствующего бита в CSU.

5. В команде GENERATE AC теперь не проверяется установка резервных бит параметра P1 в 0.

6. Теперь в данных сертификата Signed Dynamic Application Data формируется уникальный ICC Dynamic Number каждый раз, когда в команде GENERATE AC запрашивается подпись CDA (ICC Dynamic Number отличается в первой и второй командах GENERATE AC).

7. Решение терминала об отклонении транзакции теперь не является причиной для невыполнения процедур Card Risk Management.

8. Полностью изменен алгоритм подсчета количества успешно выполненных команд с Secure Messaging и установка признаков особых ситуаций при выполнении скрипт-процессинга.

9. Второй байт AIP для контактного режима теперь всегда равен 0.

10. В команде персонализации STORE DATA значения конфиденциальных объектов персонализации теперь могут быть зашифрованы на сессионном ключе шифрования конфиденциальных данных, а устанавливаемый уровень безопасности канала безопасности может не включать шифрование данных.

11. Добавлен новый объект персонализации с DGI 45, который используется для определения ICC Private Key Modulus, если он представлен в задании на персонализацию в зашифрованном виде и в команде STORE DATA применяется зашифрование значений конфиденциальных объектов персонализации на сессионном ключе шифрования конфиденциальных данных.

12. Полностью переработана глава «Персонализация». Из неё удалено описание метода персонализации, применяемого в СКАНТЕК. Определены дополнительные требования по персонализации платежного приложения.

09.06.2015

1. Изменён принцип формирования Card Decision Reason и, главное, алгоритм предоставления информация о причине, по которой карта изменила решение терминала об обработке транзакции.

2. Исправлена ошибка, которая заключалась в том, что случайное число, сгенерированное командой GET CHALLENGE не сбрасывалось сразу после выполнения команды VERIFY, в связи с чем две последовательные команды VERIFY могли использовать одно и то же случайное число.

3. Счетчик успешно выполненных команд скрипта с Secure Messaging теперь сбрасывается перед выполнением очередной транзакции и только в том случае, когда при выполнении предыдущей транзакции не был выполнен некритичный скрипт-процессинг и существовали условия для сброса счетчика (была выполнена успешная аутентификация эмитента или в Issuer Policy разрешен сброс счетчика, если аутентификация эмитента не выполнена или аутентификация эмитента провалилась).

4. Исправлена ошибка, когда сформированная криптограмма AAC сохранялась для дальнейшего использования в скрипт-процессинге в состоянии «Приложение заблокировано» независимо от того, в какой команде GENERATE AC формировалась криптограмма AAC, в результате чего команды некритичного скрипт-процессинга завершались неуспешно после блокирования приложения, если во второй команде GENERATE AC терминал запрашивал криптограмму AAC (теперь криптограмма AAC сохраняется для дальнейшего использования в скрипт-процессинге только в том случае, когда транзакция отклонена первой командой GENERATE AC в состоянии «Приложение заблокировано»).

5. Реализована новая логика проверки возможности использования команды скрипта в текущем контексте. Теперь даже в том случае, когда команда скрипт-процессинга не ожидается, фиксируется ситуация «Скрипт-процессинг провалился».

6. Внесены дополнительные пояснения в раздел «Команды персонализации», так как некоторые аспекты применения команды STORE DATA вызвали вопросы.

03.07.2015

1. В Issuer Policy добавлена новая опция эмитента, с помощью которой ситуация «PIN-код не установлен» может игнорироваться и не рассматриваться в качестве детерминированной ситуации в контактном режиме обработки транзакции.

2. В команде GET PROCESSING OPTIONS теперь не осуществляется автоматическое блокирование платежного приложения, если ATC достиг максимума на предыдущей транзакции.

3. Первый байт AIP для бесконтактного режима обработки транзакции теперь содержит корректную информацию о возможностях карты (в нем установлен единственный бит «Терминал должен выполнить процедуры управления рисками»).

4. Теперь скрипт-процессинг допускается после выполнения первой команды GENERATE AC независимо от того, какая криптограмма возвращается терминалу (для генерации сессионного ключа скрипт-процессинга может использоваться криптограмма любого типа).

5. Исправлена ошибка для Global PIN, из-за которой CVM State всегда переводился в состояние ACTIVE из состояний INVALID_SUBMISSION или VALIDATED при выборе аплета (тем самым игнорировался результат предъявления PIN-кода в текущем сеансе работы с картой другим аплетом, использующим глобальный интерфейс CVM).

6. Изменена логика обработки Global PIN и теперь в CVR всегда отображаются признаки предъявления PIN-кода (успешного или неуспешного) даже в том случае, когда в текущем сеансе работы с картой PIN-код был предъявлен другим аплетом, использующим глобальный интерфейс CVM.

7. Теперь ситуации «PIN-код не установлен» и «PIN-код заблокирован» не являются препятствием для установки в CVR признака «Получена команда проверки PIN-кода».

8. Исправлена ошибка, из-за которой в CVR устанавливался бит «Комбинированная аутентификация (CDA) выполнена» даже в том случае, когда карта возвращала криптограмму AAC и не предоставляла объект Signed Dynamic Application Data.

9. При выполнении первой команды GENERATE AC теперь не проверяется активность расходования средств в offline, если запрашивается криптограмма AAC.

10. После обновления офлайновых счетчиков и аккумуляторов во второй команде GENERATE AC теперь устанавливаются соответствующие им признаки в CVR.

11. Если эмитент не требует обновления офлайновых счетчиков и аккумуляторов, то в CVR устанавливаются признаки, соответствующие их текущим значениям (раньше признаки отражали состояние с учетом текущей транзакции).

12. В Issuer Policy добавлена новая опция эмитента, с помощью которой признаки «Превышен верхний предел офлайновых транзакций» и «Превышен верхний предел суммы всех последовательных офлайновых транзакций», установленные в CVR, могут быть проигнорированы для терминала типа 26 (unattended offline-only POS terminal), даже если в CIAC-Default определено, что в этих ситуациях транзакция должна быть отклонена.

13. Уточнена логика сброса офлайновых счетчиков и аккумуляторов, если такой сброс запрашивается эмитентом в ситуации, когда аутентификация эмитента не выполнена или провалилась.

06.09.2015

1. В Issuer Policy добавлена новая опция эмитента «Специальная обработка транзакции в другой валюте», позволяющая обработать транзакцию, валюта которой отличается от валюты платежного приложения, в офлайновом режиме.

Оглавление

Основные понятия 1
Элементы и объекты данных 2
Транзакция в контактном режиме 4
Данные на карте 8
Вопросы безопасности 9
Верификация владельца карты 13
Управление рисками терминала 15
Списки объектов данных 18
Управление рисками карты 19
Аутентификация карты 23
Онлайновая обработка транзакции 24
Транзакция в бесконтактном режиме 27
Персонализация 32
Общая схема персонализации 34
Безопасный обмен данными с картой 36
Два метода персонализации 39
Выбор платежного приложения 42
Контактный режим 44
Бесконтактный режим 46
Заблокированное платежное приложение 48
Заблокированная карта 49
Команды аплета 50
Кодирование элементов данных 52
Криптографические алгоритмы 54
Приложения 56
Объекты данных 57
Общие коды ошибок 58
Дополнительная документация 59

ГЛАВА 1

Основные понятия

Платежный аплет PayCORN – это EMV-совместимое приложение, которое удовлетворяет спецификациям Common Core Definitions (CCD). Спецификации CCD появились в версии 4.1 стандарта EMV, утвержденной в мае 2004 г. Спецификации CCD задают набор и формат данных, направляемых терминалом карте, определяют особенности обработки транзакции, а также унифицируют способ обмена данными между хостом банка и платежной сетью.

Платежный аплет PayCORN (в дальнейшем, просто *платежный аплет* или *платежное приложение*) поддерживает платежи в контактном и бесконтактном режимах, т. е. может быть размещен как на контактной карте (удовлетворяющей спецификациям ISO/IEC 7816), так и бесконтактной карте (взаимодействие с которой осуществляется на основе протоколов, описанных в ISO 14443). Платежный аплет выбирает режим обработки транзакции (контактный или бесконтактный) автоматически в зависимости от метода использования карты. Поэтому платежный аплет может быть размещен и на карте с двумя видами интерфейсов – контактным и бесконтактным (dual interface card).

Обработка транзакции платежным аплетом в контактном и бесконтактном режимах различается, поскольку в контактном режиме аплет работает в соответствии со спецификациями CCD, а в бесконтактном – по спецификациям EMV Contactless Specifications for Payment Systems – Entry Point Specification. Это связано с тем, что в бесконтактном режиме из-за определенных ограничений нельзя полностью поддержать тот же алгоритм обработки транзакции, что в контактном режиме.

Чтобы разобраться в особенностях реализации платежного аплета, необходимо сначала остановиться на основных принципах обработки транзакции в контактном и бесконтактном режимах.

Элементы и объекты данных

Любое приложение карты использует некоторый набор элементов данных. Элементы данных являются логическими структурами. Для хранения в памяти они отображаются в физические объекты данных. Существуют различные способы отображения элементов данных в объекты данных. В спецификациях EMV используется кодирование BER-TLV, формализованное в стандарте ISO/IEC 8825.

В соответствии с ISO/IEC 8825 любой объект данных определяется двумя или тремя полями: тэгом (Tag), длиной (Length) и значением (Value). Поле значения должно быть опущено, если длина равна 0.

Поле тэга состоит из одного или более байтов (в спецификациях EMV – один или два байта). Структура поля тэга показана в следующих таблицах.

Структура первого байта поля тэга.

8	7	6	5	4	3	2	1	Значение
x	x							Класс объекта данных
		0						Примитивный объект
		1						Составной объект
			1	1	1	1	1	Имеются другие байты тэга
			x	x	x	x	x	Номер тэга

Структура последующих байтов поля тэга.

8	7	6	5	4	3	2	1	Значение
1								Имеются другие байты тэга
0								Последний байт поля тэга
	x	x	x	x	x	x	x	Номер тэга (больше 0)

Два первых бита первого байта поля тэга определяют класс объекта данных следующим образом:

- 00 – универсальный класс
- 01 – прикладной класс, который содержит объекты, относящиеся к определенной индустрии (например, индустрии финансовых расчетов по карточкам)

- 10 – контекстно-определенный класс (содержит объекты, определенные для некоторого стандарта)
- 11 – частный класс, содержащий объекты, которые определены конкретными спецификациями

Для группирования данных используются составные объекты данных, которые могут содержать другие составные объекты и примитивные объекты. В спецификациях EMV составной объект данных называется template (например, FCI Template).

Поле длины объекта данных определяет количество байт в поле значения объекта. В стандарте EMV поле длины задается одним, двумя или тремя байтами. Если старший бит самого левого байта поля длины равен 0, то поле длины занимает один байт и определяет длину значения от 0 до 127. Если старший бит равен 1, то последующие биты определяют количество дополнительных байтов, используемых для представления поля длины.

Описанный выше метод кодирования данных BER-TLV позволяет для конкретного объекта данных однозначно определить его идентификатор, размер и значение. Таким образом, кодировка BER-TLV является универсальным средством представления данных.

Все объекты данных, используемые платежным аплетом и терминалом для обработки транзакции приведены в главе «Приложения».

Транзакция в контактном режиме

На рис. 1 показана общая схема обработки транзакции в контактном режиме. Приведенная диаграмма не отображает все особенности карточной операции в терминале, но позволяет понять, каким образом в этой операции участвует карта (платежный аплет). Кроме того, в нижеследующем описании обработки транзакции в контактном режиме отсутствуют важные детали, которые опущены, поскольку целью документа является не определение процесса обработки транзакции, а описание процесса взаимодействия терминала и карты.

Платежная операция начинается с выбора терминалом платежного приложения на карте. Для выполнения транзакции необходимо, чтобы терминал поддерживал платежное приложение, которое находится на карте. На этапе выбора приложения терминал проверяет факт наличия такого приложения (если на карте несколько приложений, поддерживаемых терминалом, то терминал и, возможно, владелец карты выбирают, какое из них следует использовать для выполнения текущей транзакции). Для выбора приложения используется команда SELECT, которая передает терминалу некоторые данные о выбранном приложении.[1]

После того как приложение выбрано, терминал инициирует транзакцию. Для этого используется команда GET PROCESSING OPTIONS, с помощью которой терминал сообщает карте данные, необходимые ей для того, чтобы определиться с особенностями выполняемой операции.[2] В ответе на команду GET PROCESSING OPTIONS карта сообщает терминалу о своих возможностях по выполнения транзакции и требованиях к терминалу (через Application Interchange Profile – AIP) и предоставляет ссылки на данные, которые терминал должен прочитать, чтобы успешно выполнить транзакцию (определяются в Application File Locator – AFL). Подробнее об элементах данных AIP и AFL см. в главе «Кодирование элементов данных».

Затем терминал считывает все записи, указанные картой, с помощью команд READ RECORD и приступает к выполнению процедур проверки ограничений по применению карты (номера версии приложения, срока действия карты и т. п.) и проверке стоп-листов.

Терминал просматривает список методов верификации владельца карты, предоставленный картой, выбирает подходящий метод и верифицирует владельца карты.

---

[1] Эти данные называются File Control Information (FCI) и содержат имя приложения (AID), метку приложения и предпочтительные языки общения терминала с владельцем карты. Более подробно, данные FCI описаны в главе «Выбор приложения».

[2] В контактном режиме для инициирования транзакции никакие данные от терминала не нужны и с командой GET PROCESSING OPTIONS никакие данные не передаются.

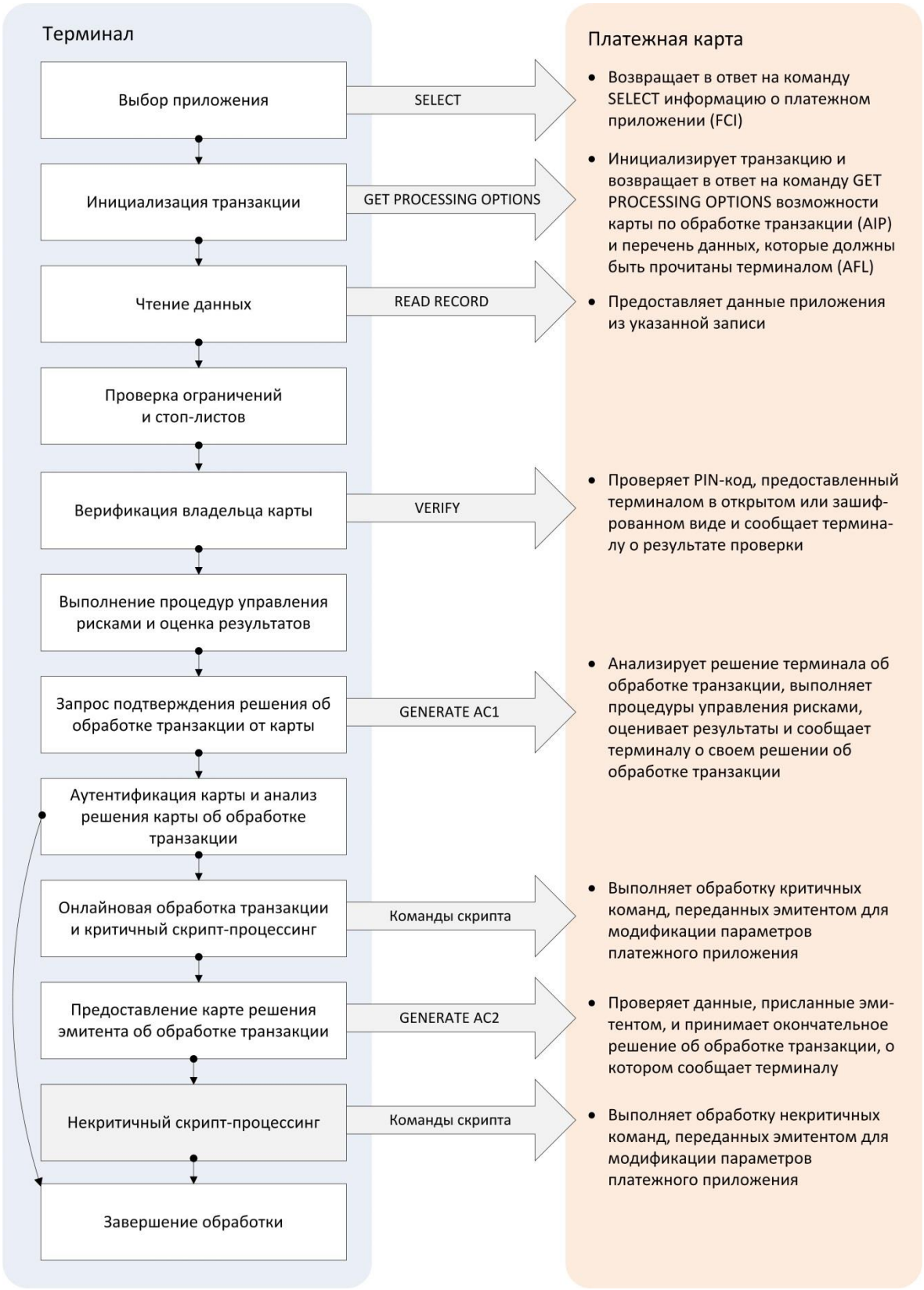

Рис. 1. Обработка транзакции в контактном режиме.

Терминал по указанию карты может также выполнить процедуры управления рисками, контролирующие сумму и количество последовательных транзакций, выполненных в offline.

Далее терминал производит анализ результатов выполненных им проверок с использованием критериев, сформулированных терминалу обслуживающим банком и эмитентом карты. В результате терминал принимает одно из трех решений об обработке транзакции:

1. Одобрить транзакцию в офлайновом режиме.

2. Передать транзакцию на авторизацию эмитенту карты.

3. Отвергнуть транзакцию в офлайновом режиме.

О своём решении терминал сообщает карте с помощью первой команды GENERATE AC. Вместе с командой GENERATE AC карте передаются данные транзакции, результаты процедуры управления рисками терминала и реквизиты терминала. На основе полученных данных карта выполняет собственные процедуры управления рисками и производит анализ результатов выполненных ею проверок с использованием критериев, сформулированных терминалу эмитентом карты. В результате карта принимает собственное решение об обработке транзакции – одно из трех решений, описанных выше, но ранг решения карты всегда не ниже ранга решения терминала (например, если терминал принял решение отвергнуть транзакцию, то карта не имеет права изменить это решение). В EMV эта зависимость является фундаментальной и контролируется как картой, так и терминалом.

Если карта принимает решение о том, что транзакция должна быть отправлена на авторизацию эмитенту, то карта формирует специальную криптограмму, представляющую собой подпись реквизитов транзакции, карты и терминала. Во всех остальных случаях формируются криптограммы, информирующие о завершении транзакции (одобрении или отклонении транзакции в офлайновом режиме). Криптограмма заносится в специальный пакет данных, который зашифровывается на ключе карты и служит для аутентификации карты терминалом.

Терминал, получив данные от карты по команде GENERATE AC, выполняет аутентификацию карты (расшифровывает пакет данных с криптограммой и убеждается, что данные в пакете корректны), после чего проверяет, какую криптограмму вернула карта. Если возвращена криптограмма одобрения или отклонения транзакции в офлайновом режиме, то обработка на этом завершается. Когда транзакция должна быть отправлена на авторизацию эмитенту, терминал отправляет (через хост обслуживающего банка) криптограмму и другие относящиеся к транзакции данные эмитенту. Получив данные, эмитент проверяет криптограмму (аутентифицирует карту) и выполняет ряд других проверок. По результатам этих проверок эмитент

принимает решение о том, отклонить или одобрить транзакцию, и, в свою очередь, формирует криптограмму, которая используется картой для аутентификации эмитента. Ответ эмитента отправляется обслуживающему банку, который направляет его терминалу.

В ответе эмитента могут присутствовать команды скрипт-процессинга, предназначенные для карты. С помощью этих команд эмитент имеет возможность менять параметры платежного приложения, разблокировать или изменить PIN-код, заблокировать приложение карты. Эмитент может присвоить каждой команде скрипт-процессинга статус критичной (команда направляется карте сразу после получения её терминалом до выполнения второй команды GENERATE AC) или некритичной (команда передается карте после выполнения второй команды GENERATE AC).

Терминал, получив ответ эмитента, передает карте решение эмитента, криптограмму эмитента и уточненные данные своих проверок во второй команде GENERATE AC. Карта аутентифицирует эмитента путем проверки криптограммы. Если аутентификация эмитента прошла успешно, то карта выполняет решение эмитента о завершении транзакции и формирует криптограмму одобрения или отклонения транзакции. Когда аутентификация эмитента провалилась, транзакция обычно картой отвергается.

Ниже приводится более подробное описание отдельных этапов обработки транзакции и элементов данных, которыми обмениваются терминал и карта.

Данные на карте

В соответствии со стандартом ISO 7816-4 на карте существует два вида файлов:

- файлы DF (Dedicated File), или каталоги, которые могут содержать другие DF–файлы или EF-файлы (Elementary File). Являясь каталогами, DF–файлы не содержат данные, а являются так называемыми точками доступа к другим файлам. Информация о DF–файле содержится в заголовке файла, который называется File Control Information (FCI)

- файлы EF (Elementary File), содержащие данные карты и её приложений. Каждый EF-файл должен принадлежать одному из DF-файлов

Реально, на современных картах нет физических объектов файловой структуры (DF и EF) – это только логические понятия. Можно определить следующие связи логических и физических объектов на карте:

- DF –это приложение на карте. Именем DF является идентификатор приложения, который называется AID (Application Identifier). Для выбора приложения на карте используется команда SELECT. В ответе на команду SELECT приложение возвращает FCI

- EF – это логический набор данных, которому соответствует определенный идентификатор (Short File Identifier – SFI). EF состоит из нескольких логических записей, каждой из которых присвоен номер. Не только EF, но и любая запись могут быть представлены на карте в виде отдельных экстентов памяти, связанных только логически

Данные, которые необходимы для выполнения транзакции, считываются терминалом из записей файлов платежного приложения по команде READ RECORD. Но не все данные, которые могут потребоваться терминалу, расположены в записях файлов. Некоторые данные хранятся в виде отдельных объектов и при необходимости терминал извлекает их с карты с помощью команды GET DATA.

Вопросы безопасности

Важнейшим свойством платежного приложения является использование криптографических функций для повышения безопасности финансовых операций.

Основные задачи для повышения безопасности финансовых операций, решаемые приложением, следующие.

1. Обеспечение аутентификации приложения на карте (чаще говорят аутентификации карты, хотя это и не совсем верно). Под аутентификацией карты понимается процесс доказательства её подлинности, т. е. того факта, что карта эмитирована банком, авторизованным на эмиссию карт. В случае офлайновой транзакции эмитент полностью делегирует карте функцию принятия решения по результату выполнения операции. Очевидно, что только решениям карты, подлинность которой доказана, можно доверять. Поэтому так важна её надежная аутентификация. Аутентификация карты осуществляется терминалом и (или) эмитентом. В офлайновых операциях аутентификация карты производится только терминалом и называется офлайновой аутентификацией. В случае онлайновой транзакции аутентификация карты может осуществляться и терминалом, и эмитентом. Аутентификация карты, выполняемая эмитентом, называется онлайновой аутентификацией.

2. Обеспечение аутентификации эмитента и проверки целостности присылаемых им данных за счёт проверки подписей данных, формируемых эмитентом.

3. Гарантирование эмитенту невозможности отказа держателя его карты от результата выполненной операции. Это обеспечивается тем, что по каждой выполненной транзакции эмитент получает от карты в своё распоряжение криптограмму приложения, которая представляет собой подпись наиболее критичных данных транзакции. Соответствие криптограммы данным транзакции подтверждает факт её выполнения.

4. Проверка целостности обмена критическими данными между картой и терминалом.

5. Обеспечение конфиденциальности данных в информационном обмене между картой и эмитентом, картой и терминалом.

6. Предоставление механизма надежной верификации владельца карты с помощью офлайновой проверки PIN-кода.

Платежное приложение реализует функции обеспечения безопасности транзакции на используемых в стандарте EMV асимметричном алгоритме шифрования RSA и симметричном алгоритме шифрования DES.

Сначала рассмотрим асимметричный алгоритм шифрования RSA. Если отвлечься от терминологии RSA, которая иногда может только запутать, то асимметричное шифрование основывается на том, что существуют два ключа – публичный и секретный. Если данные зашифрованы на публичном ключе, то они могут быть расшифрованы на секретном ключе и наоборот. Необходимость применения асимметричных алгоритмов шифрования в процедурах обеспечения безопасности вызвана тем, что терминал не должен обладать секретами карты (симметричное шифрование всегда подразумевает знание участниками информационного обмена общего секрета).

Для того чтобы терминал мог использовать асимметричные ключи карты для реализации функций обеспечения безопасности, необходимо создание PKI-инфраструктуры (PKI – Public Key Infrastructure). PKI-инфраструктура платежной системы в соответствии со стандартом EMV показана на рис. 2.

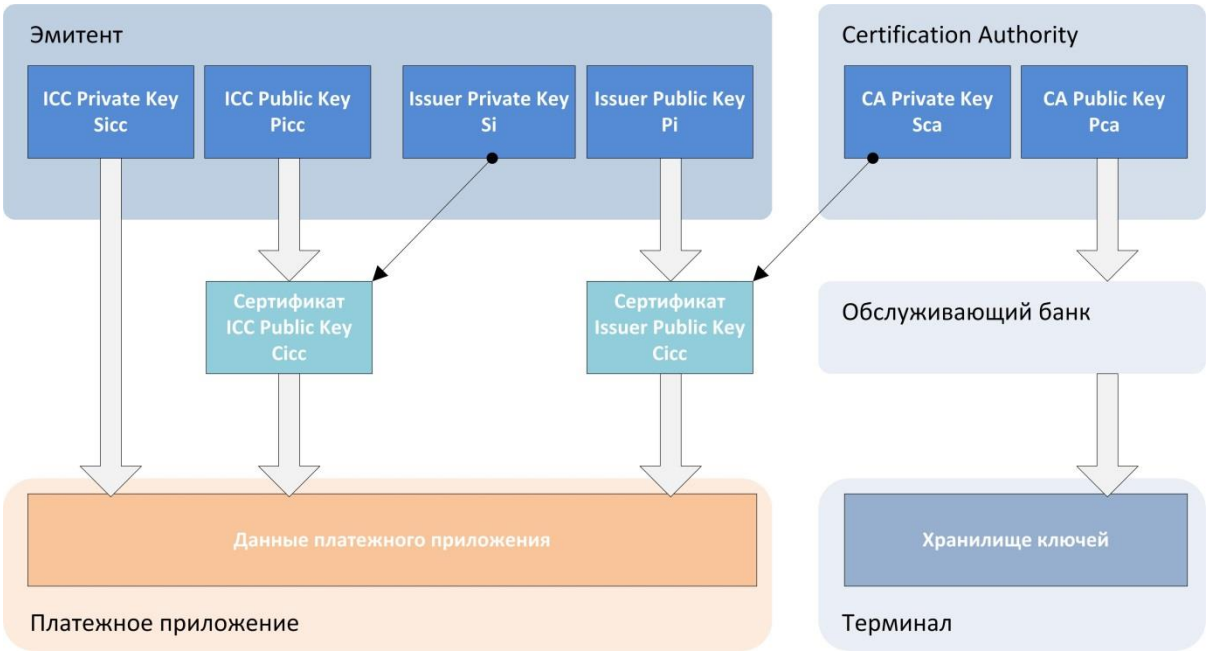

Рис. 2.PKI-инфраструктура платежной системы.

Корнем PKI-инфраструктуры является центр сертификации (Certification Authority – CA) платежной системы. Центр сертификации генерирует пары открытых и секретных ключей RSA и рассылает открытые ключи обслуживающим банкам для их загрузки в терминалы платежной системы.

На втором уровне дерева PKI-инфраструктуры находятся центры сертификации эмитентов, являющихся участниками платежной системы. Эти центры также генерируют пары открытых и секретных ключей RSA и получают в центре сертификации платежной системы сертификаты своих открытых ключей. Сертификат открытого ключа представляет собой реквизиты открытого ключа (включая и сам открытый ключ), его срок

действия, идентификатор эмитента и т. п., подписанные на секретном ключе центра сертификации платежной системы.[1]

Наконец, на нижнем уровне инфраструктуры располагаются открытые и секретные ключи карт эмитента. Открытые ключи карт сертифицируются (подписываются) в центре сертификации эмитента на секретном ключе центра сертификации эмитента.

В процессе персонализации платежного приложения на карту записываются сертификаты открытого ключа эмитента и открытого ключа карты, а также секретный ключ карты.

Для того чтобы получить открытый ключ карты, терминал считывает с карты сертификаты открытых ключей эмитента и карты. Далее с помощью открытого ключа CA, хранящегося в терминале, терминал проверяет правильность сертификата открытого ключа эмитента.[2]

После доказательства правильности сертификата открытого ключа эмитента терминал с помощью восстановленного открытого ключа эмитента проверяет правильность сертификата открытого ключа карты. Если сертификат корректен, то терминал получает доступ к открытому ключу карты. Открытый ключ карты используется терминалом для аутентификации приложения на карте и зашифрования конфиденциальных данных, пересылаемых приложению (например, PIN-кода). Аутентификация приложения осуществляется путем проверки подписи определенных данных, сформированной картой на секретном ключе карты. Если подпись карты верна, то это означает, что она была сделана секретным ключом карты, соответствующим её открытому ключу, сертифицированному эмитентом. Это в свою очередь означает, что карта была эмитирована банком-эмитентом, открытый ключ которого был сертифицирован центром сертификации платежной системы. Таким образом, доказывается факт выпуска карты банком, уполномоченным эмитировать карты платежной системы.

Для формирования криптограмм приложения, подписей данных, передаваемых между эмитентом и картой, а также для зашифрования обмена конфиденциальными данными между ними, используется симметричный алгоритм 3DES (ISO 11568-2). В этом случае участники информационного обмена имеют общий секрет – секретный ключ 3DES. Вернее, набор

[1] В терминологии EMV для формирования сертификатов используется функция Sign на секретном ключе. Эта функция зашифровывает сертификат на секретном ключе и он может быть расшифрован только на открытом ключе, соответствующем секретному ключу.

[2] Сертификат расшифровывается на публичном ключе. В терминологии EMV эта функция называется Recover. В результате расшифрования сертификата терминал получает доступ к открытому ключу эмитента, который хранится в сертификате. Но в сертификате хранится не весь открытый ключ эмитента, а только его часть. Другая часть этого ключа извлекается из объекта данных, который называется Issuer Public Key Remainder.

секретных ключей, поскольку в целях безопасности платежное приложение и эмитент используют три ключа:

- ключ Ka (Application Cryptogram Key) – используется для вычисления криптограмм приложения
- ключ Ki (MAC Key) – применяется для вычисления подписи данных (Message Authentication Code – MAC) в командах скрипт-процессинга
- ключ Kc (Encipherment Key) – используется для зашифрования конфиденциальных данных в командах скрипт-процессинга

Эмитент хранит только мастер-ключи симметричного криптографического алгоритма, которые в процессе персонализации карты преобразуются (диверсифицируются) в уникальные ключи платежного приложения (диверсификация мастер-ключа эмитента с использованием Application PAN и Application PAN Sequence Number описана в главе «Криптографические элементы»). Таким образом, каждое платежное приложение использует для криптографических операций набор уникальных ключей. Более того, уникальный ключ платежного приложения диверсифицируется для каждой транзакции в сессионный ключ. Для получения сессионного ключа используется Application Transaction Counter (ATC) – счетчик, который увеличивается при выполнении каждой транзакции (процесс генерации сессионного ключа описана в главе «Криптографические элементы»).

В результате для каждой транзакции карта и эмитент используют уникальный ключ симметричного криптографического алгоритма, что значительно повышает безопасность.

Верификация владельца карты

Верификация владельца карты выполняется для того, чтобы установить факт идентичности лица, получившего карту от её эмитента (клиента банка), с лицом, совершающим по карте операцию.

Ключевым объектом данных для верификации владельца карты является объект Cardholder Verification Method (CVM) List, который хранится на карте. Этот объект определяет список методов и условий верификации владельца карты (см. главу «Кодирование элементов данных»).

Одним из основных методов верификации владельца карты является процедура проверки PIN-кода. Проверка PIN-кода хотя и является только одним из возможных методов верификации, но получает все большее распространение в силу своей надёжности и эффективности.

Существует два различных метода офлайновой проверки PIN-кода:

- проверка PIN-кода, передаваемого на карту в открытом виде
- проверка PIN-кода, передаваемого на карту в зашифрованном виде

Офлайновая проверка PIN-кода всегда начинается с того, что терминал выдает команду GET DATA для получения объекта данных PIN Try Counter. Значение этого объекта – это количество оставшихся попыток ввода PIN-кода. Если у владельца карты еще остались попытки для ввода PIN-кода, то терминал получает от владельца карты значение его PIN-кода (от 4-х до 12-ти цифр) и формирует из PIN-кода PIN-блок в ISO Format 2, который представлен на рис. 3.

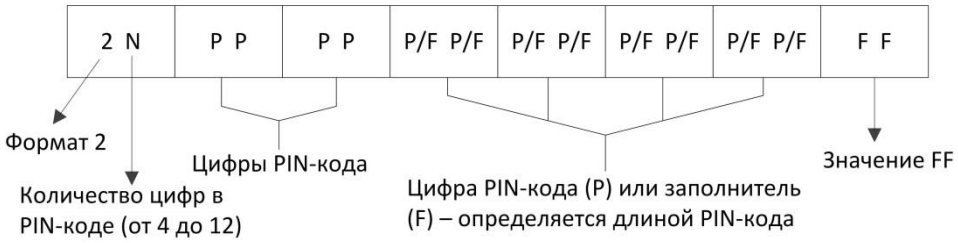

Рис. 3.PIN-блок в ISO Format 2.

Если в CVM указано, что PIN-код должен быть предъявлен в незашифрованном виде, то терминал передает на карту команду VERIFY, в поле данных которой содержится значение PIN-блока в ISO Format 2. Когда PIN-код должен быть предъявлен в зашифрованном виде, то сначала терминал получает от карты случайное число с помощью команды GET CHALLENGE, затем формирует сертификат PIN-кода, который содержит PIN-блок и полученное случайное число, и зашифровывает сертификат на открытом ключе карты. Зашифрованный сертификат PIN-кода подается на

карту в качестве данных команды VERIFY. Карта сравнивает полученное значение PIN-кода со значением, хранимым на карте (предварительно расшифровывая сертификат PIN-кода на секретном ключе карты и проверяя сертификат, когда это необходимо). Если они совпадают, то проверка PIN-кода считается выполненной успешно. Каждый раз, когда предъявлен неправильный PIN-код, количество оставшихся попыток ввода PIN-кода уменьшается на 1 (при достижении нуля PIN-код будет заблокирован).

Онлайновая проверка PIN-кода никак не связана с картой. Устройство ввода PIN-кода запрашивает PIN-код от владельца карты и зашифровывает его по специальному алгоритму, используя алгоритм DUKPT (Derived Unique Key Per Transaction), который описан в главе «Криптографические алгоритмы». Верификация владельца карты на этом завершается, поскольку значение PIN-кода проверяет эмитент, а не карта.

Управление рисками терминала

Процедуры управления рисками, выполняемые терминалом, являются элементом обеспечения безопасности платежных операций и включаю в себя три механизма борьбы с карточным мошенничеством:

- контроль размера операций, выполненных по карте
- случайный выбор транзакции для её онлайновой авторизации эмитентом
- проверка офлайновой активности использования карты

По мере выполнения процедур аутентификации карты, проверки ограничений на обработку транзакции, верификации владельца карты, управления рисками терминал формирует поле с набором признаков, информирующих о результатах проверок, выполняемых терминалом в процессе обработки транзакции, которое называется Terminal Verification Results (TVR). После завершения всех процедур терминал выполняет анализ всех ситуаций, зафиксированных в TVR. Целью такого анализа является выработка терминалом рекомендательного решения о том, каким образом с точки зрения терминала должна быть продолжена обработка транзакции. Возможны три решения терминала:

- транзакция должна быть одобрена в офлайновом режиме
- транзакция должна быть направлена для авторизации эмитенту
- транзакция должна быть отклонена в офлайновом режиме

Когда говорят о решении «с точки зрения терминала», то имеют в виду, что в действительности решение формируется на основе правил, определенных обслуживающим банком (платежной системой) и эмитентом карты. Для этого в спецификациях EMV определены два множества объектов данных, которые называются Issuer Action Codes (IAC) и Terminal Action Codes (TAC). В свою очередь, каждое из этих множеств состоит из трех объектов с суффиксами Denial, Online и Default. Таким образом, используются следующие объекты.

IAC-Denial	TAC-Denial
IAC-Online	TAC-Online
IAC-Default	TAC-Default

Каждый из этих объектов имеет тот же формат, что и TVR. IAC и TAC не являются обязательными с точки зрения спецификаций EMV. Но ведущие платежные системы требуют их присутствия на карте и в терминале.

TAC загружаются в терминал обслуживающим банком (например, VISA и MasterCard определяют обязательные TAC для обслуживающих банков). Эти

объекты зависят от типа терминала и от карточного продукта. IAC определяются эмитентом карты и заносятся на карту во время её персонализации. Эти объекты определяют политику эмитента в области обеспечения безопасности своих операций. Назначение объектов поясняется в нижеприведенной таблице (в этой таблице используется синоним словосочетания «обслуживающий банк» –эквайер).

Объект	Источник	Назначение	По умолчанию
IAC-Denial	Эмитент	Условия, при которых транзакция должна быть отвергнута	Все нули
IAC-Online	Эмитент	Условия, при которых транзакция должна быть отправлена на авторизацию эмитенту	Все единицы
IAC-Default	Эмитент	Условия, при которых терминал, неспособный обслужить транзакцию в online, должен её отвергнуть	Все единицы
TAC- Denial	Эквайер	Условия, при которых транзакция должна быть отвергнута	Все нули
TAC- Online	Эквайер	Условия, при которых транзакция должна быть отправлена на авторизацию эмитенту	Все нули
TAC-Default	Эквайер	Условия, при которых терминал, неспособный обслужить транзакцию в online, должен её отвергнуть	Все нули

Хотя по умолчанию все биты объектов TAC-Online и TAC-Default равны 0, в то же время настоятельно рекомендуется, чтобы обслуживающий банк определил по крайней мере следующие признаки:

- не выполнена офлайновая аутентификация данных карты
- офлайновая аутентификация данных карты провалилась

Терминал формирует свое решение о способе обработки транзакции, сравнивая биты, установленные в TVR, IAC и TAC, следующим образом.

1. Если в TVR есть единичные биты и соответствующие им биты в IAC-Denial и TAC-Denial также установлены в 1[1], то транзакция должна быть отвергнута без попытки выполнения онлайновой авторизации. В противном случае, терминал переходит к шагу 2, если терминал способен выполнить транзакцию в online, или к шагу 3, когда терминал работает только в offline.

2. Когда в TVR есть единичные биты и соответствующие им биты в IAC-Online и TAC-Online также установлены в 1, терминал считает, что транзакцию нужно отправить на авторизацию эмитенту. В противном случае, терминал предлагает карте одобрить транзакцию в офлайновом режиме.

3. Если в TVR есть единичные биты и соответствующие им биты в IAC-Default и TAC-Default также установлены в 1, то терминал считает, что транзакция должна быть отклонена. В противном случае, терминал предлагает карте одобрить транзакцию в офлайновом режиме.

Принятое решение о способе обработки транзакции терминал сообщает карте в первой команде GENERATE AC. В первой команде GENERATE AC терминал может запросить от карты формирование одной из следующих криптограмм.

1. Криптограммы AAC (Application Authentication Cryptogram), если транзакция должна быть отвергнута без попытки выполнения онлайновой авторизации.

2. Криптограммы ARQC (Authorization Request Cryptogram), когда терминал принял решение отправить транзакцию на авторизацию эмитенту.

3. Криптограммы TC (Transaction Certificate), если терминал предлагает карте одобрить транзакцию в офлайновом режиме

Карта получает от терминала в команде GENERATE AC не только тип транзакции, запрашиваемой терминалом, но и данные, которые необходимы карте для выполнения собственных процедур управления рисками. Терминал узнает о том, какие данные требуются карте для выполнения процедур управления рисками и вычисления криптограммы через списки объектов данных, которые описаны в следующем разделе.

[1] Т.е. результат операции побитового логического умножения TVR и результата побитового логического сложения IAC и TAC не равен 0.

Списки объектов данных

Для выполнения ряда команд карте требуются данные о транзакции и терминале. В реализации платежного приложения PayCORN такие данные требуются для первой и второй команд GENERATE AC, а также для команды GET PROCESSING OPTIONS в бесконтактном режиме. Список объектов данных, необходимых для выполнения команды, в общем случае называется Data Object List (DOL). В применении для конкретных команд списки называются:

- Card Risk Management Data Object List 1 (CDOL1) – список объектов данных для первой команды GENERATE AC
- Card Risk Management Data Object List 2 (CDOL2) – список объектов данных для второй команды GENERATE AC
- Processing Options Data Object List (PDOL) – список объектов данных для команды GET PROCESSING OPTIONS в бесконтактном режиме

Любой список объектов данных представляет собой перечень тэгов и длин объектов данных, которые должны быть переданы карте. Все списки записываются на карту в процессе её персонализации и считываются терминалом по команде READ RECORD. Терминал в соответствующих командах передает только значения объектов, перечисленных в списках.

Управление рисками карты

Важная роль в процессе обработки транзакции отводится карте, которой эмитентом делегируются функции, связанные с принятием решения о способе завершения транзакции. Карта, как и терминал, выполняет собственные процедуры управления рисками (Card Risk Management – CRM). На основе выполненных проверок карта проводит анализ полученных результатов и выносит решение (точнее, решение эмитента) о способе завершения транзакции.

По аналогии с терминалом результаты своих проверок карта записывает в элемент данных, который называется Card Verification Results (CVR). Этот элемент данных используется только картой (и эмитентом[1]) для принятия решений по результатам обработки транзакций. Важное отличие CVR от TVR состоит в том, что в CVR хранится часть истории, связанной с использованием карты, и информация о текущем состоянии, которая может потребоваться эмитенту.

Выполняемые картой процедуры управления рисками можно разделить по следующим типам:

- определение статуса проверки PIN-кода в offline
- анализ результатов выполнения предыдущей транзакции
- проверка офлайновых лимитов, ограничивающих количество последовательно выполненных картой офлайновых транзакций
- проверка офлайновых лимитов, ограничивающих объем средств, потраченных в последовательно выполненных картой офлайновых транзакциях
- проверка статуса и результата выполнения команд скрипт-процессинга
- проверка специальных условий обслуживания карты (признаков, которые сигнализируют о необходимости выполнения текущей транзакции в online, определение факта того, что карта ещё ни разу не была авторизована эмитентом и т. п.)

По мере выполнения процедур управления рисками карта формирует CVR. После завершения всех процедур карта выполняет анализ всех ситуаций, зафиксированных в CVR. Целью такого анализа является выработка картой решения об обработке транзакции. Решение формируется на основе правил, определенных эмитентом карты. Для этого в платежном приложении определено множество объектов данных, которые называются Card Issuer

[1] CVR передается эмитенту в качестве компонента объекта Issuer Application Data.

Action Codes (CIAC) и состоит из трех объектов с суффиксами Denial, Online и Default, т. е. CIAC-Denial, CIAC-Online и CIAC-Default.[1]

Каждый из объектов CIAC имеет формат, похожий на формат CVR. Как обсуждалось ранее, в CVR хранится некоторая информация о текущем состоянии, которая не используется в выработке решений картой. В связи с этим в CIAC определены только биты CVR, информирующие о результатах выполнения предыдущей транзакции и исключительных ситуациях, зафиксированных для текущей транзакции.

Выработка картой решения об обработке транзакции включает следующие шаги.

1. Если терминал запрашивает у карты криптограмму AAC, то карта не анализирует CVR и генерирует запрашиваемую криптограмму.

2. Если терминал запрашивает у карты криптограмму ARQC, то действия карты зависят от типа терминала. Когда терминал способен выполнить транзакцию в online, терминал соглашается с решением терминала и формирует криптограмму ARQC. В случае офлайнового терминала транзакция отвергается и формируется криптограмма AAC.

3. Когда терминал запрашивает у карты криптограмму TC, и в CVR установлены биты, которым найдено соответствие в CIAC-Denial[2], транзакция отвергается, карта формирует криптограмму AAC и возвращает её терминалу. В противном случае карта переходит либо к шагу 4, если терминал способен выполнить транзакцию в online, либо к шагу 5 – в противном случае.

4. Карта проверяет, не соответствуют ли ненулевым битам в CVR биты, установленные в CIAC-Online. Если такое соответствие найдено, то карта считает, что транзакция нужно отправить на авторизацию эмитенту, и формирует криптограмму ARQC. В противном случае карта считает, что транзакцию нужно одобрить в офлайновом режиме, и формирует криптограмму TC.

5. Если в CVR установлены биты, которым найдено соответствие в CIAC-Default, транзакция отвергается и карта формирует криптограмму AAC. В противном случае карта предлагает одобрить транзакцию в офлайновом режиме, и формирует криптограмму TC.

Рассмотрим, каким образом эмитент может управлять решением карты об обработке транзакции с помощью признаков, установленных в CIAC, на

[1] При обработке транзакции в бесконтактном режиме используется еще один объект – CIAC-CVM, который будет описан позже.

[2] Т. е. результат операции побитового логического умножения CVR и CIAC не равен 0.

примере офлайновых счетчиков. В платежном приложении существует два вида счетчиков:

- однобайтный счетчик количества последовательных операций, выполненных в offline (Consecutive Offline Transaction Number – COTN)
- сумма всех последовательных операций, выполненных в offline, которая представляется в формате n12 в валюте платежного приложения с неявной десятичной точкой (Consecutive Offline Transaction Amount – COTA)

Каждому из счетчиков соответствует два лимита, определяемых эмитентом: нижний лимит и верхний лимит. Карта устанавливает в CVR признаки превышения заданных лимитов. Любой из счетчиков может использоваться для ограничения объема средств, потраченных в последовательно выполненных картой офлайновых транзакциях. Например, эмитент хочет использовать для этой цели количество последовательных операций, выполненных в offline, как это показано на рис. 4.

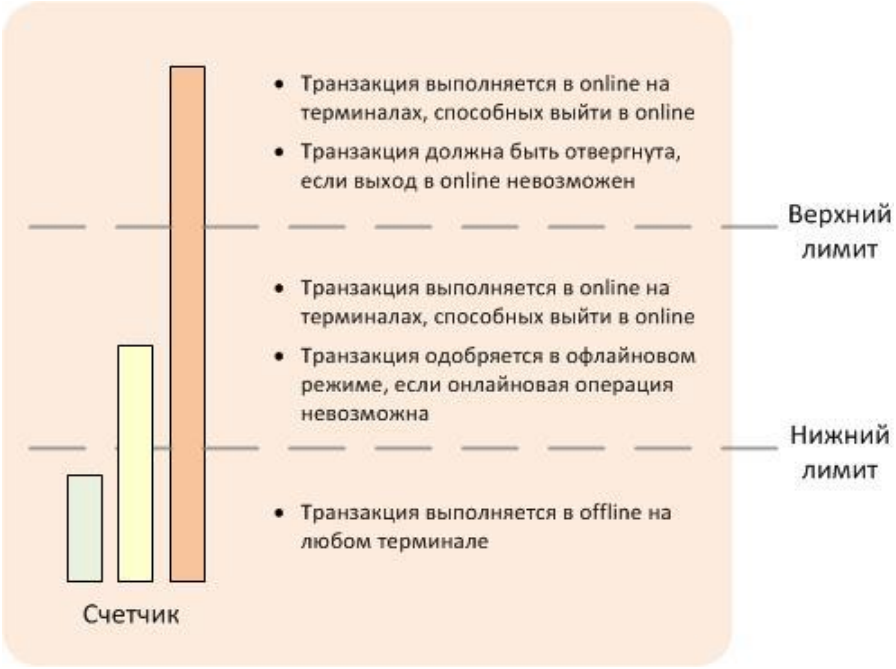

Рис. 4. Типичное использование офлайнового счетчика.

Логику ограничения последовательно выполненных картой офлайновых транзакций можно описать следующим образом:

- если счетчик меньше или равен нижнему лимиту, то транзакция должна быть выполнена в offline
- когда счетчик больше нижнего лимита, но не выше верхнего лимита, транзакция должна аутентифицирована эмитентом на терминалах, которые способны выйти в online, или должна быть одобрена в офлайновом режиме, если онлайновая операция невозможна[1]
- если счетчик превышает верхний лимит, то транзакция должна быть выполнена в online на терминалах, которые способны выйти в online, или должна быть отвергнута, если онлайновая операция невозможна

Чтобы карта могла поддержать такую логику ограничения последовательно выполненных офлайновых транзакций, эмитент должен установить в CIAC-Online биты «Превышен нижний предел офлайновых транзакций» и «Превышен верхний предел офлайновых транзакций», а в CIAC-Default –бит «Превышен верхний предел офлайновых транзакций» (см. раздел «Card Issuer Action Code» в главе «Кодирование элементов данных»»).

Тогда, если терминал запрашивает криптограмму TC, карта увеличивает значение счетчика на 1, сверяет его лимитами и устанавливает в CVR признак «Превышен нижний предел офлайновых транзакций», если счетчик больше нижнего предела, и признак «Превышен верхний предел офлайновых транзакций», если счетчик больше верхнего лимита. Если в CVR не установлен ни один из признаков превышения лимита, то карта сформирует криптограмму TC независимо от типа терминала. В случае установки в CVR признака превышения нижнего лимита (но не верхнего) для онлайнового терминала будет найдено соответствие признаков в CVR и CIAC-Online и карта вернет криптограмму ARQC, а для офлайнового терминала – криптограмму TC, поскольку в CIAC-Default признак «Превышен нижний предел офлайновых транзакций» не установлен. Наконец, если в CVR установлены оба признака превышения лимитов, то карта сформирует криптограмму ARQC для онлайнового терминала и криптограмму AAC для офлайнового терминала, поскольку в CIAC-Default установлен признак «Превышен верхний предел офлайновых транзакций».

[1] Потому что терминал работает только в offline, или потому что терминал не может в данный момент выполнить транзакцию в online (unable to go online).

Аутентификация карты

Процедура офлайновой аутентификации карты терминалом является одним из ключевых элементов безопасности операций, выполняемых с помощью карты. Платежное приложение PayCORN поддерживает только один метод офлайновой аутентификации – CDA (Combined Dynamic Data Authentication/ Application Cryptogram Generation).

С помощью метода CDA не только осуществляется проверка целостности важных статических данных приложения и подлинности карты, но и обеспечивается целостность критичных данных, циркулирующих между картой и терминалом. Для реализации метода CDA списки CDOL1 и CDOL2 должны обязательно содержать объект данных Unpredictable Number (тэг 9F37), значение которого представляет собой случайное число, сформированное терминалом. В общих словах суть метода CDA состоит в том, что аутентификация карты производится в рамках процедуры обработки команды GENERATE AC, а не до выполнения этой команды, как в случае аутентификации карты по другим методам (SDA и DDA). Поскольку именно при выполнении команды GENERATE AC между картой и терминалом происходит обмен наиболее критичными данными транзакции, при использовании метода CDA можно одновременно выполнить динамическую аутентификацию карты и обеспечить целостность критичных данных.

При выполнении команды GENERATE AC, карта формирует пакет данных, который содержит критичные данные транзакции (тип криптограммы и криптограмму приложения), а также Transaction Data Hash Code (хэш определенных параметров транзакции). Этот пакет подписывается (зашифровывается) на секретном ключе карты и предоставляется терминалу в виде сертификата Signed Dynamic Application Data.[1]

Терминал производит проверку цифровой подписи CDA[2], сравнивает тип криптограммы, полученный из подписанных данных и из поля данных ответа на команду GENERATE AC, убеждается в равенстве значений Transaction Data Hash Code, полученных из подписанных данных и вычисленных терминалом самостоятельно. Карта считается успешно аутентифицированной, когда все перечисленные проверки завершились успешно. Более подробно алгоритм получения подписи CDA и её проверки описан в главе «Криптографические алгоритмы».

[1] Подпись CDA формируется только в том случае, когда карта возвращает криптограмму TC или ARQC. Для криптограммы AAC подпись CDA никогда не формируется.

[2] Сертификат Signed Dynamic Application Data расшифровывается на публичном ключе карты, который терминал получает в результате процедур «Восстановление публичного ключа эмитента» и «Восстановление публичного ключа карты». Эти процедуры также являются частью офлайновой аутентификации карты. Они описаны в главе «Криптографические алгоритмы».

Онлайновая обработка транзакции

Если карта в ответе на первую команду GENERATE AC указывает, что транзакцию нужно отправить на авторизацию эмитенту, терминал в соответствии с используемым протоколом работы терминала с хостом передает сообщение, которое содержит информацию, относящуюся к карте и результатам обработки транзакции. На основе этих данных хост обслуживающего банка формирует авторизационный запрос (сообщение x100 стандарта ISO 8583), содержащий такие данные, как:

- криптограмма ARQC
- информация о карте (PAN, срок действия карты и т. п.)
- информация о терминале (идентификатор, тип и т. п.)
- последовательный номер транзакции ATC
- элемент Issuer Authentication Data, сформированный картой

При получении авторизационного запроса эмитент проверяет, является ли карта, по которой проводится операция, подлинной. Для этого эмитент вычисляет сессионный ключ генерации криптограммы, после чего использует его и данные транзакции для получения криптограммы. Полученная криптограмма сравнивается с предоставленным значением ARQC. Если значения совпадают, аутентификация карты считается завершенной успешно, а сама карта – подлинной.

Далее эмитент проводит стандартные проверки (проверка активности карты, истории использования карты, отсутствия карты в списке заблокированных карт, наличие средств на карт-счете, и т. п.).

Данные, получаемые эмитентом в авторизационном запросе, позволяют ему принять правильное решение по способу завершения текущей операции, повлиять на выполнение следующей транзакции и произвести коррекцию данных карты с помощью команд скрипт-процессинга и элемента данных Card Status Update (CSU).

После принятия решения о результате операции эмитент генерирует элемент данных Issuer Authentication Data. Этот элемент содержит криптограмму эмитента ARPC, которая используются картой для аутентификации эмитента, и Card Status Update (CSU), который указывает, одобрена или отклонена транзакция эмитентом, а также определяет действия, которые с точки зрения эмитента должны быть выполнены картой. В авторизационный ответ обслуживающему банку (сообщение x100 стандарта ISO 8583) помещается элемент данных Issuer Authentication Data, код авторизации эмитента а также команды скрипт-процессинга, если они должны быть переданы карте.

Если терминал не получил авторизационный ответ или получил его слишком поздно, терминал продолжает обрабатывать транзакцию, считая, что запрос невозможно передать эмитенту (эта ситуация обычно называется «unable to go online»). В любом случае, терминал должен получить криптограмму приложения от карты с помощью второй команды GENERATE AC. Однако, перед выполнением этой команды терминал должен передать карте команды критичного скрипт-процессинга, если они присутствуют в авторизационном ответе.

Во второй команде GENERATE AC терминал передает карте данные, полученные от эмитента (код авторизации эмитента, элемент Issuer Authentication Data)[1] и уточненные данные своих проверок (TVR). Во второй команде GENERATE AC терминал может запросить от карты формирование одной из следующих криптограмм.

1. Криптограммы AAC, если транзакция должна быть отклонена.

2. Криптограммы TC, если терминал считает, что транзакцию следует одобрить.

Процесс принятия картой решения после получения второй команды GENERATE AC включает следующие шаги.

1. Если терминал запрашивает криптограмму AAC, то карта генерирует запрашиваемую криптограмму.

2. Когда терминал информирует о ситуации «unable to go online», карта проверяет, не соответствуют ли ненулевым битам в CVR[2] биты, установленные в CIAC-Default. Если такое соответствие найдено, то транзакция отвергается и карта формирует криптограмму AAC. В противном случае карта может одобрить транзакцию после проведения дополнительных проверок.

3. Карта аутентифицирует эмитента (проверяет значение криптограммы ARPC в элементе Issuer Authentication Data). Если аутентификация эмитента не выполнена, то обычно транзакция отвергается (формируется криптограмма AAC).[3]

4. Если аутентификация эмитента выполнена успешно, дальнейшие действия карты определяются выбором эмитента, который определен в

[1] Если онлайновую обработка невозможна (ситуация «unable to go online»), то терминал сообщает об этом через код авторизации эмитента (элемент Issuer Authentication Data при этом обнулен).

[2] Следует учесть, что во время второй команды GENERATE AC информация в CVR может быть обновлена, поскольку CVR всегда отображает текущее состояние транзакции. Таким образом, можно утверждать, что карта, как и терминал, уточняет данные своих проверок.

[3] Эмитент может определить другую политику с помощью элемента данных Issuer Policy (IP), который описан в главе «Кодирование элементов данных.

Card Status Update (CSU). Карта формирует криптограмму TC или AAC в зависимости от решения эмитента об обработке транзакции, и выполняет действия, которые определил эмитент в CSU (например, установку счетчика оставшихся предъявлений PIN-кода, сброс офлайновых счётчиков и т. п.). Кроме этого, карта сбрасывает признаки особых ситуаций, которые были зафиксированы при выполнении предыдущей или текущей транзакции.

Решение карты, принятое при выполнении второй команды GENERATE AC, является окончательным.

После завершения второй команды GENERATE AC терминал должен передать карте команды некритичного скрипт-процессинга, если они присутствуют в авторизационном ответе.

Транзакция в бесконтактном режиме

Интерес к бесконтактным картам можно объяснить несколькими их техническими преимуществами:

- удобство использования карты (карту не нужно передавать кассиру, правильно ориентировать и вставлять в прорезь ридера, даже можно не вытаскивать из бумажника)
- более высокая скорость выполнения транзакции
- более высокая надежность использования карт и терминалов – из-за отсутствия механического контакта карты и терминала обеспечивается более низкий уровень их физического износа
- лучшая защищенность бесконтактных терминалов от случаев вандализма

Компания EMVCo уже давно прорабатывает вопрос о создании единого бесконтактного приложения EMV Contactless Application – аналога приложения Common Payment Application (CPA) для контактных карт. Однако компания EMVCo не торопится с разработкой стандарта для EMV Contactless Application. Причина, озвучиваемая EMVCo, – недостаток опыта использования бесконтактных карт. Каждый год проводится внутреннее обсуждение вопроса о целесообразности разработки стандарта, но положительного решения до сих пор нет. Появление стандарта на единое бесконтактное приложение в будущем очевидно. Поэтому EMVCo, предваряя появление такого стандарта, разработала спецификацию EMV Contactless Specifications for Payment Systems – Entry Point Specification (в дальнейшем будем для краткости называть его Entry Point Specification), решающую ряд вопросов.

‘Этот стандарт определяет общую схему обработки транзакции по бесконтактной карте на стороне терминала (Entry Point) и достаточно подробно останавливается на описании процедур, предшествующих выбору приложения. На рис. 5 показана общая схема обработки транзакции в бесконтактном режиме. Не останавливаясь на деталях обработки отметим, что после предварительной обработки и активации протокола терминал начинает процедуру выбора бесконтактного приложения. Если терминал поддерживает единственное бесконтактное приложение, то он сразу выбирает это приложение с помощью команды SELECT. Иначе, терминал выбирает приложение PPSE (Proximity Payment System Environment), которое имеет идентификатор 2PAY.SYS.DDF01, и получает в ответ объект FCI, который содержит информацию о бесконтактных приложениях на карте. Далее терминал определяет, какие из поддерживаемых им приложений находятся на карте, локализует приложение с наивысшим приоритетом и второй командой SELECT выбирает его.

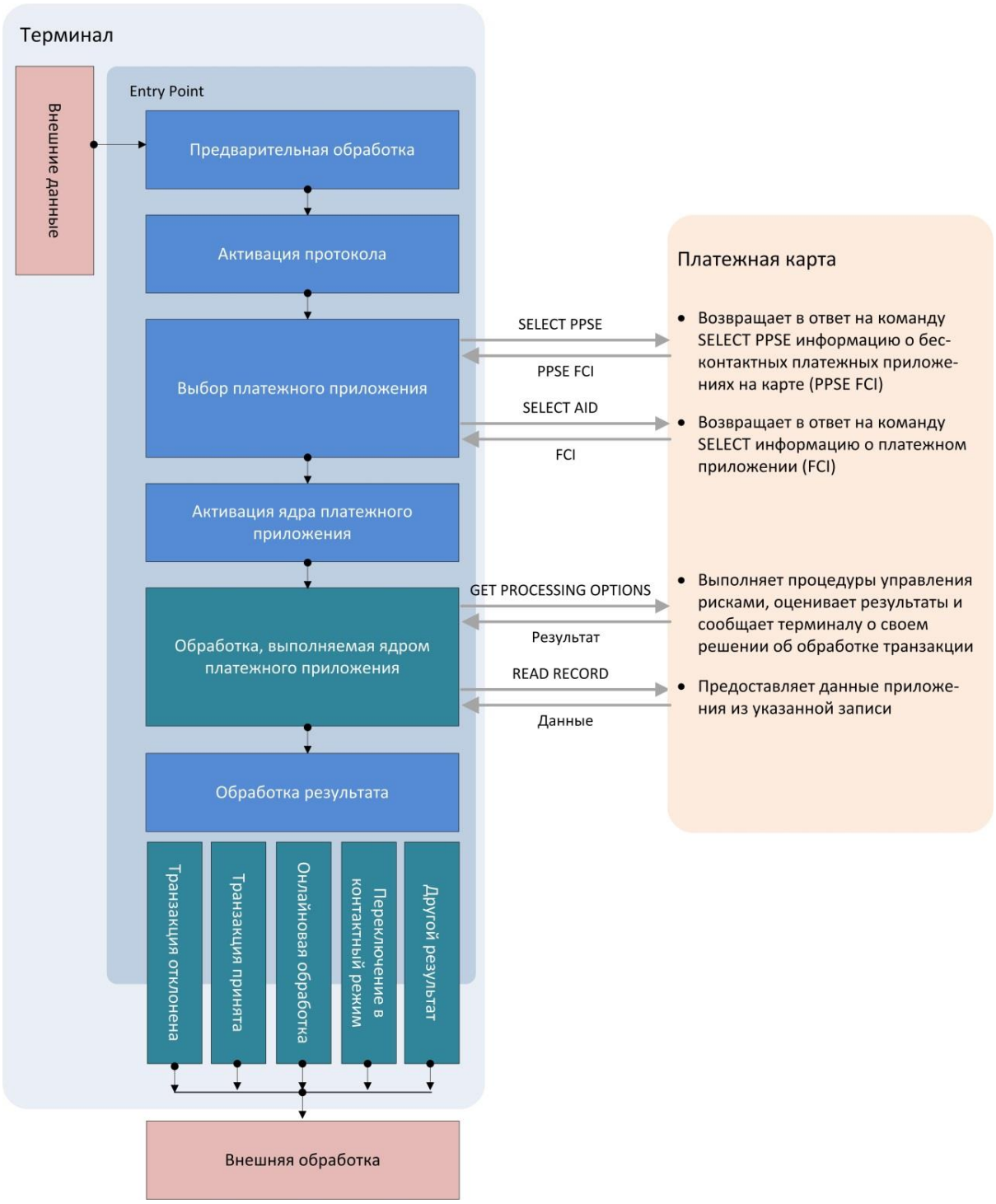

Рис. 5. Обработка транзакции в бесконтактном режиме.

После выбора приложения происходит активация ядра, соответствующего приложению, которому Entry Point терминала передает управление. Ядро завершает обработку формированием результата для Entry Point. Возможные результаты обработки ядра – отклонение транзакции или одобрение в

офлайновом режиме, направление транзакции для авторизации эмитенту, требование переключения в контактный режим и т. п.

На этапе проектирования платежного аплета PayCORN было поставлено требование, которое определило особенности работы платежного приложения в бесконтактном режиме. Это требование можно сформулировать следующим образом: «В бесконтактном режиме транзакция не может быть одобрена в офлайновом режиме». А требование появилось из-за того, что в офлайновом режиме невозможна надежная верификация владельца карты.[1]

Процедуры управления рисками, выполняемые терминалом в бесконтактном режиме, достаточно просты. Во-первых, терминал должен сравнить сумму транзакции с пороговой суммой Contactless Transaction Limit. Если сумма транзакции превышает это значение, то бесконтактная транзакция не выполняется. Во-вторых, терминал сравнивает сумму транзакции с пороговой суммой CVM Required Limit. Когда сумма транзакции превышает это пороговое значение, терминал считает, что должна быть выполнена верификация владельца карты.

Далее терминал подает на карту команду GET PROCESSING OPTIONS, с которой передаются данные, необходимые карте для принятия решения о способе обработки транзакции. Эти данные определяются списком PDOL[2] и включают также Terminal Transaction Processing Information (TTPI) – элемент, который описывает возможности терминала и результат предварительной обработки терминалом транзакции в бесконтактном режиме. TTPI информирует карту о том, поддерживается ли контактный режим обработки транзакции, есть ли у терминала возможность выхода в online, поддерживает ли терминал ввод PIN-кода в online. Кроме того, через TTPI терминал информирует карту, считает ли он, что требуется верификация владельца карты.

Карта выполняет процедуры управления рисками и формирует CVR. При установке отдельных битов CVR учитываются признаки особых ситуаций, возникших при обработке предыдущей транзакции в контактном режиме (признаки особых ситуаций в бесконтактном режиме не устанавливаются). Поскольку бесконтактная транзакция может быть выполнена только в online,

[1] Действительно, верификация владельца карты методом офлайновой проверки PIN-кода нежелательна по многим причинам. Недаром, общая черта спецификаций бесконтактных приложений MasterCard и VISA – отказ от предъявления PIN-кода в offline. Эти бесконтактные приложения используют два метода верификации – предъявление PIN-кода в online (при направлении транзакции для авторизации эмитенту) и подпись (в офлайновом режиме). При проектировании платежного аплета PayCORN подпись была признана ненадежным методом верификации владельца карты. Поэтому и пришлось полностью отказаться от обработки бесконтактной транзакции в офлайновом режиме.

[2] Список PDOL предоставляется терминалу в данных FCI (File Control Information), возвращенных в ответ на команду SELECT для выбора платежного приложения.

активность расходования средств в offline не проверяется и соответствующие признаки в CVR никогда не устанавливаются. Если в TTPI установлен признак того, что требуется верификация владельца карты, то в CVR устанавливается признак «Требуется ввод PIN-кода в online».

После завершения всех процедур карта принимает решение относительно завершения транзакции в соответствии с массивом CIAC. В бесконтактном режиме используются только CIAC-Denial и CIAC-CVM. Если в CVR не установлен признак «Требуется ввод PIN-кода в online», то проверяется, есть ли соответствие признаков в CVR и CIAC-CVM. Если такие соответствия найдены, то в CVR устанавливается признак «Требуется ввод PIN-кода в online». В завершение проверяется, не произошли ли какие-то события, из-за которых транзакция должна быть отклонена. Сравниваются признаки в CVR и CIAC-Denial. Если найдены соответствия признаков, то принимается окончательное решение об отклонении транзакции. Иначе, принимается предварительное решение о выполнении транзакции в online.

В результате предварительной обработки может быть принято одно из следующих решений:

- требуется выполнение транзакции в online (предварительное)
- требуется отклонение транзакции (окончательное)

Для первого решения выполняются дополнительные проверки, которые позволяют карте изменить свое решение в зависимости от возможностей терминала, определенных в TTPI. Если терминал работает только в offline, или в CVR установлен признак «Требуется ввод PIN-кода в online», а терминал не поддерживает ввод PIN-кода в online, принимается решение о переключении в контактный режим. После этого проверяется, поддерживает ли терминал контактный режим. Если нет, то решение о переключении в контактный режим меняется на решение «требуется отклонение транзакции». Таким образом, после выполнения дополнительных проверок решение всегда согласовано с возможностями терминала и является окончательным.

В зависимости от окончательного решения карта возвращает разные данные. Единственный элемент, который всегда входит в состав возвращаемых данных – это Contactless Cryptogram Information Data (CCID), значение которого информирует терминал об окончательном выборе карты относительно завершения транзакции или о необходимости переключения на контактный интерфейс.

Если принято решение об отклонении транзакции или выполнении транзакции в online, то карта возвращает элемент данных Issuer Application Data, криптограмму (AAC или ARQC), а также другие данные, которые могут потребоваться терминалу для продолжения обработки. Среди этих данных присутствует элемент Application File Locator (AFL), содержащий ссылки на данные, которые терминал должен прочитать, чтобы успешно выполнить

транзакцию. После чтения данных, определяемых AFL, с помощью команды READ RECORD обработка ядра завершается формированием результата. Начиная с этого момента времени, карта уже не нужна для дальнейшей обработки и может быть удалена из зоны считывания терминала.

Далее действия терминала зависят от результата обработки ядра. Возможны следующие варианты.

1. Требуется переключение в контактный режим обработки. Терминал предпринимает попытку выполнить транзакцию в контактном режиме.

2. Транзакция отклонена. Терминал завершает обработку транзакции.

3. Должна быть выполнена онлайновая обработка. Терминал запрашивает ввод PIN-кода, если это требуется, формирует авторизационный запрос для эмитента (см. раздел «Онлайновая обработка транзакции») и пересылает этот запрос хосту. Основное отличие онлайновой обработки транзакции в бесконтактном режиме заключается в том, что ответ эмитента проверяет не карта (она уже удалена), а терминал. Конечно, терминал не может выполнить аутентификацию эмитента. Его решение об обработке транзакции должно соответствовать выбору эмитента, который определен в Card Status Update (CSU).

Таким образом, бесконтактный режим по сравнению с контактным режимом имеет ряд существенных ограничений. Вот основные из них.

1. Бесконтактная транзакция может быть выполнена только на терминале с возможностью онлайновой обработки.

2. Бесконтактная транзакция не может быть одобрена в офлайновом режиме.

3. Ни один из элементов платежной системы не выполняет аутентификацию эмитента при выполнении бесконтактной транзакции, что понижает уровень безопасности платежной системы.

4. Не могут быть выполнены команды скрипт-процессинга эмитента.

5. Отсутствует возможность выполнения действий, определяемых эмитентом в CSU (например, установки счетчика оставшихся предъявлений PIN-кола, сброса офлайновых счётчиков и т. п.).

ГЛАВА 2

Персонализация

Персонализация карты – это важная часть подготовки карты к работе, которую выполняет эмитент. В процессе персонализации платежный аплет, загруженный на карту, настраивается таким образом, чтобы он мог выполнять функции элемента платежной системы в зависимости от рисков, связанных с владельцем персонализируемой карты. Чтобы лучше понять процесс персонализации, нужно ознакомиться с некоторыми общими положениями из индустрии производства карт. Жизненный цикл карты принято делить на пять основных фаз:

- фаза производства микросхемы
- фаза предперсонализации карты и загрузки на неё платежного аплета
- фаза персонализации карты
- фаза использования карты
- фаза блокировки карты

Деление на фазы позволяет контролировать безопасность карты на разных этапах её существования и обеспечивает распределение ответственности между всеми участниками процессов производства, персонализации и использования карт. Поскольку в этой главе речь идет только о третьей фазе жизненного цикла, подробное описание других фаз можно опустить, но следует обратить внимание на несколько основных положений, без которых дальнейшее изложение не будет понятным.

На первых двух фазах жизненного цикла на карту загружаются два элемента, уникальные для каждой карты:

- серийный номер карты
- секретный ключ, который используется операционной системой карты для контроля доступа к карте

Хотя секретный ключ и уникален для каждой карты, но он может быть получен путем диверсификации мастер-ключа всего лота (партии) карт, поставленных производителем, на уникальном серийном номере карты.[1]

С помощью секретного ключа операционная система карты устанавливает защищенное соединение с внешней для карты программой[2]. Для этого сначала выполняется взаимная аутентификация карты и внешней программы, а затем организуется защищенное соединение для передачи от внешней программы на карту данных персонализации.

Процесс подготовки данных персонализации выполняется в бэк-офисной системе эмитента. Этот процесс может использовать данные, хранящиеся на разных хостах банка. Его цель – подготовить данные, которые должны быть загружены на карту. Часть подготавливаемых данных может быть общей для всего набора эмитируемых карт. Некоторые данные меняются от карты к карте. Часть данных может передаваться карте в открытом виде. В то же время имеются данные (ключи, PIN-коды), которые во время всего процесса персонализации должны находиться в зашифрованном виде.

Подробно весь процесс персонализации разбирается в документе EMV Card Personalization Specification. Version 1.1. July 2007. Поэтому остановимся только на общих положениях и особенностях персонализации платежного аплета PayCORN.

[1] На самом деле для контроля доступа к карте используется не один ключ, а три, но все они получаются путем диверсификации мастер-ключа. Кроме того, следует сказать, что ряд производителей карт использует другую схему дистрибуции ключей, когда с лотом карт поставляется не один ключ, а три, но для понимания принципов персонализации это не важно.

[2] Терминологически не совсем верно, поскольку за установку защищенного соединения отвечает специальный аплет карты, который называется Issuer Security Domain.

Общая схема персонализации

Самая общая схема персонализации платежного приложения приведена на рис. 6.

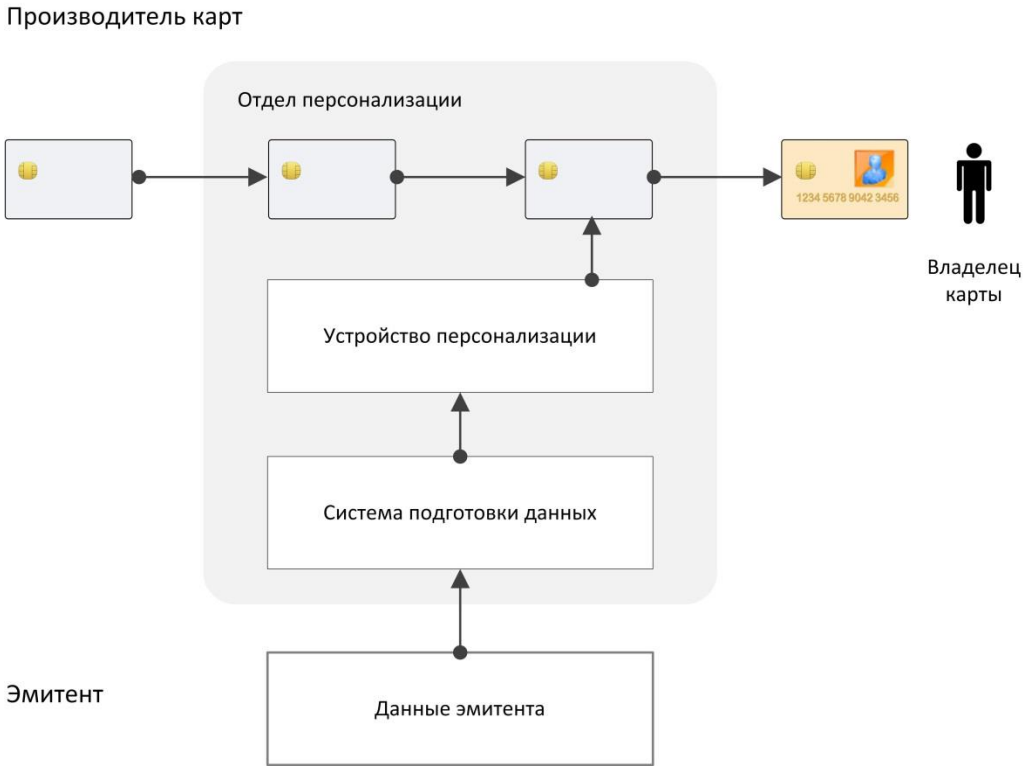

Рис. 6. Общая схема персонализации.

На этом рисунке присутствуют основные объекты, отвечающие за персонализацию платежного приложения.

- Эмитент. Юридическое лицо, которое от своего имени выпускает платежные карты для финансирования или развития своей деятельности.
- Производитель карт. Юридическое лицо, отвечающее за правильную персонализацию карт в соответствии с данными эмитента.
- Система подготовки данных. Система, которая выполняет проверки, подготовки и форматирования данных, которые пересылаются устройству персонализации.
- Устройство персонализации. Устройство, которое принимает данные от системы подготовки данных и посылает команды персонализации карте.

В процессе персонализации принимают участие несколько физических модулей устройства персонализации (модули эмбоссирования, кодирования магнитной полосы, занесения данных на карту), каждому из которых требуется

свой набор данных. Нас будет интересовать только один модуль – занесения данных на карту.

Чтобы персонализировать платежное приложение, устройству персонализации должны быть предоставлены данные эмитента. Эти данные представляются в виде объектов персонализации платежного приложения. С каждым объектом персонализации сопоставляется DGI (Data Group Identifier) – тэг, под которым объект персонализации известен платежному приложению.

Существуют такие данные персонализации (ключи, PIN-коды), которые во время всего процесса персонализации должны находиться в зашифрованном виде. Будем их в дальнейшем называть конфиденциальными данными. Любой обмен конфиденциальными данными между объектами системы персонализации должен быть зашифрован на определенных ключах. Обычно ключи, на которых осуществляется обмен конфиденциальными данными, отличаются для различных объектов системы персонализации. Например, на приведенном рисунке конфиденциальные данные должны быть зашифрованы на уникальных ключах в следующих случаях:

- при передаче данных эмитента системе подготовки данных
- при пересылке отформатированных данных из системы подготовки данных в устройство персонализации
- в командах персонализации, которые посылаются на персонализируемую карту

Вопросы безопасности очень важны и здесь будут кратко рассмотрены. Начнем с обсуждения безопасного обмена данными с картой.

Безопасный обмен данными с картой

Для персонализации платежного приложения используется канал безопасности, по которому данные передаются карте. Чтобы понять, как организован процесс персонализации, необходимо иметь общие понятия о механизме безопасности карты, называемом Secure Channel Protocol версии 02, или просто SCP02, подробное описание которого приведено в документе GlobalPlatform. Card Specification. Version 2.2. March 2006. Принципы безопасного обмена данными с картой показаны на рис. 7.

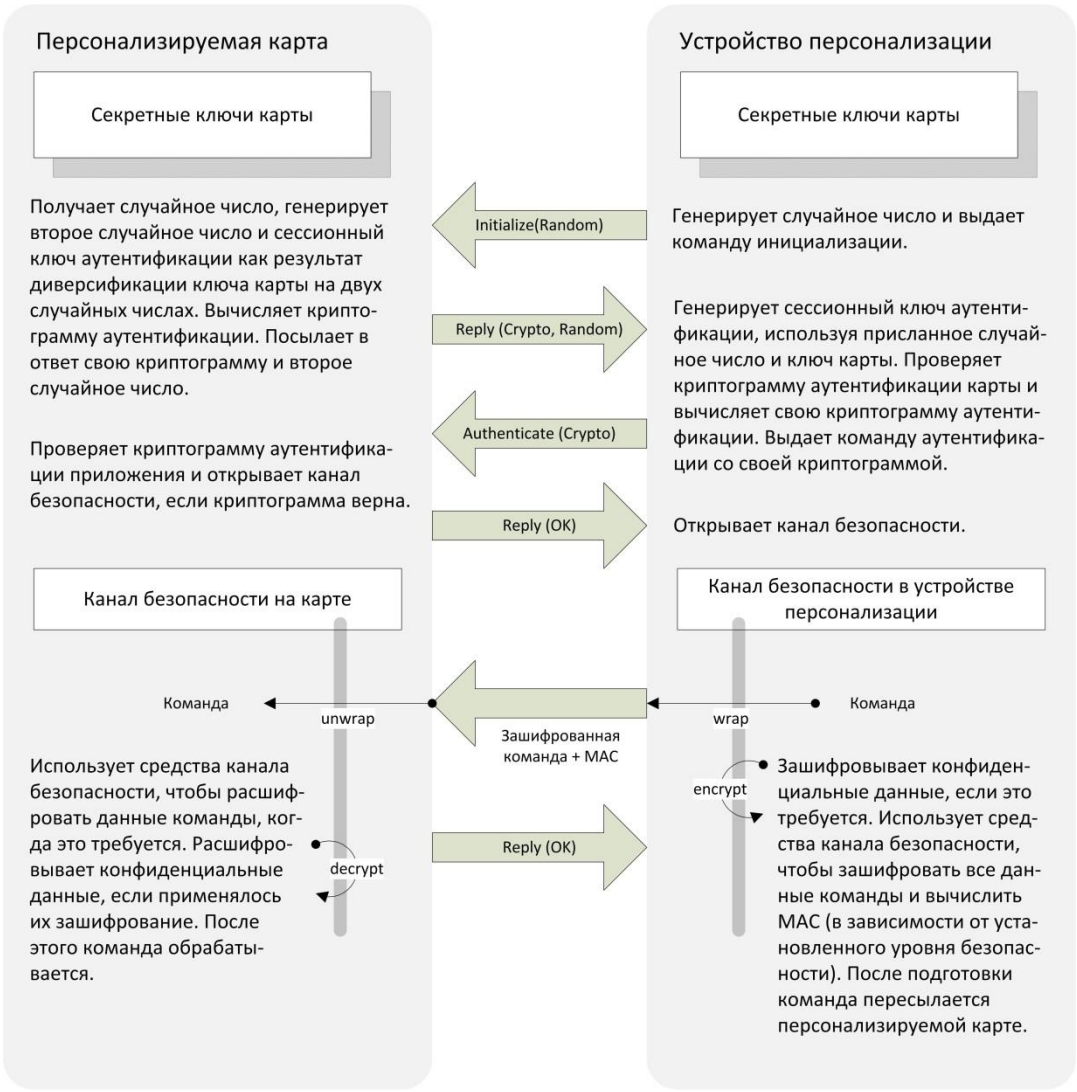

Рис. 7. Канал безопасности при работе с персонализируемой картой.

Чтобы организовать безопасный обмен с картой, устройству персонализации должны быть доступны секретные ключи карты.[1] Процесс создания канала безопасности начинается с того, что устройство персонализации выбирает на карте платежное приложение, и посылает ему команду инициализации, с которой передает случайное число ($Random_1$).

Платежное приложение, получив команду инициализации, направляет её домену безопасности. Домен безопасности – это специальный аплет карты, которому известны секретные ключи карты, и который имеет право на работу с ними. Получив команду инициализации, домен безопасности генерирует второе случайное число ($Random_2$) и получает сессионные ключи канала безопасности, диверсифицируя ключи карты на двух случайных числах. В канале безопасности используются следующие ключи:

- ключ аутентификации и зашифрования данных при обмене по каналу безопасности
- ключ целостности данных при обмене по каналу безопасности (ключ, на котором вычисляется MAC)
- ключ шифрования конфиденциальных данных

Чтобы объект, инициировавший открытие канала безопасности, мог аутентифицировать карту, домен безопасности вычисляет криптограмму аутентификации[2] на основе случайного числа $Random_1$, и посылает в ответе на команду инициализации криптограмму аутентификации и случайное число $Random_2$.

Устройство персонализации, получив ответ на команду инициализации, генерирует сессионные ключи канала безопасности по тем же принципам, что и домен безопасности, после чего проверяет криптограмму аутентификации, присланную в ответе. Если карта аутентифицирована (т. е. ключи карты и устройства персонализации совпадают), то устройство персонализации генерирует свою криптограмму аутентификации на основе случайного числа $Random_2$, после чего посылает на персонализируемую карту команду аутентификации с этой криптограммой, чтобы персонализируемая карта могла аутентифицировать объект, инициировавший открытие канала безопасности (такая взаимная аутентификация гарантирует, что обе стороны владеют одинаковым секретным ключом).

[1] Разумеется, секретные ключи хранятся в секретном модуле. Обычно, это Hardware Security Module (HSM). Устройству персонализации секретные ключи доступны через функции HSM.

[2] Для этого используется сессионный ключ аутентификации.

В команде аутентификации определяется также уровень безопасности, на котором должен функционировать канал безопасности.[1] Могут быть определены следующие уровни безопасности.

- Конфиденциальность и целостность. Данные команды зашифровываются на сессионном ключе аутентификации и для них вычисляется MAC (в терминологии GlobalPlatform – C-MAC) на сессионном ключе целостности данных.

- Целостность. Данные команды не зашифровываются, но для них вычисляется MAC на сессионном ключе целостности данных.

- Ни конфиденциальности, ни целостности. Данные команды не зашифровываются, и для них не вычисляется MAC.

Если домен безопасности на персонализируемой карте аутентифицировал объект, инициировавший открытие канала безопасности, то он открывает канал безопасности с указанным уровнем безопасности и посылает ответ с уведомлением, что канал открыт. Получив ответ, подтверждающий открытие канала безопасности, устройство персонализации открывает свой канал безопасности.

Далее канал безопасности может использоваться разными способами. Но для персонализируемого платежного приложения, которому ничего неизвестно о ключах карты, наиболее важны два способа.

Во-первых, платежное приложение может обратиться к домену безопасности, чтобы он применил установленный уровень безопасности к пришедшим данным и проверил данные на соответствие уровню безопасности (для этого используется метод unwrap). Например, если установлен уровень безопасности «Конфиденциальность и целостность», то будет проверен MAC и данные будут расшифрованы.

Во-вторых, платежное приложение может с помощью домена безопасности расшифровать конфиденциальные данные, зашифрованные устройством персонализации на сессионном ключе шифрования конфиденциальных данных (для этого используется метод decryptData).

Из всего вышеизложенного следует, что обмен данными персонализации с платежным приложением может быть абсолютно безопасным. Все данные персонализации (или только конфиденциальные данные) могут быть зашифрованы на секретном ключе персонализируемой карты, и подтверждены криптографическими контрольными суммами, полученными на другом ключе персонализируемой карты.

[1] Домен безопасности не обязан безоговорочно принимать указанный уровень безопасности. Он может сообщить, что уровень безопасности должен быть повышен.

Два метода персонализации

Как описано в документе EMV Card Personalization Specification. Version 1.1. July 2007, существует два подхода к персонализации. Поскольку они отличаются достаточно сильно, разберем оба эти метода.

Первый из методов называется непрямой метод персонализации (Indirect Method). Общая схема этого метода показана на рис. 8.

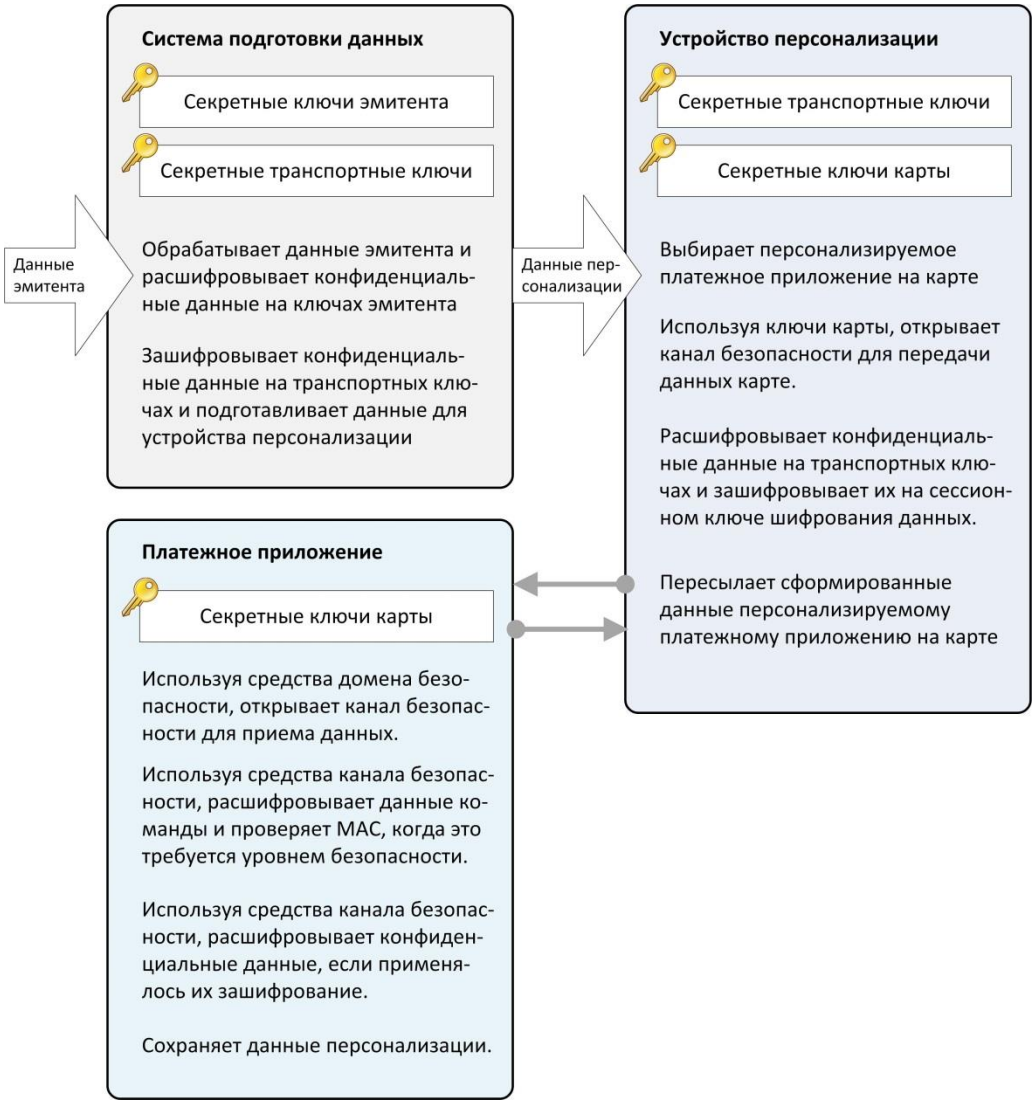

Рис. 8. Непрямой метод персонализации.

Непрямой метод персонализации был определен в самых ранних спецификациях персонализации EMV. Это стандартный подход, при котором данные эмитента поступают в систему подготовки данных, которая обрабатывает данные эмитента и расшифровывает конфиденциальные данные на ключах эмитента. После этого система подготовки данных

формирует данные для устройства персонализации. Но конфиденциальные данные не могут быть переданы устройству персонализации в открытом виде. Поэтому они зашифровываются на секретных транспортных ключах, которые известны как системе подготовки данных, так и устройству персонализации.

Устройство персонализации, получив данные от системы подготовки данных, выполняет персонализацию карты следующим образом:

- выбирает платежное приложение на карте и открывает канал безопасности для передачи данных карте
- расшифровывает конфиденциальные данные на транспортных ключах и зашифровывает их на сессионном ключе шифрования конфиденциальных данных
- пересылает сформированные данные персонализируемому платежному приложению

При таком подходе уровень безопасности, на котором функционирует канал безопасности не так важен. Конечно, рекомендуется использовать проверку целостности данных, но конфиденциальность является избыточной, поскольку конфиденциальные данные зашифрованы на сессионном ключе шифрования конфиденциальных данных.[1]

Второй метод называется прямой метод персонализации (Direct Method). Особенность этого метода (см. рис. 9) в том, что система подготовки данных и устройство персонализации объединены в единый объект (на рисунке он называется системой подготовки данных) и нет необходимости пересылать конфиденциальные данные устройству персонализации, зашифровывая их на транспортных ключах. Поэтому транспортные ключи при таком подходе отсутствуют. Но самое главное, система подготовки данных сама открывает канал безопасности и формирует команды для персонализации платежного приложения.[2] При этом на систему подготовки данных накладываются определенные обязательства:

- система подготовки данных должна уметь получать данные от карты и их обрабатывать
- системе подготовки данных должны быть известны данные о персонализируемом платежном приложении

[1] Если установлен уровень безопасности «Конфиденциальность и целостность», то будет выполняться двойное зашифрование конфиденциальных данных.

[2] Физическое устройство персонализации не исключается из системы персонализации, но оно просто выполняет уже сформированные команды.

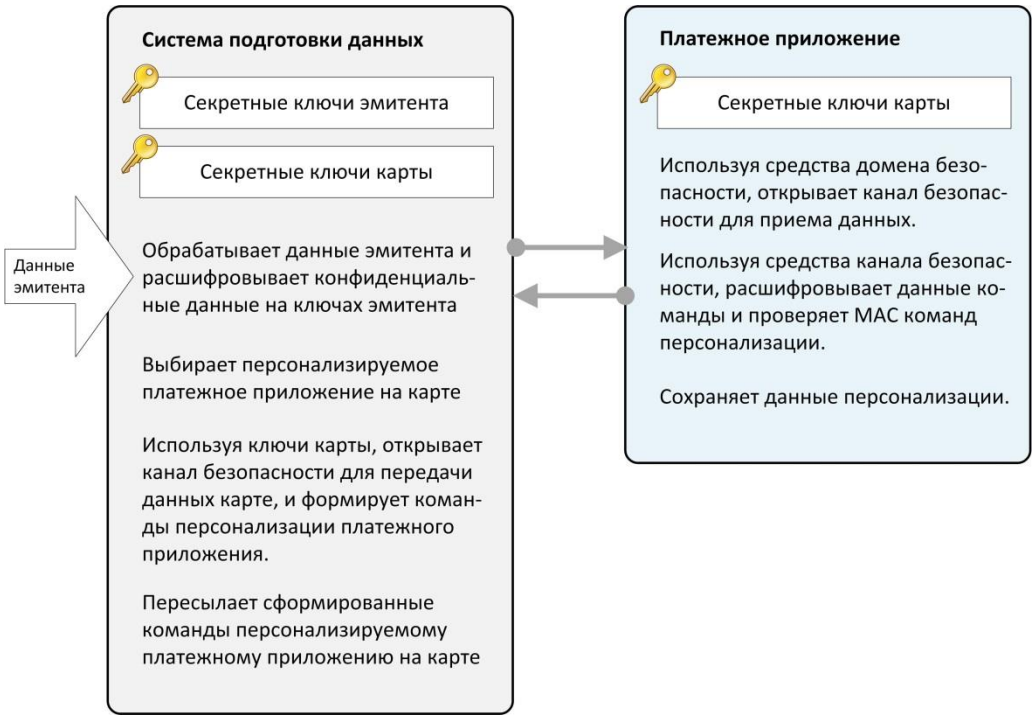

Рис. 9. Прямой метод персонализации.

В результате того, что система подготовки данных формирует команды для персонализации платежного приложения, можно отказаться от зашифрования конфиденциальных данных на сессионном ключе шифрования конфиденциальных данных. Действительно, намного проще установить уровень безопасности «Конфиденциальность и целостность» для канала безопасности и все данные передавать в зашифрованном виде. Именно такой подход и продемонстрирован на рис. 9.

Непрямой метод персонализации хорош тем, что позволяет максимально абстрагироваться от особенностей персонализируемой карты. Но у него есть существенный недостаток. По сравнению с прямым методом на персонализацию карты тратится намного больше времени.

Остальная информация из этой главы является конфиденциальной. Она может быть получена после подписания Соглашения о неразглашении (Non-Disclosure Agreement – NDA). Обратитесь с запросом в СКАНТЕК.

.

ГЛАВА 3

Выбор платежного приложения

Для выбора платежного приложения на карте используется команда SELECT, с которой карте всегда передается идентификатор приложения (AID – Application Identifier). В ответ выбранное платежное приложение всегда возвращает FCI (File Control Information). FCI содержит некоторую информацию о платежном приложении (например, метку приложения, предпочтительные языки общения терминала с владельцем карты и т. п.).

Но главное в другом – только после успешного выбора приложения с помощью команды SELECT последующие команды будут направляться операционной системой карты выбранному приложению. Поскольку на карте может быть несколько приложений, операционной системе карты нужно знать, какому именно приложению адресованы команды и данные, подаваемые на карту. Вот это и определяется с помощью команды SELECT.

Для выбора платежного приложения в контактном и бесконтактном режимах используются разные методы. В контактном режиме процедура выбора платежного приложения состоит в построении списка приложений, одновременно поддерживаемых картой и терминалом (списка приложений-кандидатов), и в выборе из списка приложений-кандидатов приложения, с помощью которого будет совершена платежная операция. Различают два способа построения списка приложений-кандидатов:

- прямой или явный выбор приложения, когда терминал строит список приложений-кандидатов с помощью многократного опроса карты на предмет нахождения на ней конкретных приложений (терминал последовательно выдает команды SELECT с идентификаторами известных ему приложений и по коду возврата команды определяет, присутствует ли на карте приложение и следует ли поместить его в список приложений-кандидатов)

- непрямой или неявный выбор приложения, когда для выбора приложения используется приложение[1] PSE (Payment System Environment). В этом случае терминал сначала выбирает на карте приложение с идентификатором 1PAY.SYS.DDF01, которое возвращает SFI файла со списком приложений на карте. Терминал последовательно с помощью команды READ RECORD читает записи этого файла и извлекает из них информацию о приложениях, присутствующих на карте.

Для выбора приложения в бесконтактном режиме терминалу требуется не более двух команд SELECT. Эта процедура основана на использовании приложения PPSE (Proximity Payment System Environment), которое имеет идентификатор 2PAY.SYS.DDF01. В ответ на команду SELECT приложение PPSE возвращает объект FCI, который содержит информацию о бесконтактных приложениях на карте. Используя полученную информацию, терминал определяет, какие из поддерживаемых им приложений находятся на карте, локализует приложение с наивысшим приоритетом и второй командой SELECT выбирает его. Если терминал поддерживает единственное бесконтактное приложение, то он с самого начала использует команду SELECT с указанием идентификатора приложения.

Платежный аплет PayCORN имеет один и тот же идентификатор (AID) как для контактного, так и бесконтактного режима.[2] Более того, контактный и бесконтактный режим реализуется одним аплетом, а не двумя. Платежный аплет выбирает режим обработки транзакции (контактный или бесконтактный) автоматически в зависимости от метода использования карты. Из этого следует, что контактный и бесконтактный режим использования платежного приложения связаны между собой логически, поскольку для реализации этих режимов используются одни и те же данные, ключи, признаки особых ситуаций и т. п.

Остановимся на особенностях команды SELECT при выборе аплета в контактном и бесконтактном режимах.

[1] В терминологии EMV – это каталог, но будем называть его приложением по причинам, которые изложены в разделе «Данные на карте».

[2] По умолчанию AID имеет шестнадцатеричный вид F00000BAC25445. Однако, во время загрузки аплета на карту ему может быть присвоен и другой AID.

Контактный режим

В контактном режиме команда SELECT кодируется следующим образом.

Поле	Код	Описание
CLA	0x00	Команда общего назначения
INS	0xA4	SELECT
P1	0x04	Индикатор выбора приложения по AID[1]
P2	0x00	Индикатор выбора первого или единственного приложения на карте[2]
Lc	xx	Длина AID платежного приложения
Данные		AID платежного приложения, установленный при его загрузке на карту
Le	xx	Длина ответа на команду SELECT

Данные, возвращаемые в ответ на команду SELECT, представляют собой объект FCI (File Control Information) и имеют следующий вид.

Тэг	Значение		
6F	FCI Template		
	84	Dedicated File (DF) Name	
	A5	File Control Information (FCI) Proprietary Template	
		50	Application Label
		5F2D	Language Preference

В качестве значения объекта Dedicated File (DF) Name всегда указан AID платежного приложения. Значения объектов Application и Label Language Preference определяются во время персонализации платежного приложения.

[1] В соответствии с ISO 7816-4 – выбор приложения по имени DF, но мы уже договорились, что DF – это приложение на карте (см раздел «Данные на карте»). В ISO 7816-4 определены и другие признаки выбора, но к выбору приложения они никакого отношения не имеют.

[2] Параметр P2 команды указывает, определен полный или частичный AID выбираемого приложения. Для выбора платежного приложения рекомендуется использовать только полный AID, поэтому параметр P2 должен быть равен 0. В ISO 7816-4 определены и другие значения параметра P2, когда выбор файла (в терминологии ISO 7816-4) осуществляется по частичному AID.

В ответ на команду SELECT могут быть возвращены следующие байты состояния:

SW1	SW2	Описание
0x90	0x00	Приложение выбрано и готово к работе. В ответе предоставлен объект FCI.
0x62	0x83	Платежное приложение заблокировано и работа с ним возможна только в ограниченных пределах (см. раздел «Заблокированное платежное приложение»). В ответе предоставлен объект FCI.
0x6A	0x81	Карта заблокирована и работа с ней невозможна (см. раздел «Заблокированная карта»)
0x6A	0x82	Платежному приложению передана команда для выбора аплета, которого нет на карте
xx	xx	Общие коды ошибок (см. соответствующее приложение)

Бесконтактный режим

В бесконтактном режиме команда SELECT кодируется точно также как и в контактном режиме, отличается только формат возвращаемой информации. Формат команды следующий.

Поле	Код	Описание
CLA	0x00	Команда общего назначения
INS	0xA4	SELECT
P1	0x04	Индикатор выбора приложения по AID
P2	0x00	Индикатор выбора первого или единственного приложения на карте
Lc	xx	Длина AID платежного приложения
Данные		AID платежного приложения, установленный при его загрузке на карту
Le	xx	Длина ответа на команду SELECT

Данные, возвращаемые в ответ на команду SELECT, представляют собой объект FCI (File Control Information) и имеют следующий вид.

Тэг	Значение		
6F	FCI Template		
	84	Dedicated File (DF) Name	
	A5	File Control Information (FCI) Proprietary Template	
		50	Application Label
		5F2D	Language Preference
		9F38	Processing Options Data Object List (PDOL)

В качестве значения объекта Dedicated File (DF) Name всегда указан AID платежного приложения. Значения объектов Application и Label Language Preference определяются во время персонализации платежного приложения.

Processing Options Data Object List (PDOL) представляет собой список данных, которые должны быть переданы платежному приложению в команде GET PROCESSING OPTIONS. В список входят следующие объекты данных.

Тэг	Длина	Значение
C7	2	Terminal Transaction Processing Information (TTPI)
9F02	6	Amount, Authorized (Numeric)
9F1A	2	Terminal Country Code
5F2A	2	Transaction Currency Code
9A	3	Transaction Date
9C	1	Transaction Type
9F37	4	Unpredictable Number

Таким образом, общая длина данных, определенных в PDOL и предоставляемых в команде GET PROCESSING OPTIONS равна 20 байтам.

В ответ на команду SELECT могут быть возвращены следующие байты состояния:

SW1	SW2	Описание
0x90	0x00	Приложение выбрано и готово к работе. В ответе предоставлен объект FCI.
0x62	0x83	Платежное приложение заблокировано и работа с ним возможна только в ограниченных пределах (см. раздел «Заблокированное платежное приложение»). В ответе предоставлен объект FCI.
0x6A	0x81	Карта заблокирована и работа с ней невозможна (см. раздел «Заблокированная карта»)
0x6A	0x82	Платежному приложению передана команда для выбора аплета, которого нет на карте
xx	xx	Общие коды ошибок (см. соответствующее приложение)

Заблокированное платежное приложение

Эмитент может заблокировать платежное приложение, прислав в ответе на запрос авторизации элемент данных Card Status Update (CSU) с установленным признаком того, что приложение должно быть заблокировано.

Если выполнена успешная аутентификация эмитента (ответ эмитента признан достоверным), то приложение блокируется по требованию эмитента – переводится в состояние APPLICATION BLOCK. В этом состоянии текущая транзакция может быть завершена любым образом и после второй команды GENERATE AC могут быть выполнены команды некритичного скрипт-процессинга. Но любая следующая транзакция может быть обработана только с определенными ограничениями:

- команда SELECT для выбора платежного приложения всегда завершается с кодом, информирующим о том, что платежное приложение заблокировано и работа с ним возможна только в ограниченных пределах (но в ответе предоставляется объект FCI)

- после выбора платежного приложения без ограничения могут использоваться команды GET PROCESSING OPTIONS, GET DATA, READ RECORD, GET CHALLENGE и VERIFY

- на команду GENERATE AC платежное приложение всегда возвращает криптограмму AAC (отвергает транзакцию)

- не допускаются никакие команды скрипт-процессинга, кроме команды APPLICATION UNBLOCK, использующейся для разблокирования платежного приложения.

Разблокирование платежного приложения, осуществляемое с помощью команды APPLICATION UNBLOCK, может быть выполнено только на специальном терминале, поскольку процедура разблокирования отличается от стандартного алгоритма выполнения транзакции и скрипт-процессинга. Более подробная информация о разблокировании платежного приложения приведена в разделе «Команда APPLICATION UNBLOCK».

Заблокированная карта

Эмитент может заблокировать не только платежное приложение, как это описано в предыдущем разделе, но и карту. Если выполнена успешная аутентификация эмитента (ответ эмитента признан достоверным), то карта блокируется по требованию эмитента. После этого работа ни с одним из приложений на карте становится невозможной.

В платежном приложении обработка требования эмитента по блокированию карты выполняется следующим образом:

- платежное приложение переводится в состояние CARD BLOCK
- с помощью средств GlobalPlatform среде GlobalPlatform Environment, отвечающей за управление состоянием карты, сообщается, что карта должна быть заблокирована

Для того, чтобы платежное приложение могло заблокировать карту, должны выполняться следующие условия:

- платежное приложение должно иметь привилегию Card Lock
- текущее состояние карты должно быть SECURED

Если карта успешно заблокирована, то текущая транзакция может быть завершена любым образом и после второй команды GENERATE AC могут быть выполнены команды некритичного скрипт-процессинга. Но в дальнейшем при попытке выбора любого приложения на карте (а не только приложения, заблокировавшего карту), среда GlobalPlatform Environment будет сообщать, что карта заблокирована и работа с ней невозможна (команда SELECT для выбора любого аплета, кроме аплета с привилегий Final Application[1], возвращает байты состояния 6A81).

Разблокирование карты хотя и возможно, но не имеет никакого отношения к спецификациям EMV. Более подробная информация о блокировании и разблокировании карты приведена в документе «GlobalPlatform. Card Specification. Version 2.2. March 2006».

[1] Обычно, аплет с привилегий Final Application – это Issuer Security Domain.

ГЛАВА 4

Команды аплета

Терминал взаимодействует с платежным аплетом с помощью команд, которые он направляет карте. В ответ терминал получает данные, необходимые ему для обработки платежной операции. В этой главе описаны все команды аплета, которые используются терминалом для обработки транзакции. Часть команд аплета определяется спецификациями EMV, а часть является командами общего назначения, которые определены в ISO 7816-4 (например, команда READ RECORD).

Единственная команда, которая не описана в данной главе, но которая также может потребоваться терминалу при обработке транзакции, – это команда GET RESPONSE. Связано это с тем, что необходимость этой команды определяется используемым протоколом обмена данными с картой, а детали обмена обычно скрыты от терминала. Команда GET RESPONSE описана в ISO 7816-4.

Необходимо сказать, что логика выполнения всех команд, за исключением команды GET PROCESSING OPTIONS, не зависит от режима, в котором используется аплет – контактном или бесконтактном. Команда GET PROCESSING OPTIONS в контактном и бесконтактном режимах выполняет совершенно различные функции, на что следует обратить особое внимание.

Описание команд приведено в алфавитном порядке. Но перед описанием команд следует определить некоторые внутренние признаки и переменные платежного аплета.

Остальная информация из этой главы является конфиденциальной. Она может быть получена после подписания Соглашения о неразглашении (Non-Disclosure Agreement – NDA). Обратитесь с запросом в СКАНТЕК.

ГЛАВА 5

Кодирование элементов данных

В этой главе описывается кодирование элементов данных, используемых при обработке транзакции. Элементы данных приведены в алфавитном порядке. Большинство описанных элементов данных используются независимо от режима выполнения транзакции (контактного или бесконтактного). Если какой-либо элемент данных используется только в бесконтактном режиме, то это специально оговаривается.

В таблицах, описывающих установку бит в байтах, все биты определены слева направо. Самый левый (старший) бит имеет номер 8, а самый правый (младший) – номер 1. Все байты в полях нумеруются с 1, начиная с самого левого байта поля. Зарезервированные поля и биты, помеченные как RFU, должны быть установлены в 0.

Остальная информация из этой главы является конфиденциальной. Она может быть получена после подписания Соглашения о неразглашении (Non-Disclosure Agreement – NDA). Обратитесь с запросом в СКАНТЕК.

ГЛАВА 6

Криптографические алгоритмы

В данной главе приведено описание криптографических алгоритмов, которые используются для обработки транзакции картой и терминалом. Все эти алгоритмы подробно описаны в спецификациях EMV, которые являются основным документом для реализации алгоритмов криптографических операций в различных компонентах платежной системы. Здесь эти алгоритмы приведены только для справки и не должны использоваться в других целях.

Кроме того, в данной главе описан алгоритм расшифрования PIN-блока, полученного от устройства ввода PIN-кода, использующего алгоритм DUKPT (Derived Unique Key Per Transaction). DUKPT описан в «ANSI X9.24. Retail Financial Services Symmetric Key Management. Part 1: Using Symmetric Techniques». Здесь краткое изложение DUKPT приведено только для информации.

Далее используются следующие соглашения по определению криптографических операций.

1. Для определения операций зашифрования и расшифрования данных с использованием алгоритма Triple DES используются следующие обозначения:

   - 3DESECB (Data, K) – зашифрование данных Data на ключе K в режиме ECB
   - $3DESECB^{-1}$ (Data, K) – расшифрование данных Data на ключе K в режиме ECB
   - 3DESCBC (Data, K, IV) – зашифрование данных Data на ключе K в режиме CBC с начальным вектором IV
   - $3DESCBC^{-1}$ (Data, K, IV) – расшифрование данных Data на ключе K в режиме CBC с начальным вектором IV

2. Для обозначения операция получения 8-ми байтной подписи данных по алгоритму Triple DES (MAC) используется следующая формула:

   - 3DESMAC (Data, K, IV) – получение подписи данных Data на ключе K в режиме CBC с начальным вектором IV

3. Операции зашифрования и расшифрования данных с использованием алгоритма DES обозначаются следующим образом:

   - DESECB (Data, K) – зашифрование данных Data на ключе K в режиме ECB

   - $DESECB^{-1}$ (Data, K) – расшифрование данных Data на ключе K в режиме ECB

4. Операция получения 8-ми байтной подписи данных по алгоритму DES (MAC) определяется через следующую формулу:

   - DESMAC (Data, K, IV) – получение подписи данных Data на ключе K в режиме CBC с начальным вектором IV

5. Для определения операций зашифрования и расшифрования данных с использованием алгоритма RSA используются следующие обозначения:

   - RSASPRV (Data, Ks) – получение подписи данных Data на секретном ключе Ks (т. е. зашифрование данных на секретном ключе)

   - RSARPUB (Sign, Kp) - восстановление данных из подписи Sign на открытом ключе Kp (т. е. расшифрование данных на открытом ключе)

Остальная информация из этой главы является конфиденциальной. Она может быть получена после подписания Соглашения о неразглашении (Non-Disclosure Agreement – NDA). Обратитесь с запросом в СКАНТЕК.

ГЛАВА 7

Приложения

Эта глава содержит сведения справочного характера, которые могут использоваться для анализа обмена данными при выполнении платежной транзакции и протекании процессов в терминале и платежном приложении. Частично эти сведения представлены и других главах документа, но только в этой главе они систематизированы и сведены в таблицы.

Объекты данных

В приведенной ниже таблице показаны элементы данных, которые используются при выполнении платежной транзакции, и их отображение в объекты данных, которыми обмениваются терминал, платежное приложение и эмитент. Элементы данных сгруппированы в алфавитном порядке.

В колонках S, F, T и L приведены следующие данные:

- S – источник данных:
  - T – терминал
  - ICC – карта
  - I – эмитент
- F – формат данных:
  - b – двоичные данные
  - n – десятичные цифры (по две цифры на байт), которые выровнены по правой границе и дополнены слева нулем
  - cn - десятичные цифры (по две цифры на байт), которые выровнены полевой границе и дополнены справа шестнадцатеричной цифрой F
  - an – алфавитно-цифровые символы (допускаются буквы A – Z в любом регистре и цифры 0 – 9)
  - ans – алфавитно-цифровые символы (допускаются буквы A – Z в любом регистре и цифры 0 – 9) и специальные символы из первой половины таблицы ANSI
- T – тэг объекта данных
- L – длина значения объекта данных.

В таблице специальным цветом выделены строки для тех объектов данных платежного приложения, значения которых могут быть получены по команде GET DATA (эти объекты не предоставляются в записях платежного приложения).

Информация об объектах данных является конфиденциальной. Она может быть получена после подписания Соглашения о неразглашении (Non-Disclosure Agreement – NDA). Обратитесь с запросом в СКАНТЕК.

Общие коды ошибок

При выполнении любой команды может быть зафиксирована ошибка, которая либо не связана с логикой выполнения команды (касается только синтаксиса), либо связана с текущим состоянием платежного приложения, либо не может быть локализована (вернее, локализация которой не выполняется на уровне платежного приложения, исполняющего команду). Для информирования о таких ошибках используются общие коды ошибок.

Общие коды ошибок команд приведены в следующей таблице:

SW1	SW2	Описание
0x67	0x00	Некорректная длина данных (в поле Lc или Le)
0x6A	0x86	Некорректные параметры команды (P1/P2)
0x6D	0x00	Неправильная инструкция (поле INS)
0x6E	0x00	Неправильный класс команды (поле CLA)
0x6F	0x00	При выполнении команды произошло программное исключение (данный код должен трактоваться, как серьезная ошибка, которая требует дополнительного изучения)

Дополнительная документация

При чтении документации на платежный аплет PayCORN может потребоваться дополнительное изучение следующих спецификаций и стандартов.

1. EMV. Integrated Circuit Card Specifications for Payment Systems. Book 1. Application Independent ICC to Terminal Interface Requirements. Version 4.2. June 2008.

2. EMV. Integrated Circuit Card Specifications for Payment Systems. Book 2. Security and Key Management. Version 4.2. June 2008.

3. EMV. Integrated Circuit Card Specifications for Payment Systems. Book 3. Application Specification. Version 4.2. June 2008.

4. EMV. Integrated Circuit Card Specifications for Payment Systems. Book 4. Cardholder, Attendant, and Acquirer Interface Requirements. Version 4.2. June 2008.

5. EMV. Contactless Specifications for Payment Systems. Book A. Architecture and General Requirements. Version 2.1. March 2011.

6. EMV. Contactless Specifications for Payment Systems. Book B. Entry Point Specification. Version 2.1Ю March 2011.

7. EMV Card Personalization Specification. Version 1.1. July 2007.

8. ISO/IEC 7816-4. Identification Cards – Integrated Circuit Cards. Part 4. Organization, security and commands for interchange.

9. ISO/IEC 14443-1. Identification Cards – Contactless Integrated Circuit Cards – Proximity Cards. Part 1. Physical characteristics. 2008.

10. ISO/IEC 14443-2. Identification Cards – Contactless Integrated Circuit Cards – Proximity Cards. Part 2. Radio frequency power and signal interface. 2010.

11. ISO/IEC 14443-3. Identification Cards – Contactless Integrated Circuit Cards – Proximity Cards. Part 3. Initialization and anti-collision. 2011.

12. ISO/IEC 14443-4. Identification Cards – Contactless Integrated Circuit Cards – Proximity Cards. Part 4. Transmission protocol. 2008.

13. GlobalPlatform. Card Specification. Version 2.2. March 2006.

14. ISO/IEC 8825-1. Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER). 2008.

15. ISO 8583-1. Financial transaction card originated messages – Interchange message specifications. Part 1: Messages, data elements and code values. 2003.

16. ISO 8583-2. Financial transaction card originated messages – Interchange message specifications. Part 2: Application and registration procedures for Institution Identification Codes (IIC). 1998.

17. ISO 8583-3. Financial transaction card originated messages – Interchange message specifications. Part 3: Maintenance procedures for messages, data elements and code values. 2003.

18. ISO 4217. Currency Codes. 2008.

19. ISO 639-1. Codes for the Representation of Names of Languages. Part 1. Alpha-2 code. 2002.

20. ISO 8583-1. Financial transaction card originated messages – Interchange message specifications. Part 1. Messages, data elements and code values. 1987.

21. ANSI X9.24-2009. Retail Financial Services Symmetric Key Management. Part 1: Using Symmetric Techniques. 2009.

22. ISO 9564-1. Financial services – Personal Identification Number (PIN) management and security. Part 1: Basic principles and requirements for PINs in card-based systems. 2011.

Если вы хотите более глубоко изучить проблемы, связанные с безналичными платежами, платежными картами, спецификациями EMV, и еще узнать много интересного, рекомендуется прочесть книгу Игоря Михайловича Голдовского «Банковские микропроцессорные карты». ISBN 978-5-9614-1233-8; 2010 г.

ДЛЯ ЗАМЕТОК