Common Payment Application Contactless Extension

CPACE

Functional Specification

Terminal Kernel

Version 1.0
12.07.2018

Revision History

Version	Date	Author	Object
1.0	2018-07-12	ECPC	First public release

Table of Contents

Tables

1 Introduction

CPACE Terminal Kernel is designed for interoperability with a CPACE payment application. This document specifies the functionalities required to implement the CPACE Terminal Kernel.

CPACE Terminal Kernel is to be integrated in a POS System that follows EMV Specifications to accept contactless payments. It is to be used as the other kernels specified in EMV contactless specifications, but is specific to the acceptance of cards and mobile devices with a CPACE payment application.

The CPACE Terminal Kernel may also be integrated in an ATM System.

1.1 Document organization

This document includes the following sections and annexes:

Section 1: contains general information that helps understanding and using this specification.

Section 2: contains the references, abbreviations and definitions used in this document.

Section 3: presents the specification framework followed in this document.

Section 4: provides a high-level overview of transaction processing according to this specification.

Section 5: defines the extensions on requirements on the Entry Point to work with a Kernel compliant with this specification.

Section 6: defines the data elements the Terminal shall provide to the Kernel as well as the data elements returned by the Kernel to the Entry Point.

Section 7: Describes the card commands coding and the data elements returned

Sections 8 – 17: specify the processing that shall be performed by the Kernel in a contactless transaction and the card commands and requirements for the transaction.

Sections 18 – 20: clarify that Online, Issuer Script and Completion Processing functions are out of the scope of the Kernel.

Section 21: specifies the default procedures of the Kernel to handle the errors that may occur while processing a transaction, and that have not been treated elsewhere in the specification.

Section 22: specifies the Kernel Outcomes and related parameters.

Section 23: details the Data Elements that are new or that have changed compared to the EMV specifications.

2 References, Abbreviations and Document Conventions

2.1 References

Reference	Title	Date
[CPACE-DIC]	Common Payment Application Contactless Extension, Functional Specification, CPACE for Dual Interface Cards, Version 1.0	2017-10
[CPACE-HCE]	Common Payment Application Contactless Extension, Functional Specification, CPACE for Host Card Emulation, Version Draft 1.0	
[EMV Book A]	EMV Contactless Specifications for Payment Systems, Architecture and General Requirements, Version 2.7	2018-04
[EMV Book B]	EMV Contactless Specifications for Payment Systems, Entry Point Specification, Version 2.7	2018-04
[EMV Book D]	EMV Contactless Specifications for Payment Systems, Contactless Communication Protocol Specification, Version 2.7	2018-04
[EMV Book 2]	EMV Integrated Circuit Card Specifications for Payment Systems, Book 2, Security and Key Management, Version 4.3	2011-11
[EMV Book 3]	EMV Integrated Circuit Card Specifications for Payment Systems, Book 3, Application Specification, Version 4.3	2011-11
[EMV Book 4]	EMV Integrated Circuit Card Specifications for Payment Systems, Book 4, Application Specification, Version 4.3	2011-11
[ISO 3166-1]	Codes for the representation of names of countries and their subdivisions – Part 1: Country codes	
[ISO 7810]	Identification cards – Physical characteristics	
[ISO 7816]	Identification cards – Integrated circuit cards	
[ISO 14443]	Identification cards – Contactless integrated circuit cards - Proximity cards	

2.2 Definitions

The following definitions are used in this specification. In case there is a conflict between definitions in the References and applicable EMV Specification Bulletins published after, and the definitions hereafter, the last prevail.

Approved — A final Outcome from the Kernel, meaning that the transaction was successful.

ATM System	The ATM System is the term given to an ATM that integrates a contactless Reader.
Card	As used in these specification, the same as Contactless Payment Device.
Card Authentication Method	Method used to authenticate the card. Based on the Application Cryptogram for online CAM and on the Offline Data Authentication for the offline CAM
Contactless Payment Device	A Device that performs the contactless payments. In this specification it may be a Dual Interface Card, a Contactless only Payment Device or a Mobile Device.
Contactless Only Payment Device	A Contactless Payment Device without a cardholder interface.
Contactless Reader	The Contactless Reader is the device that supports the Kernel(s) and provides the contactless interface used by the Card. It is considered in this specification as a separate logical entity, although it may be an integral part of the POS System.
Declined	A Final Outcome from the Kernel, meaning that the transaction was not successful.
Dual Interface Card	A type of Contactless Payment Device, with contact and contactless interfaces, supported by this specification.
Entry Point	In this specification, Entry Point is the software in the POS System that is responsible for: • Performing pre-processing; • Selecting a contactless application that is supported by both the card and the reader; • Activating the appropriate Kernel; • Processing of the Outcome returned by the Kernel, and passing selected Outcomes to the Terminal.
Final Outcome	A result provided to the Terminal: • Upon Entry Point processing of an Outcome from the Kernel; or • Provided directly by the Entry Point under exception conditions.
Kernel	The Kernel implements interface routines, security and control functions necessary to carry out a contactless transaction. It manages a set of commands and responses to retrieve data from the Card.

Mobile Device	A Contactless Payment Device with cardholder interface (data entry and output capability), supporting either HCE (Host Card Emulation) or SE (Secure Element).
Online Request	A Final Outcome from the Kernel, meaning that the transaction needs to be approved online.
Outcome	A result from the Kernel processing, provided to the Entry Point.
POS System	The POS System is the term given to the payment infrastructure present at the merchant. It is made up of the Terminal and Reader. As used in this specification, it may also refer to an ATM System.
Reader	As used in this specification, the same as Contactless Reader.
Reader Contactless Floor Limit	Establishes the amount above which the reader requires online processing for the transaction.
Reader Contactless Transaction Limit	Establishes the amount above which a contactless transaction is not permitted.
Reader CVM Required Limit	Establishes the amount above which cardholder verification shall be performed.
Select Next	An Outcome from the Kernel, meaning that the Entry Point shall try to select the next application.
Terminal	Entity that connects to the Acquirer network and that, together with the Reader, makes up the POS System. The Terminal and the Reader are considered separate logical entities; they may physically exist in a single integrated device.
Try Again	An Outcome from the Kernel, meaning that the transaction shall be repeated.
Try Another Interface	A Final Outcome from the Kernel, meaning that the transaction shall be repeated using a different interface.
User Interface (UI) Request	A Request that the Kernel invokes to the Terminal, in order to display a Message. The Message is referenced by an Identifier. E.g. '21' ("Present Card Again"). This request may be invoked alone or as a Transaction Outcome parameter.

2.3 Abbreviations

The following definitions are used in this specification. In case there is a conflict between definitions in 2.1 References and applicable EMV Specification Bulletins published after, and the definitions hereafter, these prevail.

ATM	Automated Teller Machine
CAM	Card Authentication Method
CDCVM	Consumer Device Cardholder Verification Method

CHV&CS	Cardholder Verification and Confirmation Status
GPO	GET PROCESSING OPTIONS
HCE	Host Card Emulation
ODA	Offline Data Authentication
POS	Point of Sale
SE	Secure Element
UI	User Interface

2.4 Document conventions

This specification uses the data element format conventions and terminology defined in Sections 4.3 and 4.4 of [EMV Book 3] and Section 4 of [EMV Book A]. This specification uses the notation defined in Section 4.2 of [EMV Book 3], with the additions and modifications described in Section 2.4.1 below.

In this specification, the term "data dictionary" refers to Annex A of [EMV Book 3] and [EMV Book 4], with the additions and modifications in Section 23 of this document.

Whenever there is the need to introduce modifications on the EMV specification text, the following conventions apply:

- Deletions to the text extracted from EMV specifications are represented by ~~strikethrough~~;
- Additions to the EMV specifications text are represented by underlining.

Example:

If the status words ~~'6985' are~~ returned are different from ‘9000’, the ~~terminal~~ Kernel shall ~~eliminate the current application from consideration and return to the Application Selection function to select another application~~ return the Control to Entry Point with an Outcome “Select Next” as in Section 22.2.10.

2.4.1 Notation

In accordance with the EMV specification (e.g. [EMV Book 3] Section 5, Annexes A and B), the following notation is used for data description in this specification:

- An item of information is called a **data element**. A data element is the smallest piece of information that may be identified by a name, a description of logical content, a format, and a coding.
- A **data object** consists of a tag, a length, and a value (TLV). The value field of a data object may consist of either a single data element or one or more data objects. When a data object encapsulates a single data element, it is called a **primitive data object**. When a data object encapsulates one or more data objects, it is called a **constructed data object**. The value field of a constructed data object is called a **template**.

The names of templates and data elements defined in the data dictionary and used in this specification are written in italics to distinguish them from the text, e.g. *Application Interchange Profile*.

In addition to or as replacement of those described in Section 4.2 of [EMV Book 3], the notational conventions described below are used in this specification.

'Name of Sub-Element' in *Data Object Name*	Reference to a sub-element of a data object defined in the data dictionary, e.g. 'CVM Condition' in *CVM Results*
A <> B	Value of A is different from the value of B.
A <= B	Value of A is less than or equal to the value of B.
A >= B	Value of A is greater than or equal to the value of B.
A XOR B	The bit-wise exclusive-OR of the data blocks A and B. If one data block is shorter than the other then it is first padded to the left with sufficient binary zeros to make it the same length as the other.
[x:y]	Range of bytes of the referenced data element. For example, *Additional Terminal Capabilities*[1:3] represents bytes 1, 2, and 3 of the *Additional Terminal Capabilities.*
[bx:y]	Range of bits of the referenced data element. For example, *Application Priority Indicator*[b4:1] represents bits 4, 3, 2, and 1 of *Application Priority Indicator.*

3 Kernel specification framework

3.1 Introduction

The CPACE Terminal Kernel Functional Specification shall be read in conjunction with the EMV specifications referenced in Section 2.1. It does not intend to duplicate the content of the EMV specifications.

The section 3.2 defines the relationship between this document and EMV documents, identifying:

- Sections in EMV documents not to be considered for the CPACE Terminal Kernel;
- Sections in EMV documents that shall be modified to be considered for the CPACE Terminal Kernel;
- Sections added to this document, relevant for the Kernel Specification and that does not exist in EMV documents.

3.2 Relationship with EMV specifications

The CPACE Terminal Kernel specified in this document is meant to be integrated in a POS System that is aligned with the architecture defined in [EMV Book A] and supports the Entry Point specified in [EMV Book B] with the modifications described in section 5, and has a contactless interface compliant with [EMV Book D].

This specification is to be used in the same way as the other contactless kernels specified in EMV contactless specifications (Books C1 to C7), but is specific to the acceptance of contactless cards and mobile devices with a payment application compliant with [CPACE-DIC] and [CPACE-HCE].

The CPACE Terminal Kernel specification is based on EMV specifications for contact cards, as defined in [EMV Book 2], [EMV Book 3] and [EMV Book 4].

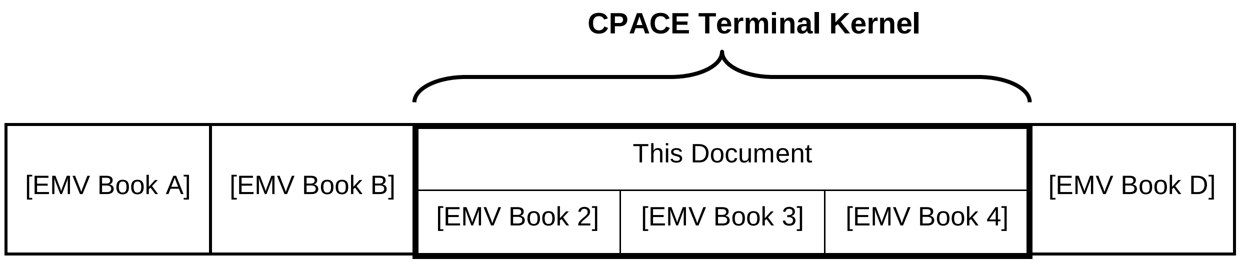

Figure 1: Specifications relationship

The CPACE Terminal Kernel shall be implemented according to [EMV Book 2] and [EMV Book 3] over a contactless interface according to [EMV Book D] instead of a contact interface according to [ISO 7816] Part 3.

Whenever “Terminal” is mentioned in [EMV Book 2] and [EMV Book 3], it shall be read as “Kernel” for the functions in the scope of the CPACE Terminal Kernel specification, as defined below.

3.2.1 Not supported EMV functions

The following functions defined in EMV specifications are not supported by CPACE Terminal Kernel. Specific data elements associated with these functions are not supported as well.

[EMV Book B]:

- Pre-Processing.

[EMV Book 2]:

- Static Data Authentication (SDA);
- Dynamic Data Authentication (DDA);
- Personal Identification Number (PIN) Encipherment;
- Secure Messaging.

[EMV Book 3]:

- Online Processing[1];
- Issuer-to-Card Script Processing;
- Completion[1].

3.2.2 Modified EMV functions

The following functions defined in EMV specifications shall be modified as described later in this document:

[EMV Book A]:

- Entry Point Processing.

[EMV Book B]:

- Combination Selection.

[EMV Book 3]:

- Initiate Application Processing;
- Offline Data Authentication;
- Cardholder Verification;
- Terminal Risk Management;
- Card Action Analysis.

Whenever the function flowcharts in the modified sections of the EMV Books have not been adapted according to the modified or new requirements in this specification, the text in this specification prevails.

[1] To be performed outside the kernel by the Terminal.

3.2.3 Additional functionalities

Additional functionalities not covered by EMV specifications, are needed for CPACE Contactless transactions and shall be implemented by the Kernel, according to this specification:

- Relay Resistance Protocol;
- Transaction Outcome.

3.2.4 Document sections mapping

Table 1 establishes a relationship between sections in this specification and sections in EMV specifications.

Section of this Specification		**Corresponding Section of EMV specifications**	
1	Introduction	--	--
2	References, Abbreviations and Document Conventions	[EMV Book A] and [EMV Book 3] Section 4	--
3	Kernel Specification Framework	--	--
4	CPACE transaction processing	[EMV Book 3] Section 8	Transaction Flow
5	Requirements on Entry Point	[EMV Book A] Section 5.8.2	Application Selection and Kernel Activation
		[EMV Book B] Section 3.3	Combination Selection
6	Kernel Input and Output Data	--	--
7	Commands for Financial Transactions	[EMV Book 3] Section 6	Commands for Financial Transaction
8	Kernel Activation	--	--
9	Initiate Application Processing	[EMV Book 3] Section 10.1	Initiate Application Processing
10	Relay Resistance Protocol	--	--
11	Read Application Data	[EMV Book 3] Section 10.2	Read Application Data

12	Offline Data Authentication	[EMV Book 3] Section 10.3	Offline Data Authentication
13	Processing Restrictions	[EMV Book 3] Section 10.4	Processing Restrictions
14	Cardholder Verification	[EMV Book 3] Section 10.5	Cardholder Verification
15	Terminal Risk Management	[EMV Book 3] Section 10.6	Terminal Risk Management
16	Terminal Action Analysis	[EMV Book 3] Section 10.7	Terminal Action Analysis
17	Card Action Analysis	[EMV Book 3] Section 10.8	Card Action Analysis
18	Online Processing	[EMV Book 3] Section 10.9	Out of scope of the CPACE Kernel Specification
19	Issuer-to-Card Script Processing	[EMV Book 3] Section 10.10	Not supported by the CPACE Kernel
20	Completion	[EMV Book 3] Section 10.11	Out of scope of the CPACE Kernel Specification
21	Error Handling	--	--
22	Transaction Outcomes	[EMV Book A] Section 6.2	Outcome Parameters
23	Data Elements Dictionary	[EMV Book A], [EMV Book 3], [EMV Book 4] Annex A	Data Elements Dictionary

Table 1: Relationship between CPACE Terminal Kernel Functional Specification and EMV Specifications

3.3 Contactless Payment Devices supported

CPACE Terminal Kernel supports the following cardholder Contactless Payment Devices:

- Dual interface card (ID 1 format according to ISO 7810);
- Contactless only Payment Device without a cardholder interface, (e.g. a contactless only chip card in ID 1-Format or in another format, a watch, a wristband, a key fob, a ring or a sticker);
- Mobile Device with Host Card Emulation (HCE) or Secure Element (SE).

The card's data element **'**Device Type' in *Third Party Data* identifies the type of device used in a contactless transaction.

Whenever "card" or "contactless card" is used in this specification, it may refer to any of the contactless payment devices mentioned above, unless otherwise stated.

Whenever "ICC" is mentioned in EMV specifications, it may refer to any of the payment devices mentioned above.

4 Transaction processing

4.1 Introduction

Transactions that require card authentication and/or cardholder verification, shall follow the CPACE transaction flow, as specified in Section 4.2. This is the case for, but not limited to:

- Purchase
- Cash Advance
- Cash Withdrawal
- Cashback

For transactions that do not require card authentication nor cardholder verification, as is the case of the Refund transaction, the Kernel shall perform the simplified transaction flow specified in Section 4.3.

4.2 Transaction flow

The CPACE Terminal Kernel processing follows [EMV Book 3] Section 8, with the modifications specified in Sections 7 to 17.

The CPACE transaction flow is implemented by the CPACE Terminal Kernel, as depicted in Figure 2. In addition, Figure 2 shows a POS System consisting of a Terminal and a Contactless Reader, and the relationship between the Kernel, the other POS System components and the contactless payment device (e.g. card, wristband or mobile device).

The Kernel is activated by the Entry point, after performing Card Activation and Application Selection, according to [EMV Book B]. After being activated, the Kernel continues the transaction processing, from Initiate Application Processing to Card Action Analysis, and finishes by returning the control to the Entry Point, informing about the Transaction Outcome.

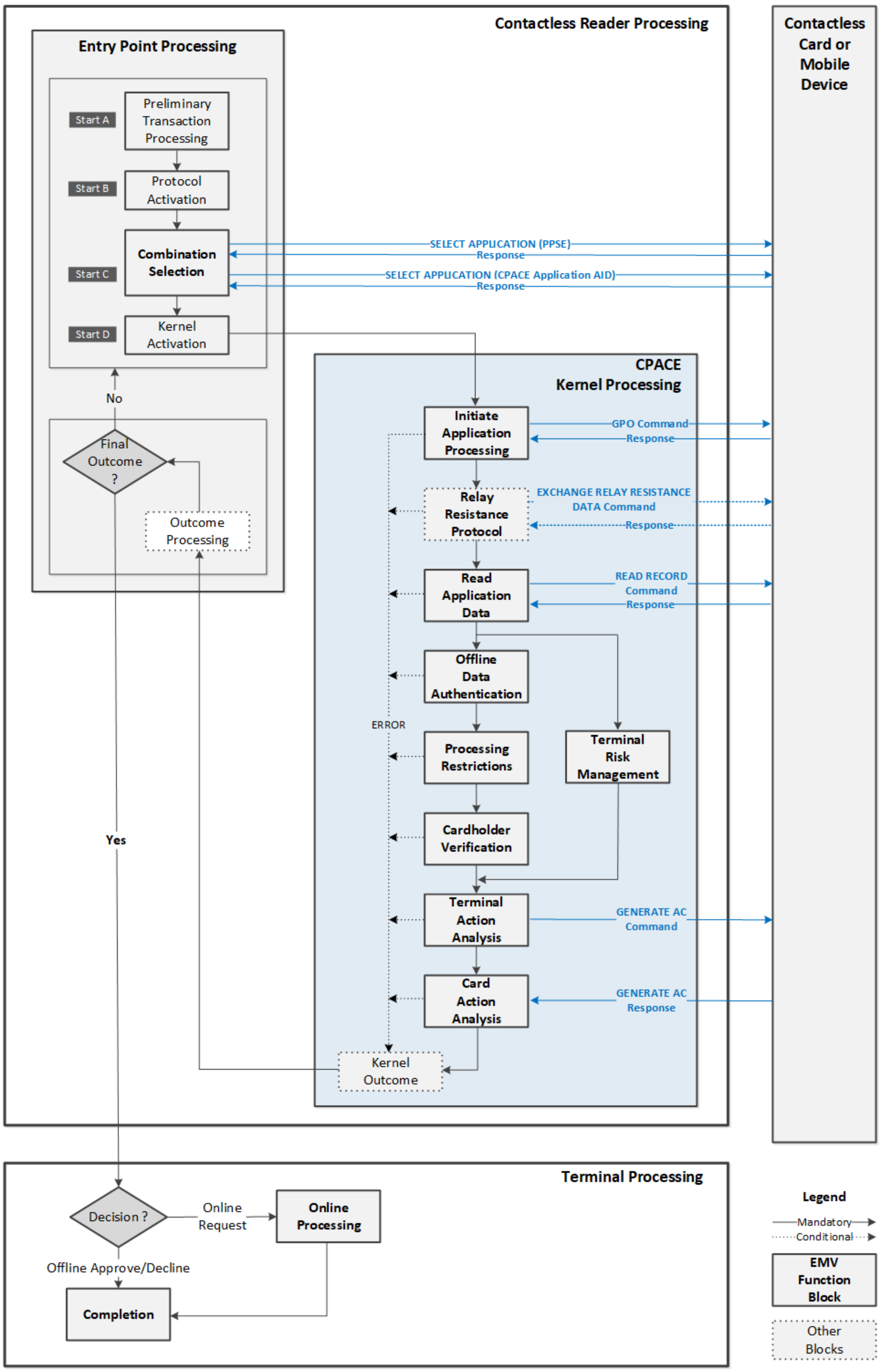

Figure 2: POS System and the contactless transaction flow

4.3 Simplified Transaction flow

For transactions that do not require card authentication nor cardholder verification, as is the case of the Refund transaction, the Kernel shall perform a simplified transaction flow as depicted in Figure 3.

During Terminal Action Analysis regardless the values of *Terminal Verification Results (TVR)*, *Issuer Action Codes* and *Terminal Action Codes*, the Kernel shall issue a GENERATE AC command requesting an AAC from the card with *Transaction Type* := Refund.

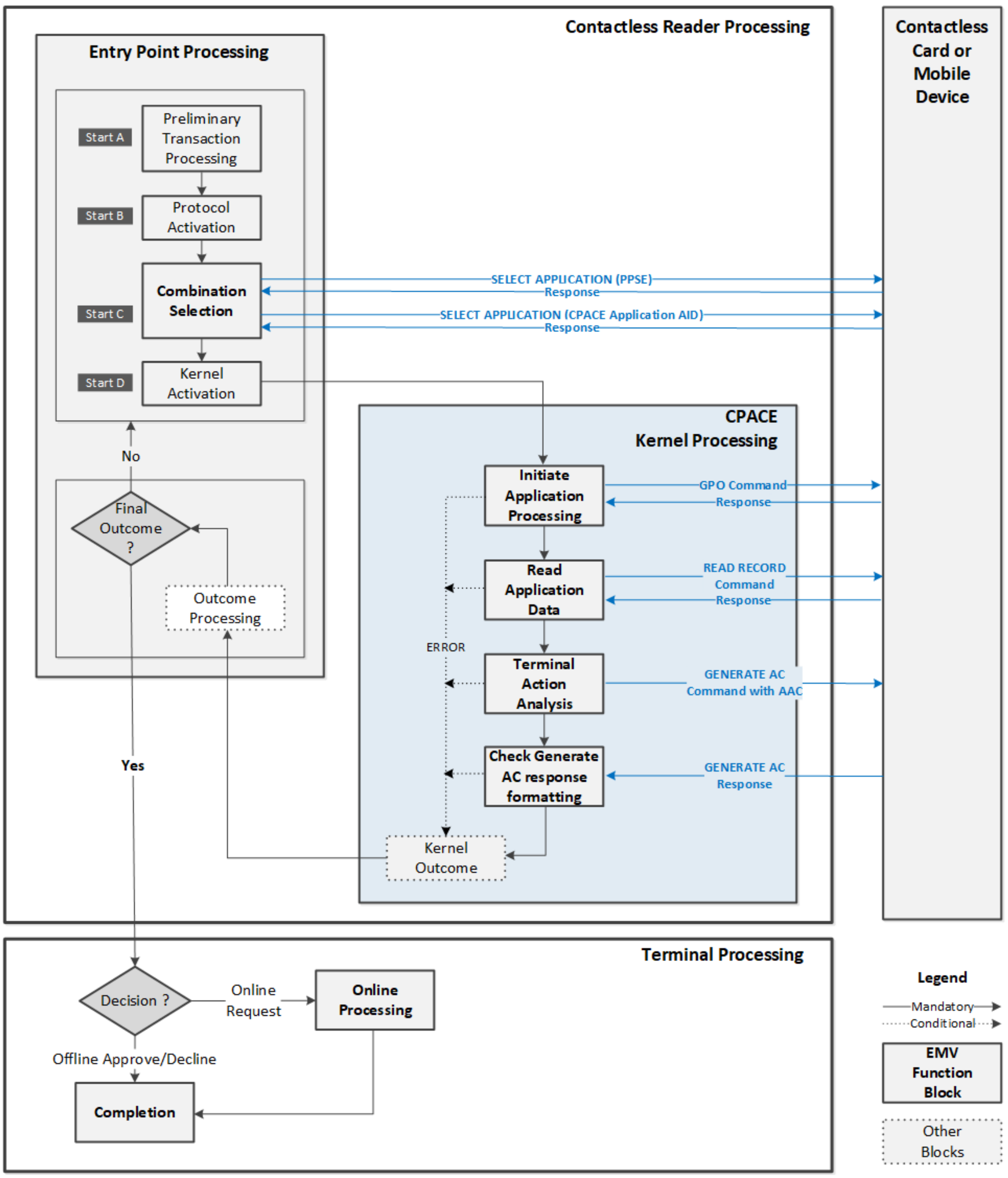

Figure 3: POS System and the contactless simplified transaction flow

5 Extensions on the requirements on Entry Point

Although the Entry Point is out of scope of this specification it is expected that an Entry Point that activates the CPACE Terminal Kernel complies with the requirements defined in this section.

Application and Combination Selection shall be performed by Entry Point as described in [EMV Book A] and [EMV Book B] with the following extensions:

In section 5.6.5 and 5.7 of [EMV Book A]:

The Entry Point Pre-Processing Indicators defined in Table 5-3 of section 5.7 are not needed for the CPACE Terminal Kernel. Therefore, Entry Point Configuration Data defined in Table 5-2 of section 5.6.5 shall not be configured for the combinations using the Kernel defined in this specification, and the Entry-Point may skip Pre-Processing defined in section 3.1 of [EMV Book B].

In section 5.8.2 of [EMV Book A] and section 3.3.2 of [EMV Book B]:

The list of Kernels shown in section 5.8.2 of [EMV Book A] and section 3.3.2 of [EMV Book B] has to be extended to assign the CPACE Terminal Kernel Identifier to the AIDs of schemes using the CPACE Terminal Kernel as their default kernel.

Note: Since the number of AIDs using the CPACE Terminal Kernel may change over time, it is recommended to make this configurable, at least for the CPACE Terminal Kernel.

6 Kernel Input and Output Data

Prior to activation of the Kernel for a transaction, the Terminal shall configure the Kernel with the Data Objects needed for its correct operation. The Kernel will use default Data Objects if not configured by the Terminal as defined in 6.1.1.

Whenever the Kernel is activated for a transaction, the Entry Point shall send to the Kernel the Selected Application FCI and the transaction's data as defined in 6.1.2.

At the end of a transaction the Kernel shall return to the Entry Point the Outcome with the parameters defined in 6.2.1 and if the transaction reaches a final state with no errors, a Data Record with all the data collected during the processing of the transaction as defined in 6.2.2.

6.1 Input Data

6.1.1 Configuration Data

Data that shall be configured by the Terminal prior to the activation of the Kernel to perform a contactless transaction. For the Data Objects not configured by the Terminal Default Values shall be used.

Data Object	**Default Value**
Additional Terminal Capabilities	'0000000000'
Application Version Number	'0001'
CHV&CS Message Table	See 23.4
Contactless Transaction Limit with CDCVM	'000000000000'
Contactless Transaction Limit without CDCVM	'000000000000'
CVM Capabilities (above CVM Limit)	'00'
CVM Capabilities (below or equal CVM Limit)	'00'
Field Off Hold Time	'0D'
Kernel Configuration	'30'
Max Time Relay Resistance Tolerance	'0032'
Merchant Name and Location	''
Merchant Category Code	'0000'
Message Hold Time	'000013'
Min Time Relay Resistance Tolerance	'0014'
Reader Contactless Floor Limit	'000000000000'
Reader CVM Required Limit	'000000000000'
Relay Resistance Min Time Difference Limit	'012C'
Relay Resistance Transmission Time Mismatch Limit	'32'
Terminal Action Code – Default	'840000000C'
Terminal Action Code – Denial	'840000000C'
Terminal Action Code – Online	'840000000C'
Terminal Capabilities	'000000'
Terminal Country Code	'0000'
Terminal Transmission Time For Relay Resistance Command	'0012'

Data Object	**Default Value**
Terminal Transmission Time For Relay Resistance Response	'0018'
Terminal Type	'00'

Table 2: Configuration Data

The Terminal’s Public Keys of the Payment Scheme(s) which use the CPACE Terminal Kernel shall be available to the Kernel for the RIDs of the AIDs processed by the Kernel.

6.1.2 Transaction Data

On Kernel activation the Entry Point shall provide the Kernel with the Data Objects included in Table 3.

Data Object	**Presence**
Amount, Authorised	Mandatory
Amount, Other	Optional
FCI of the Selected Application	Mandatory
Transaction Date	Mandatory
Transaction Currency Code	Mandatory
Transaction Currency Exponent	Mandatory
Transaction Time	Mandatory
Transaction Type	Optional If not provided by the Terminal the default value ‘00’ shall be used by the Kernel

Table 3: Transaction Data

6.2 Output Data

6.2.1 Outcome Record

Once the Kernel finishes processing, it returns the control to the Entry Point sending the Outcome Parameters described in Table 4.

Parameters
Outcome
Start
Online Response Data
CVM
UI Request on Outcome Present
UI Request on Restart Present
Data Record Present
Discretionary Data Present
Alternate Interface Preference
Receipt
Field Off Request
Removal Timeout

Table 4: Outcome Parameters

6.2.2 Outcome Data Record

For the Outcomes Approved, Declined and Online Request the Kernel returns to the Entry Point a list of TLV encoded transaction's Data Objects for the terminal to perform the Online (for Online Request) and the Completion functions. The Data Objects included in the Data Record are described in Table 5.

Tag	Length	Name	Presence	Source
'9F26'	8	*Application Cryptogram*	M	Card
'5F24'	6	*Application Expiration Date*	M	Card
'9F42'	2	*Application Currency Code*	C[2]	Card
'5F25'	6	*Application Effective Date*	C[2]	Card
'82'	2	*Application Interchange Profile (AIP)*	M	Card
'50'	1-16	*Application Label*	C[2]	Card
'5A'	16	*Application PAN*	M	Card
'5F34'	2	*Application PAN Sequence Number*	C[2]	Card
'9F12'	1-16	*Application Preferred Name*	C[2]	Card
'9F36'	2	*Application Transaction Counter (ATC)*	M	Card
'9F07'	2	*Application Usage Control*	C[2]	Card
'5F20'	2-26	*Cardholder Name*	C[2]	Card

[2] Conditional – If provided by the Card.

Tag	Length	Name	Presence	Source
‘8E’	10-252	*Cardholder Verification Method (CVM) List*	C^2	Card
‘9F34’	3	*Cardholder Verification Method (CVM) Results*	M	Kernel
‘9F27’	1	*Cryptogram Information Data*	M	Card
'84'	5-16	*DF Name*	M	Card
‘5F53’	10-34	*International Bank Account Number (IBAN)*	C^2	Card
‘9F0D’	5	*Issuer Action Code – Default*	C^2	Card
‘9F0E’	5	*Issuer Action Code – Denial*	C^2	Card
‘9F0F’	5	*Issuer Action Code – Online*	C^2	Card
‘9F10’	32	*Issuer Application Data*	C^2	Card
'9F11'	1	*Issuer Code Table Index*	C^2	Card
‘5F28’	2	*Issuer Country Code*	C^2	Card
‘9F24’	29	*Payment Account Reference (PAR)*	C^2	Card
'9F33'	3	*Terminal Capabilities*	M	Kernel
‘95’	5	*Terminal Verification Results (TVR)*	M	Kernel
‘9F6E'	5-32	*Third Party Data*	C^2	Card
‘57’	19	*Track 2 Equivalent Data*	C^2	Card
‘9B’	2	*Transaction Status Information*	M	Kernel
‘9F37’	4	*Unpredictable Number*	M	Kernel

Table 5: Outcome Data Record

7 Commands for Financial Transaction

From the list of commands in section 6.5 of EMV Book 3, only the following commands are supported by the CPACE Terminal Kernel:

- Generate AC
- Get Processing Options
- Read Record

An additional command has to be supported which is described in section 7.1. and the Generate AC command is supported with the modification described in section 7.2

7.1 EXCHANGE RELAY RESISTANCE DATA Command

7.1.1 Definition and Scope

The EXCHANGE RELAY RESISTANCE DATA Command is used to exchange data, between the kernel and the terminal, needed to protect against Relay Attacks.

7.1.2 EXCHANGE RELAY RESISTANCE DATA Command Coding

The EXCHANGE RELAY RESISTANCE DATA command message is coded as shown in Table 6.

Code	Value
CLA	'80'
INS	'EA'
P1	'00'
P2	'00'
Lc	'04'
Data	*Terminal Relay Resistance Entropy*
Le	'00'

Table 6: EXCHANGE RELAY RESISTANCE DATA Command Message

7.1.3 Data Field Returned in the Response Message

The EXCHANGE RELAY RESISTANCE DATA response data field is coded as shown in Table 7.

Position	**Value**	**Length (in bytes)**	**Format**
Byte 1	'80'	1	b
Byte 2	'0A'	1	b
Bytes 5 - 8	*Device Relay Resistance Entropy*	4	b
Bytes 9 - 10	*Min Time For Processing Relay Resistance APDU*	2	b
Bytes 11 - 12	*Max Time For Processing Relay Resistance APDU*	2	b
Bytes 13 - 14	*Device Estimated Transmission Time For Relay Resistance R-APDU*	2	b

Table 7: EXCHANGE RELAY RESISTANCE DATA Response Message Data Field

7.2 Extension to Generate AC Command

The response message of Generate AC Command is described in Table 8 when no CDA is performed and in Table 9 when CDA is performed.

Tag	**Value**		**Presence**
'77'	*Response Message Template Format 2*		M
	'9F27'	*Cryptogram Information Data (CID)*	M
	'9F36'	*Application Transaction Counter (ATC)*	M
	'9F26'	*Application Cryptogram (AC)*	M
	'9F10'	*Issuer Application Data (IAD)*	M
	'DF4B'	*Cardholder Verification and Confirmation Status (CHV&CS)*	O

Table 8: GENERATE AC Response Message Data Field – No CDA

Tag	**Value**		**Presence**
'77'	*Response Message Template Format 2*		M
	'9F27'	*Cryptogram Information Data (CID)*	M
	'9F36'	*Application Transaction Counter (ATC)*	M
	'9F4B'	*Signed Dynamic Application Data (SDAD)*	M
	'9F10'	*Issuer Application Data (IAD)*	M
	'DF4B'	*Cardholder Verification and Confirmation Status (CHV&CS)*	O

Table 9: GENERATE AC Response Message Data Field – CDA

8 Kernel Activation

After the Final Combination Selection, the Entry Point activates the kernel. On Kernel Activation the Entry Point shall provide to the Kernel the *FCI Template* returned in the response to SELECT and the transaction data defined in 6.1.2.

Once activated the Kernel shall parse the *FCI Template* (according to Annex B of [EMV Book 3]), and check that the length of all known data objects comply with the length specified in the data dictionary.

If **any** of the following is true:

- the parsing of *FCI Template* fails
- **or** the *DF Name* is missing in the *FCI Template*

Then the kernel shall:

- Return the Control to Entry Point with an Outcome “Select Next” as in Section 22.2.10.

9 Initiate Application Processing

Initiate Application Processing shall be performed as described in [EMV Book 3] Section 10.1 with the following modification and additions:

If the status words ~~'6985' are~~ returned are different from ‘9000', the ~~terminal~~ Kernel shall ~~eliminate the current application from consideration and return to the Application Selection function to select another application~~ return the Control to Entry Point with an Outcome “Select Next” as in Section 22.2.10.

If **any** of the following is true:

- *Application Interchange Profile* is missing in the GPO response
- **or** *Application File Locator* is missing in the GPO response
- **or** ‘EMV mode is supported’ (byte 2, bit 8) in *Application Interchange Profile* has the value 0b

Then the kernel shall:

- Return the Control to Entry Point with an Outcome “End Application (Other Card)” as in Section 22.2.6

If **any** the following is true:

- *Amount, Authorized* is missing
- **or** *Amount, Authorized* is empty (Length = 0)
- **or** *Transaction Currency Code* is missing
- **or** *Transaction Currency Code* is empty (Length = 0)

Then kernel shall:

- Return the Control to Entry Point with an Outcome “End Application (no restart)” as in Section 22.2.7

If **all** of the following are true:

- ‘CDCVM is Supported' (byte 1, bit 2) in *Application Interchange Profile* has the value 1b
- **and** ‘CDCVM is Supported’ (bit 6) in *Kernel Configuration* has the value 1b
- **and** *Amount, Authorized* > *Contactless Transaction Limit with CDCVM*

Then the kernel shall:

- Return the Control to Entry Point with an Outcome “Select Next” as in Section 22.2.10

Else

- If the following is true:
    - o *Amount, Authorized* > *Contactless Transaction Limit without CDCVM*

Then the kernel shall:

- Return the Control to Entry Point with an Outcome “Select Next” as in Section 22.2.10.

Relay Resistance Counter := 1

10 Relay Resistance Protocol

If **any** of the following is true:

- 'Relay Resistance Protocol Support' (byte 2, bit 1) in *Application Interchange Profile* as the value 0b;
- **or** 'Relay Resistance Protocol Support' (bit 5) in *Kernel Configuration* as the value 0b

Then the Kernel shall:

- Set 'Relay Resistance Protocol performed' (byte 5, [b1:2]) in *Terminal Verification Results (TVR)* := 01b.
- Skip the Relay Resistance Protocol defined below and go to the Read Application Data function as specified in Section 11.

Else the Kernel shall:

- Perform the Relay Resistance Protocol as follows in this Section

Generate a 4 byte random number and store it in the *Terminal Relay Resistance Entropy*.

Prepare the EXCHANGE RELAY RESISTANCE DATA command message as defined in Table 6.

Start a Timer to measure the execution time of the EXCHANGE RELAY RESISTANCE DATA command. The timer shall measure the time in hundreds of microseconds.

Send the EXCHANGE RELAY RESISTANCE DATA command to the card.

As soon as the EXCHANGE RELAY RESISTANCE DATA command response is received stop the Timer used to measure the execution time of the EXCHANGE RELAY RESISTANCE DATA command.

If the following is true:

- the status words returned are different from '9000'

Then the Kernel shall:

- Return the Control to Entry Point with an Outcome "End Application (Other Card)" as in Section 22.2.6

Parse the EXCHANGE RELAY RESISTANCE DATA command response message data field according to Section 7.1.3.

If the parsing fails then the Kernel shall:

- Return the Control to Entry Point with an Outcome "End Application (Other Card)" as in Section 22.2.6

Store in

- *Device Relay Resistance Entropy*,
- *Min Time For Processing Relay Resistance APDU*,
- *Max Time For Processing Relay Resistance APDU*
- and *Device Estimated Transmission Time For Relay Resistance R-APDU*

the data retrieved from the EXCHANGE RELAY RESISTANCE DATA command response message data field according to Section 7.1.3.

If the following is true:

- *Device Estimated Transmission Time For Relay Resistance R-APDU > Terminal Transmission Time For Relay Resistance Response*

Then:

- *Expected Min Transmission Time For RR Response* := *Terminal Transmission Time For Relay Resistance Response*

Else:

- *Expected Min Transmission Time For RR Response* := *Device Estimated Transmission Time For Relay Resistance R-APDU*

If the following is true:

- *Timer value > (Terminal Transmission Time For Relay Resistance Command + Expected Min Transmission Time For RR Response)*

Then:

- *Measured Relay Resistance Time := Timer value – Terminal Transmission Time For Relay Resistance Command – Expected Min Transmission Time For RR Response*

Else:

- *Measured Relay Resistance Time := 0*

If the following is true:

- *Min Time For Processing Relay Resistance APDU > Min Time Relay Resistance Tolerance*

Then:

- If the following is true:
  - *Measured Relay Resistance Time <* (*Min Time For Processing Relay Resistance APDU – Min Time Relay Resistance Tolerance*)
- Then the Kernel shall:
  - Return the Control to Entry Point with an Outcome “End Application (Other Card)” as in Section 22.2.6

If **all** the following are true:

- *Relay Resistance Counter* < 2
- **and** *Measured Relay Resistance Time* > *Max Time For Processing Relay Resistance APDU* + *Max Time Relay Resistance Tolerance*

Then the Kernel shall:

- Increment *Relay Resistance Counter*
- Repeat the Relay Resistance Protocol from the beginning of this Section.

Else the Kernel shall:

- Continue processing as follows in this Section

If the following is true:

- *Measured Relay Resistance Time* > (*Max Time For Processing Relay Resistance APDU* + *Max Time Relay Resistance Tolerance*)

Then:

- ‘Relay resistance time limits exceeded’ in *Terminal Verification Results (TVR)* := 1b

If **any** of the following is true:

- *Terminal Transmission Time For Relay Resistance Response* = 0
- **or** *Device Estimated Transmission Time For Relay Resistance R-APDU* = 0
- **or** *Measured Relay Resistance Time* < *Min Time For Processing Relay Resistance APDU*

Then:

- ‘Relay resistance threshold exceeded' in *Terminal Verification Results (TVR)* := 1b

Else:

If **any** the following is true:

- (*Device Estimated Transmission Time For Relay* Resistance *R-APDU* * 100 div *Terminal Transmission Time For Relay Resistance Response*) < *Relay Resistance Transmission Time Mismatch Limit*
- **or** (*Terminal* Transmission *Time For Relay Resistance Response* *100 div *Device Estimated Transmission Time For Relay Resistance R-APDU*) < *Relay Resistance Transmission Time Mismatch Limit*
- **or** *Measured Relay Resistance Time* - *Min Time For Processing Relay Resistance APDU* > *Relay Resistance Min Time Difference Limit*

Then:

- ‘Relay resistance threshold exceeded' in *Terminal Verification Results (TVR)* := 1b

'Relay resistance performed' in in *Terminal Verification Results (TVR)* := 10b

11 Read Application Data

Read Application Data shall be performed as described in [EMV Book 3] Section 10.2.

12 Offline Data Authentication

Offline Data Authentication shall be performed as described in [EMV Book 3] Section 10.3 and [EMV Book 2] section 6 with the following modification.

12.1 Modifications in [EMV Book 3] Section 10.3

The terminal supports SDA shall always evaluate as “false”, considering that the CPACE Terminal Kernel shall not support SDA, regardless of *Terminal Capabilities*.

The terminal supports DDA shall always evaluate as “false”, considering that the CPACE Terminal Kernel shall not support DDA, regardless of *Terminal Capabilities*.

12.2 Modifications in [EMV Book 2] Section 6.6

[…]

When the GENERATE AC command is issued with a CDA request, then if any of the above errors are detected subsequently, the eventual result will be an offline decline in accordance with the paragraphs beginning "If CDA fails in conjunction" in Book 4 Section 6.3.2.

In sections 6.1.1 and 6.6.2 it is assumed that:

- Both the ICC and the terminal support CDA.
- ~~The cryptogram to be requested is not an Application Authentication Cryptogram (AAC), i.e. Terminal Action Analysis has not resulted in offline decline.~~
- The TVR bit for 'CDA failed' is not set to 1 prior to final Terminal Action Analysis.
- Except when returning an AAC, where the return of a CDA signature depends on the ICC application capabilities (see *Device Application Capabilities*), the ICC always replies with a CDA signature when requested by the terminal.

In the case of the first GENERATE AC command:

- When requesting an ARQC, the terminal may request it with or without a CDA signature. When an ARQC is requested without a CDA signature, then the terminal shall set the TVR bit for 'Offline data authentication was not performed' to 1[24] prior to issuance of the GENERATE AC command. When an ARQC is requested without a CDA signature, the processes described in sections 6.6.1 and 6.6.2 are not performed.
- When requesting a TC, the terminal shall request it with a CDA signature.
- When requesting an AAC, the terminal shall request it without a CDA signature if *Device Application Capabilities* is missing in the ICC or ‘CDA Support on AAC’ (byte 2, bit 1) in *Device Application Capabilities* has the value 0b, and shall request it with a CDA signature if ‘CDA Support on AAC’ (byte 2, bit 1) in *Device Application Capabilities* has the value 1b.

6.6.1 Dynamic Signature Generation

The generation of the combined dynamic signature and Application Cryptogram takes place in the following steps.

1. The terminal issues a first or second GENERATE AC command with the 'CDA signature requested' bit in the GENERATE AC command set to 1 according to sections 6.5.5.4 and 9.3 of Book 3.
2. If the ICC is to respond with a TC or ARQC or AAC and the AAC was requested by the terminal (if *Device Application Capabilities* is present in the ICC and 'CDA Support on AAC' in *Device Application Capabilities* has the value 1b), the ICC performs the following steps:
    a. The ICC generates the TC or ARQC or AAC if requested by the terminal.
    b. The ICC applies the hash algorithm specified by the Hash Algorithm Indicator to the concatenation from left to right of the following data elements:

[...]

3. If the terminal did not request an AAC, or CDA on AAC is not supported by the ICC, and the ICC responds with an AAC, the ICC response shall be coded according to either format 1 or format 2 as specified in section 6.5.5.4 of Book 3 and shall contain at least the mandatory data elements specified in Table 21, and optionally the Issuer Application Data.

[...]

6.6.3 Sample CDA Flow

The figures on the following pages are an example of how a terminal might perform CDA. This sample flow provides a generalised illustration of the concepts of CDA. It does not necessarily contain all required steps and does not show parallel processing (for example, overlapping certificate recovery and signature generation). If any discrepancies are found between the text and flow, the text shall be followed.

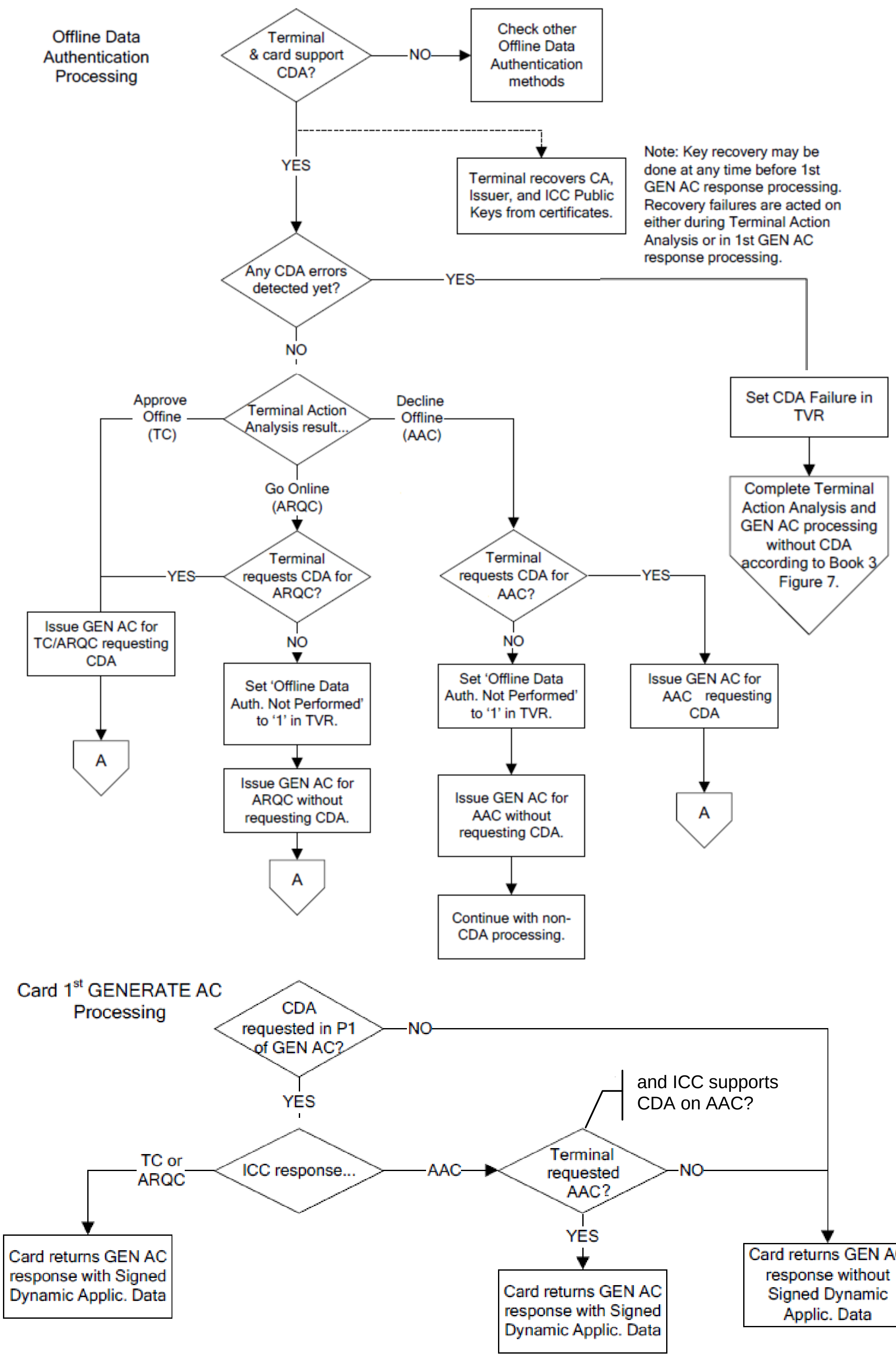
Offline Data Authentication Processing
Terminal & card support CDA?
NO
Check other Offline Data Authentication methods
YES
Terminal recovers CA, Issuer, and ICC Public Keys from certificates.
Note: Key recovery may be done at any time before 1st GEN AC response processing. Recovery failures are acted on either during Terminal Action Analysis or in 1st GEN AC response processing.
Any CDA errors detected yet?
YES
Set CDA Failure in TVR
Complete Terminal Action Analysis and GEN AC processing without CDA according to Book 3 Figure 7.
NO
Terminal Action Analysis result...
Approve Offline (TC)
Decline Offline (AAC)
Go Online (ARQC)
Terminal requests CDA for ARQC?
YES
Issue GEN AC for TC/ARQC requesting CDA
A
NO
Set 'Offline Data Auth. Not Performed' to '1' in TVR.
Issue GEN AC for ARQC without requesting CDA.
A
Terminal requests CDA for AAC?
YES
Issue GEN AC for AAC requesting CDA
A
NO
Set 'Offline Data Auth. Not Performed' to '1' in TVR.
Issue GEN AC for AAC without requesting CDA.
Continue with non-CDA processing.
Card 1st GENERATE AC Processing
CDA requested in P1 of GEN AC?
NO
YES
ICC response...
TC or ARQC
Card returns GEN AC response with Signed Dynamic Applic. Data
AAC
Terminal requested AAC?
and ICC supports CDA on AAC?
NO
YES
Card returns GEN AC response with Signed Dynamic Applic. Data
Card returns GEN AC response without Signed Dynamic Applic. Data

Figure 3: CDA Sample Flow Part 1 of 3

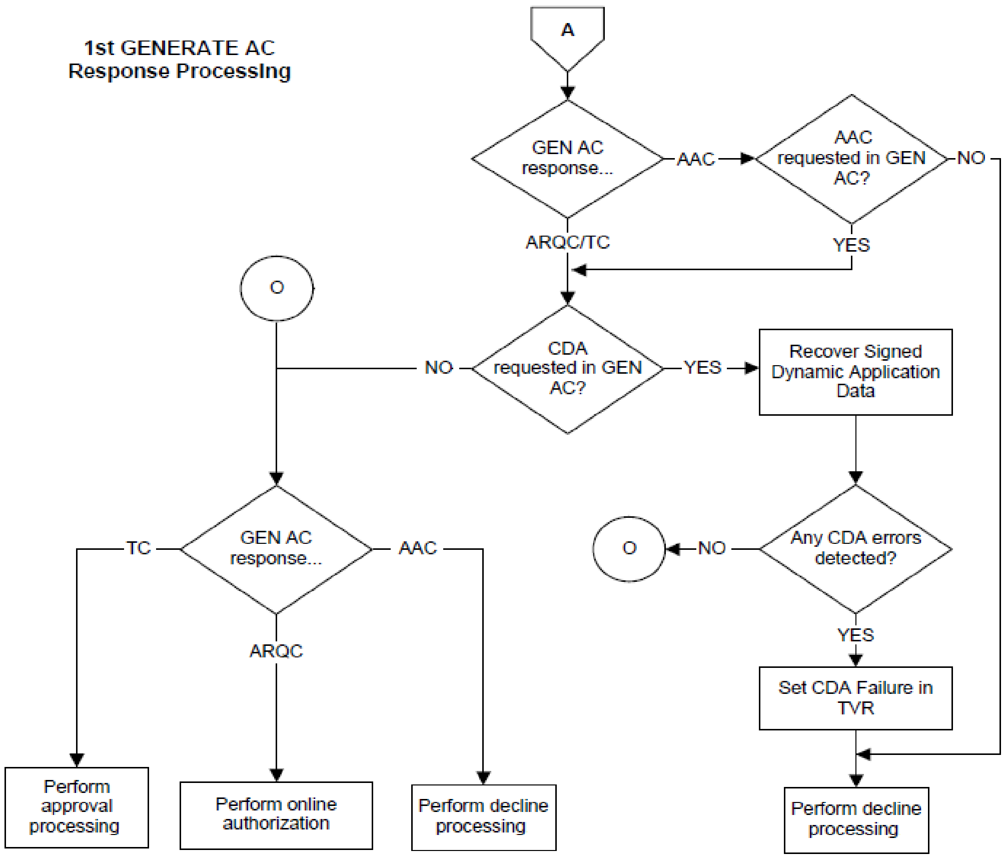

Figure 4: CDA Sample Flow Part 2 of 3

13 Processing Restrictions

Processing Restrictions shall be performed as described in [EMV Book 3] Section 10.4.

14 Cardholder Verification

The Cardholder Verification shall be performed as follows:

If the following is true:

- *Amount, Authorized* > *Reader CVM Required Limit*

Then:

- ‘CVM Capability’ in Terminal Capabilities := *CVM Capabilities (above CVM Limit)*

Else:

- ‘CVM Capability’ in Terminal Capabilities := *CVM Capabilities (below or equal CVM Limit)*

If **all** the following are true:

- ‘CDCVM is Supported’ (byte 1, bit 2) in *Application Interchange Profile* has the value 1b
- **and** ‘CDCVM is Supported’ (bit 6) in *Kernel Configuration* has the value 1b

Then:

- 'CVM Condition' (byte 2) in *CVM Results* := '00'
- 'CVM Result' (byte 3) in *CVM Results* := '02' (successful)
- If the following is true:
    - *Amount, Authorized* > Reader *CVM Required Limit*
- Then:
    - 'CVM Performed' (byte 1) in *CVM Results* := '01' (Plaintext offline PIN verification performed)

  Else:
    - 'CVM Performed' (byte 1) in *CVM Results* := '3F' (No CVM performed)

Else:

- Perform Cardholder Verification as described in [EMV Book 4] Section 6.3.4.5 and [EMV Book 3] Section 10.5, with the following modifications:
    - In section “10.5.1 Offline PIN Processing”, replace all content with the following:
        - ‘CVM Result’ (byte 3) in *CVM Results* := ‘00’ (Unknown)
        - Cardholder verification is considered successful and complete
    - In section “10.5.2 Online PIN Processing”, replace all content with the following:
        - ‘Online PIN Entered’ (byte 3, bit 3) in *Terminal Verification Results (TVR)* := 1b

- ‘CVM Result’ (byte 3) in *CVM Results* := ‘00’ (Unknown)
- Cardholder verification is considered successful and complete

- In Section “10.5.5 CVM Processing Logic”, replace the Figure 10: CVM Processing (Part 3 of 5) with the figure bellow.

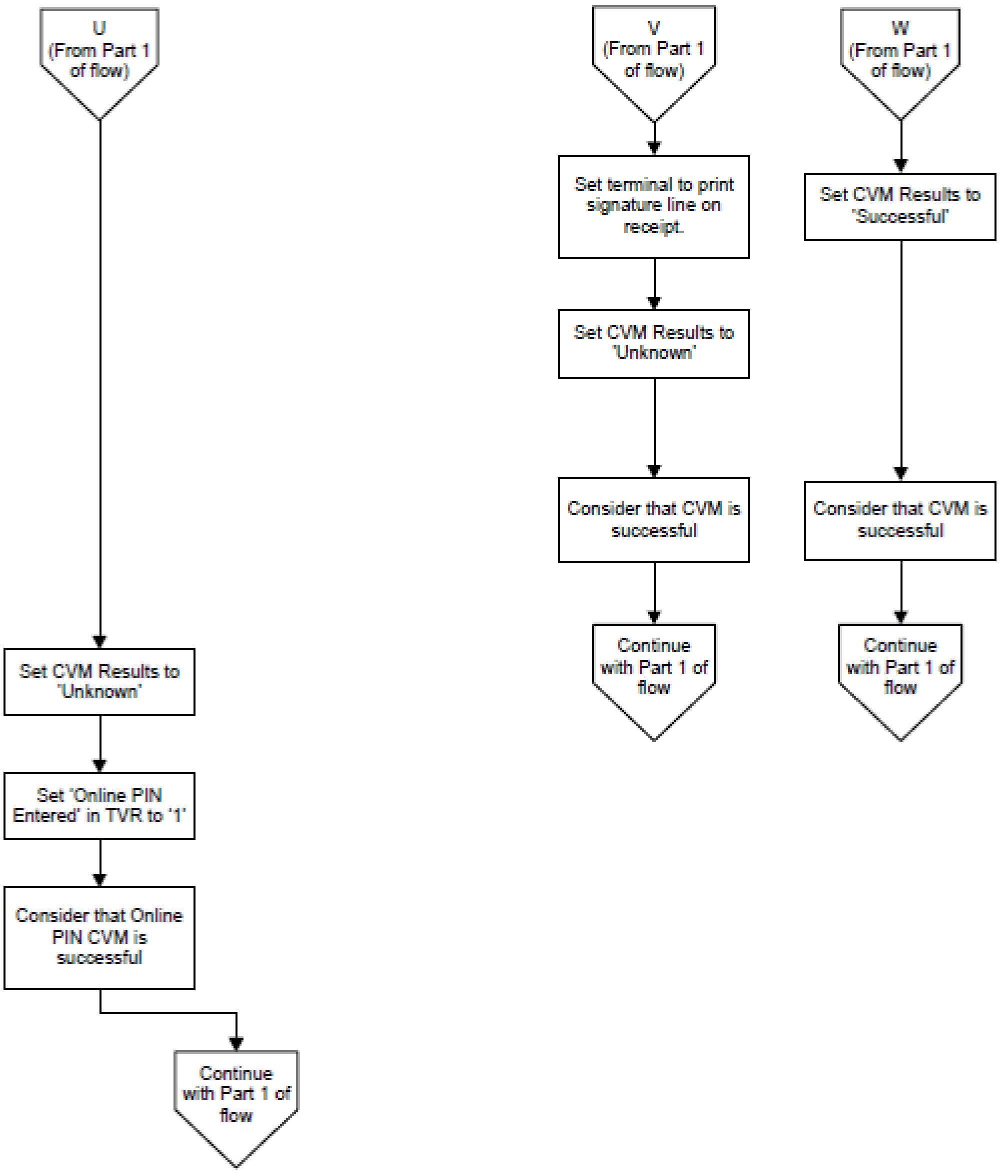
U (From Part 1 of flow)
Set CVM Results to 'Unknown'
Set 'Online PIN Entered' in TVR to '1'
Consider that Online PIN CVM is successful
Continue with Part 1 of flow
V (From Part 1 of flow)
Set terminal to print signature line on receipt.
Set CVM Results to 'Unknown'
Consider that CVM is successful
Continue with Part 1 of flow
W (From Part 1 of flow)
Set CVM Results to 'Successful'
Consider that CVM is successful
Continue with Part 1 of flow

Figure 10: CVM Processing (Part 3 of 5)

In Section “10.5.5 CVM Processing Logic”, replace the Figure 11: CVM Processing (Part 4 of 5) with the figure bellow.

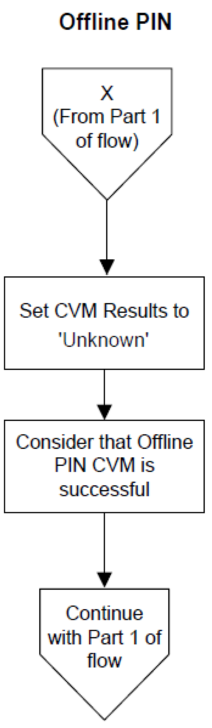
Offline PIN
X (From Part 1 of flow)
Set CVM Results to 'Unknown'
Consider that Offline PIN CVM is successful
Continue with Part 1 of flow

Figure 11: CVM Processing (Part 4 of 5)

15 Terminal Risk Management

Terminal Risk Management shall be performed as follows.

15.1 Floor Limits Check

The kernel shall perform Floor Limits check as follows.

If the following is true:

- *Amount, Authorized* > *Reader Contactless Floor Limit*

Then:

- ‘Transaction exceeds floor limit’ (byte 4, bit 8) in *Terminal Verification Results (TVR)* := 1b

15.2 Random Transaction Selection

Random Transaction Selection is not supported by CPACE Terminal Kernel.

15.3 Velocity Checking

Velocity Checking is not supported by the CPACE Terminal Kernel.

16 Terminal Action Analysis

Terminal Action Analysis shall be performed by Kernel as described in [EMV Book 3] Section 10.7.

17 Card Action Analysis

Card Action Analysis shall be performed as described hereafter.

The card action analysis process is performed when the terminal issues the GENERATE AC command for a given transaction. The type of requested cryptogram (AAC, TC or ARQC) depends on the Terminal Action Analysis performed by the kernel.

The Unpredictable Number sent in the first GENERATE AC command data must be the same as the Terminal Relay Resistance Entropy sent in the EXCHANGE RELAY RESISTANCE DATA command data.

During the execution of the GENERATE AC command the card performs the card risk management and the decision is made known to the terminal by returning a TC, an ARQC, or an AAC in response to a GENERATE AC command, as described in [EMV Book 3] section 6.5.5.

If **all** the following are true:

- SW1 SW2 = ‘9000’ is returned in the response to GENERATE AC
- **and** *Cryptogram Information Data* is returned in the response to GENERATE AC
- **and** *Application Transaction Counter* is returned in the response to GENERATE AC
- **and** *Issuer Application Data* is returned in the response to GENERATE AC
- **and any** of the following is true:
  - an AAC is returned
  - **or** a TC is returned and the Kernel requested a TC
  - **or** an ARQC is returned and the Kernel requested a TC or ARQC

Then the Kernel shall:

- Immediately send a User Interface (UI) Request Message (according to [EMV Book A] Section 5.8.3), with the parameters settings defined in Table 10.

Parameters	**Settings**
UI Request Message	• **Message Identifier** := '1E' (Clear Display) • **Status** := Card Read Successfully • **Hold Time** := ‘0000’ • **Language Preference** (Tag '5F2D'): If returned by the card during Application Selection

Table 10: Card Reading OK User Interface Request Message

Else:

- Return the Control to Entry Point with an Outcome “End Application (other card)” as in Section 22.2.6.

If the following is true:

- *Signed Dynamic Application Data* is returned in the response to GENERATE AC

Then:

- Verify the *Signed Dynamic Application Data* as in sections 6.6 of [EMV Book 2]
- If **any** the following is true:
    - 'CDA failed' (byte 1, bit 3) in *Terminal Verification Results (TVR)*
    - **or all** the following are true:
        - ‘Relay Resistance Protocol Support’ (byte 2, bit 1) in *Application Interchange Profile* has the value 0b
        - **and** the length of ICC Dynamic Data is less than 30 + Length of *ICC Dynamic Number*
    - **or all** the following are true:
        - ‘Relay Resistance Protocol Support’ (bit 5) in *Kernel Configuration* has the value 0b
        - **and** the length of ICC Dynamic Data is less than 30 + Length of *ICC Dynamic Number*
    - **or all** the following are true:
        - ‘Relay Resistance Protocol Support’ (byte 2, bit 1) in *Application Interchange Profile* has the value 1b
        - **and** ‘Relay Resistance Protocol Support’ (bit 5) in *Kernel Configuration* has the value 1b
        - **and** the length of ICC Dynamic Data is less than 44 + Length of *ICC Dynamic Number*
- Then the kernel shall:
    - Return the Control to Entry Point with an Outcome “End Application (other card)” as in Section 22.2.6.
- If **all** the following are true:
    - ‘Relay Resistance Protocol Support’ (byte 2, bit 1) in *Application Interchange Profile* has the value 1b
    - **and** ‘Relay Resistance Protocol Support’ (bit 5) in *Kernel Configuration* has the value 1b

- Then:
    - Consider that the ICC Dynamic Data recovered is as shown in Table 11 and use the Relay Resistance Protocol Data Elements in the ICC Dynamic Data (ICC DD) to compare with the Relay Resistance Protocol Data Elements obtained in Section 10.
    - If **any** of the following is true:
        - *Terminal Relay Resistance Entropy* <> *Terminal Relay Resistance Entropy* (ICC DD)
        - **or** *Device Relay Resistance Entropy* <> *Device Relay Resistance Entropy* (ICC DD)
        - **or** *Min Time For Processing Relay Resistance APDU* <> *Min Time For Processing Relay Resistance APDU* (ICC DD)
        - **or** *Max Time For Processing Relay Resistance APDU* <> *Max Time For Processing Relay Resistance APDU* (ICC DD)
        - **or** *Device Estimated Transmission Time For Relay Resistance R-APDU* <> Device Estimated Transmission Time For Relay Resistance R-APDU (ICC DD).
    - Then the kernel shall:
        - Return the Control to Entry Point with an Outcome “End Application (other card)” as in Section 22.2.6.

Field Name	**Length (in bytes)**	**Value**
ICC Dynamic Data	1	*ICC Dynamic Number Length*
	2-8	*ICC Dynamic Number*
	1	*Cryptogram Information Data*
	8	*Application Cryptogram*
	20	Transaction Data Hash Code
	4	*Terminal Relay Resistance Entropy*
	4	*Device Relay Resistance Entropy*
	2	*Min Time For Processing Relay Resistance APDU*
	2	*Max Time For Processing Relay Resistance APDU*
	2	*Device Estimated Transmission Time For Relay Resistance R-APDU*

Table 11: ICC Dynamic Data Including RRP Data

Else (Signed Dynamic Application Data is not returned in the response to GENERATE AC):

- If **any** of the following is true:
    - The *Application Cryptogram* is not returned in the response to GENERATE AC
    - **or all** the following are true:
        - CDA was required
        - **and** a TC or ARQC is returned
    - **or all** the following are true:
        - CDA was required
        - **and** an AAC is returned
        - **and** the kernel requested an AAC
- Then the kernel shall:
    - Return the Control to Entry Point with an Outcome “End Application (other card)” as in Section 22.2.6.

If **all** the following are true:

- The *Cardholder Verification and Confirmation Status (CHV&CS)* is returned in the response to GENERATE AC
- **and** (*Cardholder Verification and Confirmation Status (CHV&CS)* AND ‘00030F’) <> ‘000000’

Then the kernel shall:

- Return the Control to Entry Point with an outcome “End Application (2nd Tap)" as in Section 22.2.5.

Set ‘Card risk management was performed’ bit in the *TSI* to 1b.

If a TC is returned then the Kernel shall:

- Return the Control to Entry Point with an outcome “Approved”, as in Section 22.2.1.

If an ARQC is returned then the Kernel shall:

- Return the Control to Entry Point with an outcome “Online Request”, as in Section 22.2.3.

If an AAC is returned then:

- If **any** of the following is true:
    - *Transaction Type* = '01' (Cash Withdrawal)
    - **or** *Transaction Type* = '17' (Cash Disbursement)
    - **or** *Transaction Type* = '00' (Payment)
    - **or** *Transaction Typ*e = '09' (Payment with Cashback)
- Then:
    - If **any** of the following is true:
        - **all** the following are true:
            - ✧ 'Unique Identifier' in *Third Party Data* AND '8000' = '0000'
            - ✧ **and** 'Device Type' in *Third Party Data* <> '3030'
        - **or** ‘IC with contacts' (byte 1, bit 6) in *Terminal Capabilities* has the value 0
    - Then:
        - Return the Control to Entry Point with an outcome “Declined”, as in Section 22.2.2.
    - Else:
        - Return the Control to Entry Point with an outcome “Try Another Interface”, as in Section 22.2.4
- Else:
    - Return the Control to Entry Point with an outcome “End Application (no restart)”, as in Section 22.2.7

18 Online Processing

The Online Request Outcome is sent to the Entry Point without a restart request (Start = N/A). No additional processing is performed in the kernel.

19 Issuer-to-Card Script Processing

The Online Request Outcome is sent to the Entry Point without a restart request (Start = N/A). No additional processing is performed in the kernel.

20 Completion

It is not the responsibility of the Kernel to perform the transaction Completion.

21 Error Handling

This section specifies the default procedures of the Kernel to handle the errors that may happen while processing a transaction, and that have not specific treatment elsewhere in the specification. Possible types of errors are:

- Communication errors;
- Processing errors;
- Erroneous or missing data.

21.1 Communication errors

This section describes how the Kernel shall behave when an error occurs during the communication with the card.

If, during a transaction, the following is true:

- A Transmission, Protocol, or Timeout error (as defined in [EMV Book D] is reported to the Kernel

Then:

- If the following is true:
  - The error occurred in the GET PROCESSING OPTIONS command
- Then the Kernel shall:
  - Return the Control to Entry Point with an outcome “Try Again”, as in Section 22.2.9
- Else the Kernel shall:
  - Return the Control to Entry Point with an outcome “End Application (with restart)”, as in Section 22.2.8.

21.2 Processing errors

This section describes how the Kernel shall behave when a processing error occurs.

If, during transaction processing, in response to any command, **all** the following are true:

- The card returns a value of SW1 SW2 that is different from '9000'
- **and** SW1 SW2 does not have a specific processing defined in EMV specifications or in this specification

Then the kernel shall:

- Return the Control to Entry Point with an Outcome “End Application (other card)” as in Section 22.2.6.

21.3 Erroneous or missing data

This section describes how the Kernel shall behave when the transaction is to be terminated due to erroneous or missing data.

If, during transaction processing, **any** of the following is true:

- the application of the rules defined in [EMV Book 3] Section 7.5, results in transaction termination
- **or** the *Application PAN* (tag ‘5A’) does not match the Primary Account Number contained in *Track 2 Equivalent data* (tag ‘57’), if present in the card
- **or** a mandatory data object is missing in a command response
- **or** a command response does not parse correctly
- **or** the transaction has to be terminated according to the EMV specification

Then the Kernel shall:

- Return the Control to Entry Point with a final outcome “End Application (other card)” as in Section 22.2.6.

While the CDOL 2 is mandatory according to [EMV Book 3] Section 7.2, the presence of Data Element ‘8D’ *CDOL 2* is not required according to this specification, since CPACE Terminal Kernel does not support 2nd GENERATE AC.

22 Transaction Outcomes

This section details the CPACE Terminal Kernel Outcomes.

22.1 Introduction

After processing a transaction, the Kernel returns control to the Entry Point, passing the Outcome that defines how the transaction processing shall be continued. The Entry Point shall restart the Kernel or transfer the control to the Terminal, that will proceed depending on the outcome.

22.2 Outcomes for CPACE Terminal Kernel

Outcomes for CPACE Terminal Kernel are according to [EMV Book A] Section 6.1:

Final Outcomes:

- Approved;
- Declined;
- Online Request;
- Try Another Interface;
- End Application (2nd Tap);
- End Application (other card);
- End Application (no restart);
- End Application (with restart).

Non final Outcomes:

- Try Again;
- Select Next.

Outcome Parameters are according to [EMV Book A] Section 6.2.

22.2.1 Approved

This section details the parameters' settings for Approved outcome.

Parameters	Settings
Outcome	Approved
Start	N/A
Online Response Data	N/A
CVM	According to the 'CVM Performed' (byte 1) and 'CVM Result' (byte 3) in *CVM Results*, as determined by Cardholder Verification in Section 14, as follows: **'CVM Performed' AND '3F'** | **CVM Result** | **CVM Outcome** '02' | '00' | Online PIN '01' | '02' | Confirmation Code Verified '1E' | '00' | Obtain Signature '01' or '03' or '04' or '05' | '00' | N/A Other values not defined above | No CVM
UI Request on Outcome Present	Yes • **Message Identifier** := ▪ '1A' ("Approved Please sign") if 'CVM Performed' (byte 1) in *CVM Results* AND '3F' = '1E' ▪ '03' ("Approved") if 'CVM Performed' (byte 1) in *CVM Results* AND '3F' <> '1E' • **Status** := Not ready (Led lights switched off) • **Hold Time** := *Message Hold Time* in Configuration Data • **Language Preference** (Tag '5F2D'): If returned by the card during Application Selection
UI Request on Restart Present	No
Data Record Present	Yes According to Table 5.
Discretionary Data Present	No
Alternate Interface Preference	N/A
Receipt	N/A
Field Off Request	N/A
Removal Timeout	'00'

Table 12: Approved outcome parameters

22.2.2 Declined

This section details the parameters' settings for "Declined" outcome.

Parameters	**Settings**
Outcome	Declined
Start	N/A
Online Response Data	N/A
CVM	N/A
UI Request on Outcome Present	Yes • **Message Identifier** := '07' ("Not Authorised") • **Status** := Not ready (Led lights switched off) • **Hold Time** := *Message Hold Time* in Configuration Data • **Language Preference** (Tag '5F2D'): If returned by the card during Application Selection
UI Request on Restart Present	No
Data Record Present	Yes According to Table 5
Discretionary Data Present	No
Alternate Interface Preference	N/A
Receipt	N/A
Field Off Request	N/A
Removal Timeout	'00'

Table 13: Declined outcome parameters

22.2.3 Online Request

This section details the parameters settings for Online Request outcome.

Parameters	**Settings**
Outcome	Online Request
Start	N/A
Online Response Data	N/A
CVM	According to the ‘CVM Performed’ (byte 1) and ‘CVM Result’ (byte 3) in *CVM Results*, as determined by Cardholder Verification in Section 14, as follows:

‘CVM Performed’ AND '3F'	**CVM Result**	**CVM Outcome**
‘02’	‘00’	Online PIN
‘01’	‘02’	Confirmation Code Verified
‘1E’	‘00’	Obtain Signature
‘01’ or ‘03’ or ‘04’ or ‘05’	‘00’	N/A
Other values not defined above		No CVM

UI Request on Outcome Present	Yes • **Message Identifier** := ▪ ‘09’ (“Please enter your PIN”) if ‘CVM Performed’ (byte 1) in *CVM Results* AND '3F' = '02' ▪ '1B' (“Authorising, Please wait”) if ‘CVM Performed’ (byte 1) in *CVM Results* AND '3F' = '1E' • **Status** := Not ready (Led lights switched off) • **Hold Time:** ‘0000’ • **Language Preference** (Tag '5F2D'): If returned by the card during Application Selection
UI Request on Restart Present	No
Data Record Present	Yes According to Table 5.
Discretionary Data Present	No
Alternate Interface Preference	N/A
Receipt	N/A
Field Off Request	N/A
Removal Timeout	‘00’

Table 14: Online Request outcome parameters

22.2.4 Try Another Interface

This section details the parameters settings for Try Another Interface outcome.

Parameters	**Settings**
Outcome	Try Another Interface
Start	N/A
Online Response Data	N/A
CVM	N/A
UI Request on Outcome Present	Yes • **Message Identifier** := '1D' (“Please insert card”) • **Status** := Not ready (Led lights switched off) • **Hold Time** := *Message Hold Time* in Configuration Data • **Language Preference** (Tag '5F2D'): If returned by the card during Application Selection
UI Request on Restart Present	No
Data Record Present	No
Discretionary Data Present	No
Alternate Interface Preference	Contact Chip
Receipt	N/A
Field Off Request	N/A
Removal Timeout	‘00’

Table 15: Try Another Interface outcome parameters

22.2.5 End Application (2nd Tap)

This section details the parameters' settings for End Application outcome with restart when a 2nd Tap is needed to complete a payment with a mobile device where the CVM, or transaction confirmation, have to be performed on the device.

Parameters	**Settings**
Outcome	End Application
Start	B
Online Response Data	N/A
CVM	N/A
UI Request on Outcome Present	Yes • **Message Identifier** := 'Message' in the line of the *CHV&CS Message Table* where 'Bit in Cardholder Verification and Confirmation Status (CHV&CS)' AND *Cardholder Verification and Confirmation Status (CHV&CS)* <> '000000'. If no line meets the condition the value '07' – 'Not Authorised' shall be used. • **Status** := 'Status' in the line of the *CHV&CS Message Table* where 'Bit in Cardholder Verification and Confirmation Status (CHV&CS)' AND *Cardholder Verification and Confirmation Status (CHV&CS)*' <> '000000'. If no line meets the condition the value 'Not Ready' shall be used. • **Hold Time** := *Message Hold Time* in Configuration Data • **Language Preference** (Tag '5F2D'): If returned by the card during Application Selection
UI Request on Restart Present	Yes • **Message Identifier** := Same as above • **Status** := Ready to Read • **Hold Time** := '0000' • **Language Preference** (Tag '5F2D'): If returned by the card during Application Selection
Data Record Present	Yes According to Table 5
Discretionary Data Present	No
Alternate Interface Preference	N/A
Receipt	N/A
Field Off Request	'*Field Off Hold Time*' in Configuration Data
Removal Timeout	'00'

Table 16: End Application (2nd Tap) outcome parameters

22.2.6 End Application (other card)

This section details the parameters' settings for End Application outcome without restart when it is not possible to conclude the transaction with the current card.

Parameters	Settings
Outcome	End Application
Start	N/A
Online Response Data	N/A
CVM	N/A
UI Request on Outcome Present	Yes • **Message Identifier** := '1C' ("Insert, swipe or try another card") • **Status** := Not Ready (Led lights switched off) • **Hold Time** := *Message Hold Time* in Configuration Data • **Language Preference** (Tag '5F2D'): If returned by the card during Application Selection
UI Request on Restart Present	N/A
Data Record Present	No
Discretionary Data Present	No
Alternate Interface Preference	N/A
Receipt	N/A
Field Off Request	N/A
Removal Timeout	'00'

Table 17: End Application (other card) outcome parameters

22.2.7 End Application (no restart)

This section details the parameters' settings for End Application outcome without restart and no message is available to the user.

Parameters	**Settings**
Outcome	End Application
Start	N/A
Online Response Data	N/A
CVM	N/A
UI Request on Outcome Present	Yes • **Message Identifier** := '1E' (Clear display) • **Status** := Not Ready (Led lights switched off) • **Hold Time** := '0000' • **Language Preference** (Tag '5F2D'): If returned by the card during Application Selection
Data Record Present	No
Discretionary Data Present	No
Alternate Interface Preference	N/A
Receipt	N/A
Field Off Request	N/A
Removal Timeout	'00'

Table 18: End Application (no restart) outcome parameters

22.2.8 End Application (with restart)

This section details the parameters' settings for End Application outcome when a communication error occurred and the card has to be presented again.

Parameters	Settings
Outcome	End Application
Start	B
Online Response Data	N/A
CVM	N/A
UI Request on Outcome Present	No
UI Request on Restart Present	Yes • **Message Identifier** := '21' ("Present Card Again") • **Status** := Ready to Read (Led lights switched off) • **Hold Time** := '0000' • **Language Preference** (Tag '5F2D'): If returned by the card during Application Selection
Data Record Present	No
Discretionary Data Present	No
Alternate Interface Preference	N/A
Receipt	N/A
Field Off Request	N/A
Removal Timeout	'00'

Table 19: End Application (with restart) outcome parameters

22.2.9 Try Again

This section details the parameters' settings for Try Again outcome.

Parameters	Settings
Outcome	Try Again
Start	B
Online Response Data	N/A
CVM	N/A
UI Request on Outcome Present	No
UI Request on Restart Present	No
Data Record Present	No
Discretionary Data Present	No
Alternate Interface Preference	N/A
Receipt	N/A
Field Off Request	N/A
Removal Timeout	'00'

Table 20: Try Again outcome parameters

22.2.10 Select Next

This section details the parameters' settings for Select Next outcome.

Parameters	Settings
Outcome	Select Next
Start	C
Online Response Data	N/A
CVM	N/A
UI Request on Outcome Present	No
UI Request on Restart Present	No
Data Record Present	No
Discretionary Data Present	No
Alternate Interface Preference	N/A
Receipt	N/A
Field Off Request	N/A
Removal Timeout	'00'

Table 21: Select Next outcome parameters

23 Data Elements Dictionary

Data Elements used in a CPACE transaction are defined in [EMV Book A] Annex A, [EMV Book 3] Annex A, Annex B and Annex C and [EMV Book 4] Annex A, with the modifications specified in the following sections. There are modified elements from EMV specifications and new elements added in this specification.

23.1 Application Interchange Profile

Tag: '82'

Length: 2

Format: b

Description: Indicates the capabilities of the Card to support specific functions in the application. The Application Interchange Profile is returned in the response message of the GET PROCESSING OPTIONS command.

It is coded as specified in [EMV Book 3] Annex C.1. with the following modifications.

Byte	b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	X	-	-	-	-	-	-	-	RFU
	-	x	x	-	-	-	-	-	Not Used
	-	-	-	x	-	-	-	-	Cardholder Verification Supported
	-	-	-	0	-	-	-	-	Cardholder Verification is not Supported
	-	-	-	1	-	-	-	-	Cardholder Verification Supported
	-	-	-	-	x	-	-	-	Terminal risk management is to be Performed
	-	-	-	-	0	-	-	-	Terminal risk management is not to be Performed
	-	-	-	-	1	-	-	-	Terminal risk management is to be Performed
	-	-	-	-	-	0	-	-	Issuer Authentication using EXTERNAL AUTHENTICATE is not Supported
	-	-	-	-	-	-	x	-	CDCVM is Supported
	-	-	-	-	-	-	0	-	CDCVM is not Supported
	-	-	-	-	-	-	1	-	CDCVM is Supported
	-	-	-	-	-	-	-	x	Not Used
2	1	-	-	-	-	-	-	-	EMV Mode is Supported
	-	x	-	-	-	-	-	-	Not Used
	-	-	1	-	-	-	-	-	HCE is Supported
	-	-	-	x	x	x	x	-	RFU
	-	-	-	-	-	-	-	x	Relay Resistance Protocol Support
								0	Relay Resistance Protocol not Supported
								1	Relay Resistance Protocol Supported

Table 22: Application Interchange Profile (AIP) Coding

23.2 Device Application Capabilities

Template: 'BF0C'
Tag: ‘9F5D'
Length (in bytes): 3
Format: b

Description: This data element is returned by the card to inform the terminal about the support of the features described in Table 23.

Byte	b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	x	x	x	x	x	x	x	x	Reserved
2	x	x	x	x	x	x	x	-	Reserved
	-	-	-	-	-	-	-	x	CDA Support on AAC
	-	-	-	-	-	-	-	0	CDA not supported on AAC request
	-	-	-	-	-	-	-	1	CDA supported on AAC request
3	x	x	x	x	x	x	x	x	Reserved

Table 23 *Device Application Capabilities* coding

23.3 Cardholder Verification and Confirmation Status (CHV&CS)

Template: '77'
Tag: 'DF4B'
Length (in bytes): 3
Format: b

Description: This data element may be returned by a mobile device in the first GENERATE AC response to inform the kernel about the status of cardholder verification and cardholder confirmation and may influence the action flow of the merchant and cardholder at the POS.

Cardholder Verification and Confirmation Status (CHV&CS) is coded as shown in Table 24.

Byte	b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	X	x	x	x	x	x	x	x	Version Number
2	X	x	x	-	-	-	-	-	RFU
	-	-	-	x	-	-	-	-	Reserved
	-	-	-	-	x	-	-	-	Context is conflicting
	-	-	-	-	0	-	-	-	Context is not conflicting (no discrepancy is detected in the data used for a first presentment and the data used for a second presentment)
	-	-	-	-	1	-	-	-	Context is conflicting (a discrepancy is detected between the data used for a first presentment and the data used for a second presentment, the first and second presentment being both part of the same transaction)
	-	-	-	-	-	x	-	-	Not Used
	-	-	-	-	-	-	x	-	Cardholder confirmation (ACK) required
	-	-	-	-	-	-	0	-	ACK not required
	-	-	-	-	-	-	1	-	ACK required
	-	-	-	-	-	-	-	x	CDCVM required
	-	-	-	-	-	-	-	0	CDCVM not required
	-	-	-	-	-	-	-	1	CDCVM required
3	X	x	x	x	x	x	x	x	RFU

Table 24: *Cardholder Verification and Confirmation Status (CHV&CS)* Coding

23.4 CHV&CS Message Table

Template: --

Tag: --

Length (in bytes):

Format: B

Description: Table with the messages to be displayed to the cardholder depending on the value of *Cardholder Verification and Confirmation Status (CHV&CS)* returned in the GENERATE AC command response. The table shall be initialized with the default values defined in the Table 25: CHV&CS Message Table.

Bit in *Cardholder Verification and Confirmation Status (CHV&CS)*	UI Status	UI Message
‘000200’ (Cardholder confirmation (ACK) required)	Not Ready	'20' – See Phone
‘000100’ (CDCVM required)	Not Ready	'20' – See Phone

Table 25: CHV&CS Message Table

23.5 Contactless Transaction Limit with CDCVM

Template: --
Tag: --
Length (in bytes): 6
Format: n 12

Description: Maximum amount allowed for contactless transactions with cardholder verification performed using CDCVM

23.6 Contactless Transaction Limit without CDCVM

Template: --

Tag: --
Length (in bytes): 6
Format: n 12

Description: Maximum amount allowed for contactless transactions with cardholder verification performed using a CVM different from CDCVM

23.7 CVM Capabilities (above CVM Limit)

Tag: '--'

Length: 1

Format: b

Description: Used by the Kernel to overwrite **Terminal Capabilities** (byte 2, CVM Capability) for transactions where the value of **Amount, Authorised** is above or equal to the **Reader CVM Required Limit**.

23.8 CVM Capabilities (below or equal CVM Limit)

Tag: '--'

Length: 1

Format: b

Description: Used by the Kernel to overwrite **Terminal Capabilities** (byte 2, CVM Capability) for transactions where the value of Amount, Authorised is below the **Reader CVM Required Limit**.

23.9 Device Estimated Transmission Time For Relay Resistance R-APDU

Template: -
Tag: -
Length (in bytes): 2
Format: b

Description: *Device Estimated Transmission Time For Relay Resistance R-APDU* is a 2-byte binary value in units of hundreds of microseconds that represents the time taken to send the EXCHANGE RELAY RESISTANCE DATA response message.

Device Estimated Transmission Time For Relay Resistance R-APDU is included in the EXCHANGE RELAY RESISTANCE DATA response message and included in the generation of the dynamic signature by the card.

23.10 *Device Relay Resistance Entropy*

Template: -
Tag: -
Length (in bytes): 4
Format: b

Description: *Device Relay Resistance Entropy* is a 4-byte random number generated by the card.

Device Relay Resistance Entropy is included in the EXCHANGE RELAY RESISTANCE DATA response message and included in the generation of the dynamic signature by the card.

23.11 Field Off Hold Time

Template: --

Tag: --
Length (in bytes): 3
Format: n 6

Description: Field Off Hold Time indicates the time that the field is to be turned off before the field may be turned on again. The Field Off Hold Time is in units of 100ms.

23.12 Kernel Configuration

Template: --
Tag: ‘DF811B’
Length (in bytes): 1
Format: b

Description: Defines the kernel configuration options according to Table 26.

b8	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**Meaning**
x	x	-	-					RFU
-	-	1	-	-	-	-	-	CDCVM is Supported
-	-	-	1	-	-	-	-	Relay Resistance Protocol Support
				x	x	x	x	RFU

Table 26: *Kernel Configuration* Coding

23.13 Measured Relay Resistance Time

Template: --
Tag: --
Length (in bytes): 2
Format: b

Description: *Measured Relay Resistance Time* is a 2-byte binary value in units of hundreds of microseconds that represents the time measured by the Kernel for processing the EXCHANGE RELAY RESISTANCE DATA command.

23.14 Message Hold Time

Template: --
Tag: --
Length (in bytes): 3
Format: n 6

Description: Indicates the time that a message is to be held in the terminal’s cardholder display, before a new message may be displayed. The *Message Hold Time* is an integer in units of 100ms.

23.15 Max Time For Processing Relay Resistance APDU

Template: -
Tag: -
Length (in bytes): 2
Format: b

Description: *Max Time For Processing Relay Resistance APDU* is a 2-byte binary value in units of hundreds of microseconds that represents the maximum time for processing the EXCHANGE RELAY RESISTANCE DATA command.

Max Time For Processing Relay Resistance APDU is included in the EXCHANGE RELAY RESISTANCE DATA response message and for included in the generation of the dynamic signature by the card.

23.16 Max Time Relay Resistance Tolerance

Template: -
Tag: -
Length (in bytes): 2
Format: b

Description: *Max Time Relay Resistance Tolerance* is a 2-byte binary value in units of hundreds of microseconds that represents the allowed deviation to the maximum expect time to process the EXCHANGE RELAY RESISTANCE DATA command

23.17 Expected Min Transmission Time For RR Response

Template: -
Tag: -
Length (in bytes): 2
Format: B

Description: *Expected Min Transmission Time For RR Response* is a 2-byte binary value in units of hundreds of microseconds used to store the estimated minimum transmission time of the response APDU to the EXCHANGE RELAY RESISTANCE DATA command.

23.18 *Min Time For Processing Relay Resistance APDU*

Template: -
Tag: -
Length (in bytes): 2
Format: b

Description: *Min Time For Processing Relay Resistance APDU* is a 2-byte binary value in units of hundreds of microseconds that represents the minimum time for processing the EXCHANGE RELAY RESISTANCE DATA command.

Min Time For Processing Relay Resistance APDU is included in the EXCHANGE RELAY RESISTANCE DATA response message and included in the generation of the dynamic signature by the card.

23.19 *Min Time Relay Resistance Tolerance*

Template: -
Tag: -
Length (in bytes): 2
Format: b

Description: *Min Time Relay Resistance Tolerance* is a 2-byte binary value in units of hundreds of microseconds that represents the allowed deviation to the minimum expect time to process the EXCHANGE RELAY RESISTANCE DATA command.

23.20 *Relay Resistance Counter*

Template: -
Tag: -
Length (in bytes): 1
Format: b

Description: *Relay Resistance Counter* is a 1-byte binary counter used to count the number of times the EXCHANGE RELAY RESISTANCE DATA is executed.

23.21 Relay Resistance Min Time Difference Limit

Template: -
Tag: -
Length (in bytes): 2
Format: b

Description: *Relay Resistance Min Time Difference Limit* is a 2-byte binary value in units of hundreds of microseconds that represents the maximum allowed difference between the measured time and the minimum expected time to process the EXCHANGE RELAY RESISTANCE DATA command.

23.22 Relay Resistance Transmission Time Mismatch Limit

Template: -
Tag: -
Length (in bytes): 1
Format: b

Description: *Relay Resistance Min Time Difference Limit* is a 1-byte binary integer value that represents the maximum allowed value for the ratio between the *Device Estimated Transmission Time For Relay Resistance R-APDU* and the *Terminal Transmission Time For Relay Resistance Response* in percentage.

23.23 Terminal Transmission Time For Relay Resistance Command

Template: -
Tag: -
Length (in bytes): 2
Format: b

Description: *Terminal Transmission Time For Relay Resistance Command* is a 2-byte binary value in units of hundreds of microseconds that represents the transmission time of the EXCHANGE RELAY RESISTANCE DATA command.

23.24 *Terminal Transmission Time For Relay Resistance Response*

Template: -
Tag: -
Length (in bytes): 2
Format: b

Description: *Terminal Transmission Time For Relay Resistance Response* is a 2-byte binary value in units of hundreds of microseconds that represents the transmission time of the EXCHANGE RELAY RESISTANCE DATA command response.

23.25 *Terminal Relay Resistance Entropy*

Template: -
Tag: -
Length (in bytes): 4
Format: b

Description: *Terminal Relay Resistance Entropy* is a 4-byte random number generated by the Kernel and provided to the card in the command data field of the EXCHANGE RELAY RESISTANCE DATA command.

23.26 *Terminal Verification Results (TVR)*

Template: -
Tag: '95'
Length (in bytes): 5
Format: b

Description: *Terminal Verification Results (TVR)* indicate the status of the different functions as seen from the terminal as defined in [EMV Book 3].

In addition to the definition in [EMV Book 3], bits b4 through b1 of byte 5 that have been reserved for use by contactless specifications indicate the status of the Relay Resistance Protocol as seen from the terminal.

Terminal Verification Results (TVR) are coded as shown in Table 27.

Byte	b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	1	-	-	-	-	-	-	-	Offline data authentication was not performed
	-	1	-	-	-	-	-	-	SDA failed
	-	-	1	-	-	-	-	-	ICC data missing
	-	-	-	1	-	-	-	-	Card appears on terminal exception file
	-	-	-	-	1	-	-	-	DDA failed
	-	-	-	-	-	1	-	-	CDA failed
	-	-	-	-	-	-	x	x	RFU
2	1	-	-	-	-	-	-	-	ICC and terminal have different application versions
	-	1	-	-	-	-	-	-	Expired application
	-	-	1	-	-	-	-	-	Application not yet effective
	-	-	-	1	-	-	-	-	Requested service not allowed for card product
	-	-	-	-	1	-	-	-	New card
	-	-	-	-	-	x	x	x	RFU
3	1	-	-	-	-	-	-	-	Cardholder verification was not successful
	-	1	-	-	-	-	-	-	Unrecognised CVM
	-	-	1	-	-	-	-	-	PIN Try Limit exceeded
	-	-	-	1	-	-	-	-	PIN entry required and PIN pad not present or not working
	-	-	-	-	1	-	-	-	PIN entry required, PIN pad present, but PIN was not entered
	-	-	-	-	-	1	-	-	Online PIN entered
	-	-	-	-	-	-	x	x	RFU
4	1	-	-	-	-	-	-	-	Transaction exceeds floor limit
	-	1	-	-	-	-	-	-	Lower consecutive offline limit exceeded
	-	-	1	-	-	-	-	-	Upper consecutive offline limit exceeded
	-	-	-	1	-	-	-	-	Transaction selected randomly for online processing
	-	-	-	-	1	-	-	-	Merchant forced transaction online
	-	-	-	-	-	x	x	x	RFU
5	1	-	-	-	-	-	-	-	Default TDOL used
	-	1	-	-	-	-	-	-	Issuer authentication failed
	-	-	1	-	-	-	-	-	Script processing failed before final GENERATE AC
	-	-	-	1	-	-	-	-	Script processing failed after final GENERATE AC
	-	-	-	-	1	-	-	-	Relay resistance threshold exceeded
	-	-	-	-	-	1	-	-	Relay resistance time limits exceeded
	-	-	-	-	-	-	x	x	Relay Resistance Protocol performed
	-	-	-	-	-	-	0	0	RRP not supported
	-	-	-	-	-	-	0	1	RRP not performed
	-	-	-	-	-	-	1	0	RRP performed
	-	-	-	-	-	-	1	1	RFU

Table 27: *Terminal Verification Results (TVR)* Coding

23.27 Third Party Data

Template: 'BF0C' or '70'
Tag: '9F6E'
Length (in bytes): 5-32
Format: b

Description: *Third Party Data* contains various information, possibly including information from a third party. If present in the card, *Third Party Data* must be returned in a file read using the READ RECORD command or in the *FCI Issuer Discretionary Data* template.

Third Party Data is coded as shown in Table 28.

'Device Type' is present when the most significant bit of byte 1 of 'Unique Identifier' is set to 0b. In this case, the maximum length of 'Proprietary Data' is 26 bytes. Otherwise it is 28 bytes.

Data Field	**Length (in bytes)**	**Format**	**Value**
Country Code	2	n 3	Country Code according to [ISO 3166-1]
Unique Identifier	2	b	Value assigned by the scheme
Device Type	0 or 2	an	As shown in Table 29
Proprietary Data	1-26 or 28	b	

Table 28: *Third Party Data* Coding

Code	**Device Type**
“00”	Dual interface card
“02”	Key Fob
“03”	Watch
“04”	Tag
“05”	Wristband
“06”	Mobile Device Sleeve
“07”	Non-removable Secure Element in a Mobile Device
“08”	Removable Secure Element in a Mobile Device
“14”	Host Card Emulation in a Mobile Device
“16”	Trusted Execution Environment in a Mobile Device
Other Values	RFU

Table 29: Device Type Codes