EMV®
Contactless Specifications for Payment Systems

Book C-7

Kernel 7 Specification

Version 2.6
February 2016

Legal Notice

Unless the user has an applicable separate agreement with EMVCo or with the applicable payment system, any and all uses of these Specifications is subject to the terms and conditions of the EMVCo Terms of Use agreement available at www.emvco.com and the following supplemental terms and conditions.

Except as otherwise may be expressly provided in a separate agreement with EMVCo, the license granted in the EMVCo Terms of Use specifically excludes (a) the right to disclose, distribute or publicly display these Specifications or otherwise make these Specifications available to any third party, and (b) the right to make, use, sell, offer for sale, or import any software or hardware that practices, in whole or in part, these Specifications. Further, EMVCo does not grant any right to use the Kernel Specifications to develop contactless payment applications designed for use on a Card (or components of such applications). As used in these supplemental terms and conditions, the term “Card” means a proximity integrated circuit card or other device containing an integrated circuit chip designed to facilitate contactless payment transactions. Additionally, a Card may include a contact interface and/or magnetic stripe used to facilitate payment transactions. To use the Specifications to develop contactless payment applications designed for use on a Card (or components of such applications), please contact the applicable payment system. To use the Specifications to develop or manufacture products, or in any other manner not provided in the EMVCo Terms of Use, please contact EMVCo.

These Specifications are provided "AS IS" without warranties of any kind, and EMVCo neither assumes nor accepts any liability for any errors or omissions contained in these Specifications. EMVCO DISCLAIMS ALL REPRESENTATIONS AND WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AS TO THESE SPECIFICATIONS.

EMVCo makes no representations or warranties with respect to intellectual property rights of any third parties in or in relation to the Specifications. EMVCo undertakes no responsibility to determine whether any implementation of these Specifications may violate, infringe, or otherwise exercise the patent, copyright, trademark, trade secret, know-how, or other intellectual property rights of third parties, and thus any person who implements any part of these Specifications should consult an intellectual property attorney before any such implementation.

Without limiting the foregoing, the Specifications may provide for the use of public key encryption and other technology, which may be the subject matter of patents in several countries. Any party seeking to implement these Specifications is solely responsible for determining whether its activities require a license to any such technology, including for patents on public key encryption technology. EMVCo shall not be liable under any theory for any party's infringement of any intellectual property rights in connection with these Specifications.

Contents

Figures

Tables

Requirements

1 General

This chapter contains information that helps the reader understand and use this specification.

1.1 Scope

This document, the *EMV Contactless Specifications for Payment Systems, Kernel 7 Specification*, describes one of several kernels defined for use with Entry Point.

1.2 Audience

This specification is intended to be used by system designers in payment systems and financial institution staff responsible for implementing financial applications.

1.3 Volumes of the Contactless Specifications

This specification is part of a ten-volume set:

Book A: Architecture and General Requirements

Book B: Entry Point Specification

Book C-1: Kernel 1 Specification

Book C-2: Kernel 2 Specification

Book C-3: Kernel 3 Specification

Book C-4: Kernel 4 Specification

Book C-5: Kernel 5 Specification

Book C-6: Kernel 6 Specification

Book C-7: Kernel 7 Specification

Book D: Contactless Communication Protocol Specification

1.4 Reference Materials

The following specifications and standards contain provisions that are referenced in this specification. The latest version shall apply unless a publication date is explicitly stated.

If any provision or definition in this specification differs from those in the listed specifications and standards, the provision or definition herein shall take precedence.

[EMV 4.3]	*EMV Integrated Circuit Card Specifications for Payment Systems*, Version 4.3, November 2011, including:
[EMV 4.3 Book 1]	*EMV Integrated Circuit Card Specifications for Payment Systems*, Book 1, Application Independent ICC to Terminal Interface Requirements
[EMV 4.3 Book 2]	*EMV Integrated Circuit Card Specifications for Payment Systems*, Book 2, Security and Key Management
[EMV 4.3 Book 3]	*EMV Integrated Circuit Card Specifications for Payment Systems*, Book 3, Application Specification
[EMV 4.3 Book 4]	*EMV Integrated Circuit Card Specifications for Payment Systems*, Book 4, Cardholder, Attendant, and Acquirer Interface Requirements

1.5 Overview

This volume includes the following chapters and annexes:

Chapter 1 contains general information that helps reader to understand the structure of this specification so that they could read it effectively.

Chapter 2 introduces the main architecture of the POS System supported by Kernel7.

Chapter 4.3 gives an overview of the features of Kernel7 and its configuration and main functions.

Chapter 4 breaks down the main transaction steps, within each details are provided to describe the necessary data to be processed, commands used and Kernel processing as well.

Annex A contains the dictionary of data objects used by Kernel7,including the kernel data elements and card elements must be supported by Kernel7 as well.

Annex B is a sample of Fast Dynamic Data Authentications and provides relevant explanations.

Annex C lists authorization and clearing data needed to be provided by Kernel7 when the Kernel sends an ONLINE REQUEST or APPROVE as an output to Entry Point.

Annex D describes the process of retrieving card transaction log.

Annex E is a glossary of terms and abbreviations in this specification.

2 POS System

This chapter lists the possible physical architectures of POS system conducting EMV Mode contactless transactions; transactions using contact interface or magstripe are out of the scope of this specification.

POS System shall be able to host the following functions in Kernel7:

- Interact with contactless card
- Process application selection, complete kernel activation and transfer transaction outcomes
- Display transaction information to cardholders
- Display transaction information to merchants
- User Interface to input transaction amount
- Cardholder verification
- Online capabilities
- Store transaction data used for offline transaction clearing

In terms of hardware designing, it could be divided into the following 3 categories:

1. Programmable Contactless Readers (PCRs)

A PCR is a smart reader which is able to conduct all or partial kernel functions including providing the contactless interface with the card and processing the transaction data. A PCR is a programmable device to achieve alternative functions.

2. Combination of Terminal and Non-programmable Contactless Readers (Non-PCRs)

All the features in a Non-PCR are hardcoded and cannot be re-programmed once released. A Non-PCR is a separate entity connected with a terminal and it execute instructions from the terminal to interact with card, control visual indicators, audio indication and LCD display, and transfer outcomes as well.

Kernel functions are performed in the terminal.

3. Fully Integrated Terminal

Modules including contactless reading module are integrated in one device.

The design described in this document is based on a physical architecture that is along the lines of Fully Integrated Terminal; however it is not intended to be prescriptive.

The logical partitioning of the Fully Integrated Terminal is illustrated Figure 2-1.

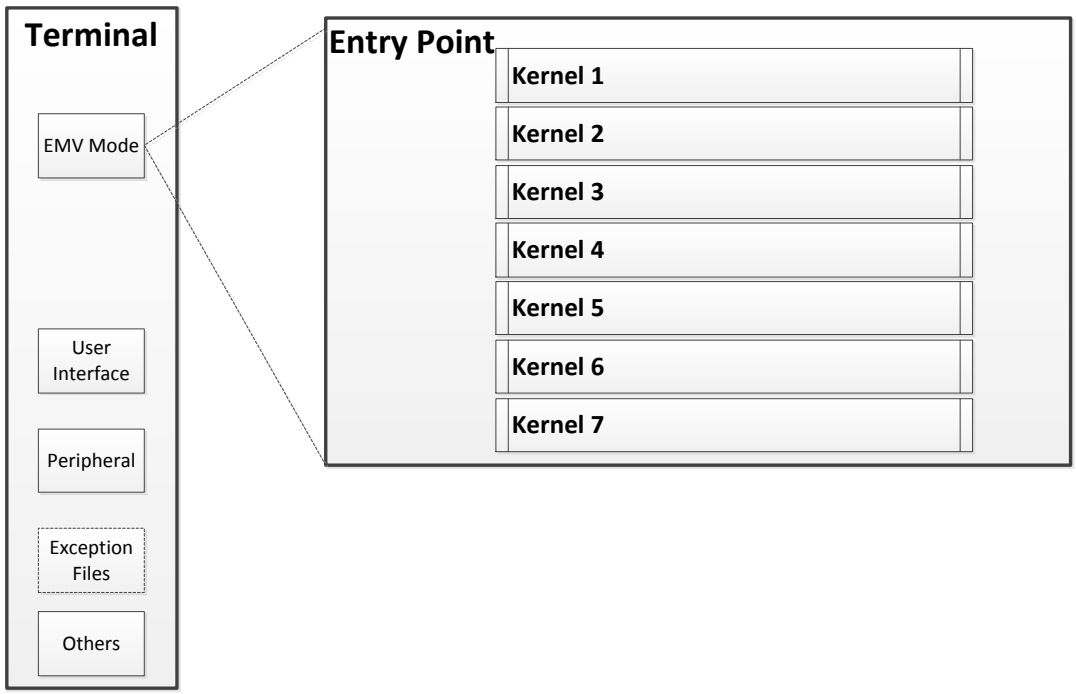

Figure 2-1 Logical Architecture

The three main components in the above figure are frequently used in this specification and each of them is responsible for specific functions as described below:

Terminal – within this specification, the Terminal is a term embraces a range of devices that provide the interaction for card acceptance at physical locations. The Terminal hosts the user interface for card holder and merchant, Payment System network connectivity and peripheral devices such as PIN pads and receipt printers.

Entry Point – see *Book A* and *Book B*.

Kernel – the Kernel within this specification refers to the Kernel 7 unless the word is followed by a specific number.

3 Processing Overview

3.1 General Description

Entry Point activates Kernel7 once completing Application Selection and passes the control to the Kernel. The subsequent processing will be completed jointly by the Kernel and Card. When the control transferred from Entry Point to the Kernel, Entry Point shall provide transaction data, relevant configuration parameters and FCI information returned in Application Selection. The EMV Mode supported by Kernel 7 is an optimization of EMV full transaction processing and command sequences to guarantee the time that the cardholder holds the payment card close to the terminal be as minimal as possible. Meanwhile, it supports the same data authentication working in EMV full procedures, which allows for a quick transaction with a required security level.

Kernel 7 possesses the following characteristics compared with EMV full processing:

1）In case that the Kernel outputs an ONLINE REQUEST, only online authorization shall be implemented; Offline Data Authentication is conducted when the transaction is processed offline.

2）Online PIN and signature are the only two Cardholder Verification Methods;

3）If the transaction is processed offline, Offline Data Authentication is performed by the Kernel and there's no requirement of card's presences in the communication field.

Once the Kernel completes its processing, it shall output the transaction Outcome to Entry Point. Meanwhile the control has been transferred from the Kernel to Entry Point for further handling, which starts from initiating an appropriate Start point based on the Outcome and parameters in the Outcome. (Start point refers to different stages of Entry Point in EMV contactless specification, including Start A, Start B, Start C and Start D. Refer to Book A: Architecture and General Requirements for details).

Kernel 7 includes the following steps:

- Kernel Activation (Mandatory)
- Application Initiation (Mandatory)
- Read Application Data (Conditional, if the transaction is authorised offline)
- Offline Data Authentication (Conditional, if the transaction is authorised offline))

- Online Processing (Conditional, if the transaction is authorized online)
- Transaction completion (Mandatory)

3.2 Overview of Transaction Processing

3.2.1 Kernel Activation

Entry Point activates the Kernel once completing application selection, transfers transaction processing control to the Kernel and provides card information as well as transaction information to the Kernel for further use. On Kernel activation, the interaction between the Kernel and the card has been established and the connection will be maintained until the Kernel returns an Outcome to Entry Point.

Entry Point's output to the Kernel includes the pre-processing results, the picked combination profile, transaction data and configuration parameters. All these information passed to the Kernel are crucial input for Transaction processing These data include the following elements:

- Transaction Amount;
- Terminal Transaction Qualifiers (TTQ) (Pre-configured in the Terminal which will be reset in Entry Point pre-processing);
- Terminal Unpredictable Number;
- Copy of TTQ (Reset in Entry Point pre-processing)
- FCI of the selected application in pre-processing;

3.2.2 Application Initialization

Once the Kernel receives the copy of from Entry Point, it shall do a reset:

1) Reset the copy of TTQ Byte 3 to "0";
2) Reset the copy of TTQ Byte 4 bit 8 to "1".

See Table 3-1 for the definition of TTQ supported by this specification. If not specified, the TTQ in subsequent context refer to TTQ reset in this step.

The Kernel informs the card of the start of transaction by issuing GET PROCESSING OPTIONS (GPO) command, and includes any data that the card requests in PDOL during pre-processing.

The card provides transaction disposition in GPO response. If the card requests an offline transaction and fDDA shall be performed to complete the transaction, the card shall generate dynamic signature data.

If the returned Application Cryptogram is Offline Authorization, the Kernel shall read card data according to Application File Locator (AFL) returned during application initialization. Details of Read Record can be found in section 3.2.3. Read Record will proceed until the last record has been transferred when card may leave the field.

If Application Cryptogram is Online Authorization, the Kernel shall not read application data, instead it shall perform online processing once got GPO response. Details of Online Processing can be found in section 3.2.5. Card may leave the field once returning GPO response.

Table 3-1 Terminal Transaction Qualifiers

Byte	Bit	Definition
1	8	RFU[1]
	7	1 – Full transaction flow in Contactless interface Support 0 – Full transaction flow in Contactless interface Not Support
	6	1 – EMV Mode Supported 0 – EMV Mode Not supported
	5	1 – Full transaction flow in contact interface Support 0 –Full transaction flow in contact interface Not Support
	4	1 – Offline-only terminal 0 – Online-capable terminal
	3	1 – Online PIN Supported 0 – Online PIN Not Supported
	2	1 – Signature Supported 0 – Signature Not Supported
	1	1 – Offline Data Authentication for Online Authorisation Supported 0 – Offline Data Authentication for Online Authorisation Not Supported

[1]All RFU bits and bytes shall be set to zero unless explicitly specified otherwise.

Byte	Bit	Definition
2	8	1 – Request Online cryptogram 0 – No Online cryptogram
	7	1 –CVM Requested 0 – No CVM Requested
	6-1	RFU
3	8	RFU
	7	1- Consumer Device CVM Supported 0- Consumer Device CVM Not Supported
	6-1	RFU
4	8	1 –fDDA v1.0 Supported
	7-1	RFU

3.2.3 Read Application Data

Read Application Data Request is only performed in offline transactions and will be followed with Offline Data Authentication.

If the card requests offline authorization, it shall return AFL in the GPO response. The Kernel uses READ RECORD command to retrieve the card data according to Short File Identifier (SFI) and record number in AFL.

The Electronic Cash(EC) balance will not be updated in a persistent memory until the last record is read by the Kernel. Once the last record is sent to the Kernel, the card may leave the field and the Kernel performs Offline Data Authentication. Details of Offline Data Authentication can be found in section 0.

3.2.4 Offline Data Authentication

Offline Data Authentication shall be supported by Offline-capable terminals and can be performed in both online and offline transaction. Fast Dynamic Data Authentication (fDDA) is used to verify the dynamic signature and authenticate the data from the card. fDDA is the Dynamic Data Authentication(DDA) specifically used in contactless interface which not only to guarantee that the card data has not been broken or manipulated since issued, but also to confirm the legitimacy of critical ICC-resident/generated data and data received from the terminal. This precludes the counterfeiting of any such card. In order to perform fDDA in offline transaction, the Kernel shall be capable to quickly shield SDA support function when necessary. Once SDA is shielded, transactions could not be approved offline unless the card supports fDDA.

fDDA is different from standard DDA in the following aspects:

- Instead of using INTERNAL AUTHENTICATE Command, the card generates Dynamic signature on receiving GPO command. DDOL is not supported by Kernel7.
- The result of fDDA verification is not stored in Terminal Verification Results (TVR) and sent to Issuer in online message, neither is it encrypted in online authorization or clearing data.

There are two versions of fDDA including version00 and version01 and an offline-capable Kernel shall only support version 01. All the fDDA in this specification refer to version01 by default unless explicitly specified otherwise.

3.2.5 Online Processing

If the card's transaction disposition indicates an Online Authorization in the GPO response, the Kernel sends an ***Online Request*** Outcome with parameters set appropriately and clearing data as well (see 4.5.2). Online–capable terminal perform online processing once Entry Point receives the outcome and alter the transaction disposition to ***Declined*** Outcome with parameters if it failed to go online .(see 4.5.4)

Requirement – Online Authorization

3.2.5.1 The Kernel shall apply the following principles when performing online authorization:

If TTQ Byte 1 bit 4 is "0", meaning the terminal is an Online-capable terminal,

then the Kernel shall provide an ***Online Request*** *Outcome*, and transfers the clearing data to Entry Point (see 4.5.2 for parameter configuration) to perform an online transaction.

Else the Kernel declines the transaction with a ***Declined*** Outcome(see 4.5.4).

The terminal sends an authorisation request to the issuer host. Online Processing allows the issuer host to review and authorise or decline transactions using the

issuer's host based risk management parameters.(Risk management in issuer host is out of this specification) If the issuer approves the transaction, the account will be updated in the host whilst the EC balance on the card remains unchanged.

On receiving the decision from the issuer, the terminal displays the transaction disposition to the cardholder. The transaction is considered as completed and the Kernel will no longer be re-activate.

3.2.6 Transaction Completion

On the perspective of the Kernel, an Outcome provided to Entry Point is considered as the last step indicating the transaction completion. The outcomes may be the followings:

- ***Approved***
- ***Online Request***
- ***Try Again***
- ***Declined***
- ***Try Another Interface***
- ***Select Next***
- ***End Application***

See 4.5 for details.

3.3 General Transaction Flow

Figure 3-1 illustrates a sample of a complete transaction in EMV Mode.

Before handing over the control to the Kernel, Entry Point finishes the pre-processing (Start A), followed with establishing contactless communication connection (Start B), then selects combination (Start C) and ends with activating Kernel (Start D).

The Kernel handles the interaction with the card until the transaction disposition has been provided to Entry Point, attached with the necessary parameters and clearing data if any.

Entry Point takes over the outcomes with the parameters and finishes the following procedure including Online processing, CVM and transaction disposition display without re-starting the Kernel.

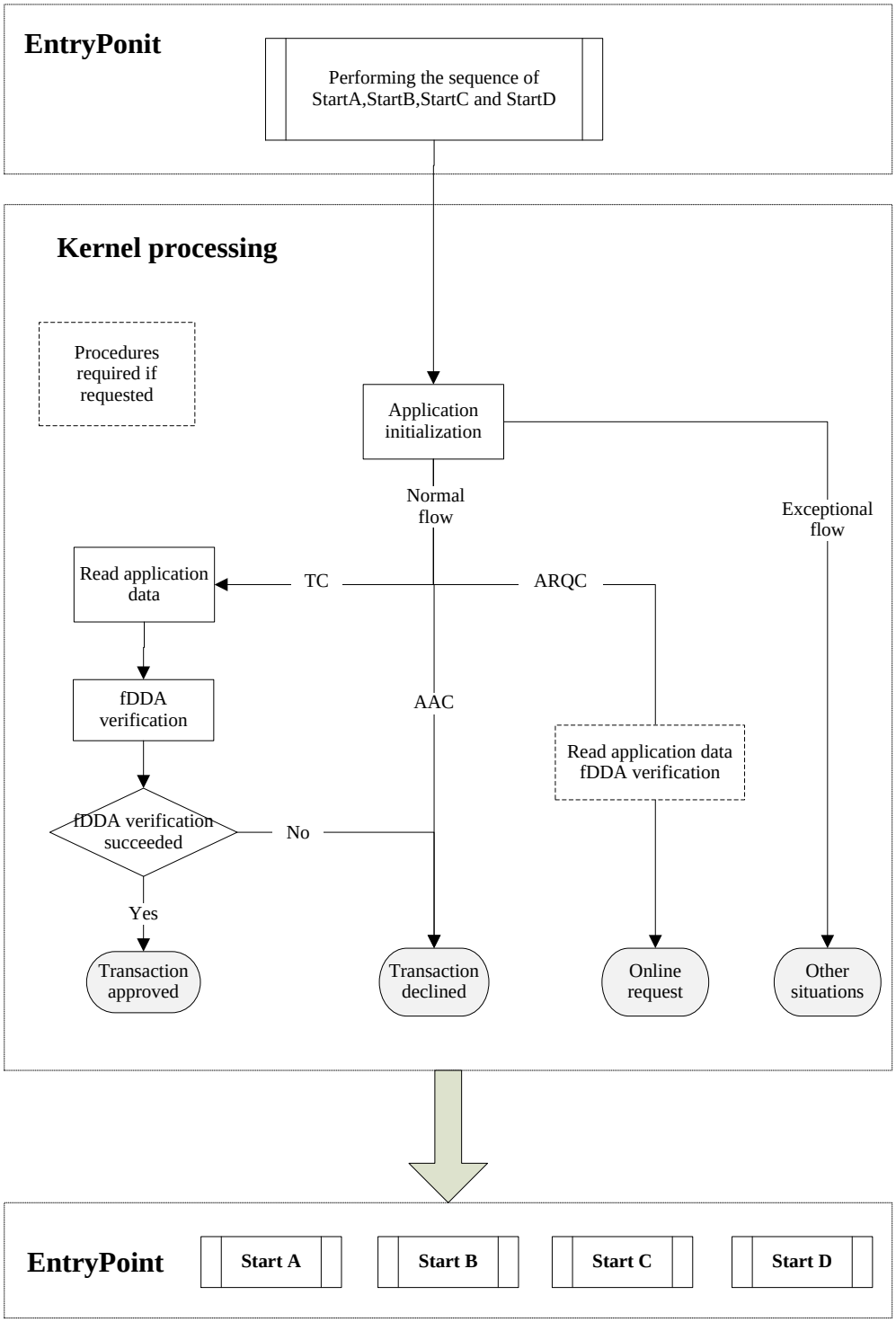

Figure 3-1 Sample of EMV Mode Transaction Flow

4 Transaction Processing

4.1 Application Initialization

4.1.1 Input

The input of Application Initialization includes the data passed from Entry Point which is the result of pre-processing, the data can be found in 3.2.1.

Table 4-1 illustrates response of SELECT AID command which includes the FCI template of the selected application:

Table 4-1 Response of SELECT AID

Tag	Value			Existence
'6F'	FCI Module			M
	'84'	DF name		M
	'A5'	Special module for FCI data		M
		'9F38'	PDOL	M
		'50'	Application tag	O
		'87'	Application priority indicator	O
		'5F2D'	Preferred language	O
		'9F11'	Issuer Code Table Index	O
		'9F12'	Application Priority Name	O
		'BF0C'	Issuer Discretionary data (FCI)	O
		'xxxx'	One or more additional (special) data elements from application provider, issuer or IC card provider	O

4.1.2 Commands

The Kernel uses GPO command in Application Initialization. The format of GPO command conforms to [*EMV 4.3 Book 3*].

Definition and Scope

The GET PROCESSING OPTIONS command initiates the transaction within the ICC.

Command Message

Table 4-2 GPO Command Message

Code	Value
CLA	“80”
INS	“A8”
P1	“00”; all other values are RFU
P2	“00”; all other values are RFU
Lc	var.
Data	Processing Options Data Object List (PDOL) related data
Le	“00”

Data Field Sent in the Command Message

The data field of the command message contains all the Kernel data requested in the PDOL returned during pre-processing in Entry Point.

Data Field Returned in the Response Message

The coding of the data object shall be according to Format 2 described in [EMV 4.3 Book3] see 6.5.8.4.

4.1.3 Flow Chart

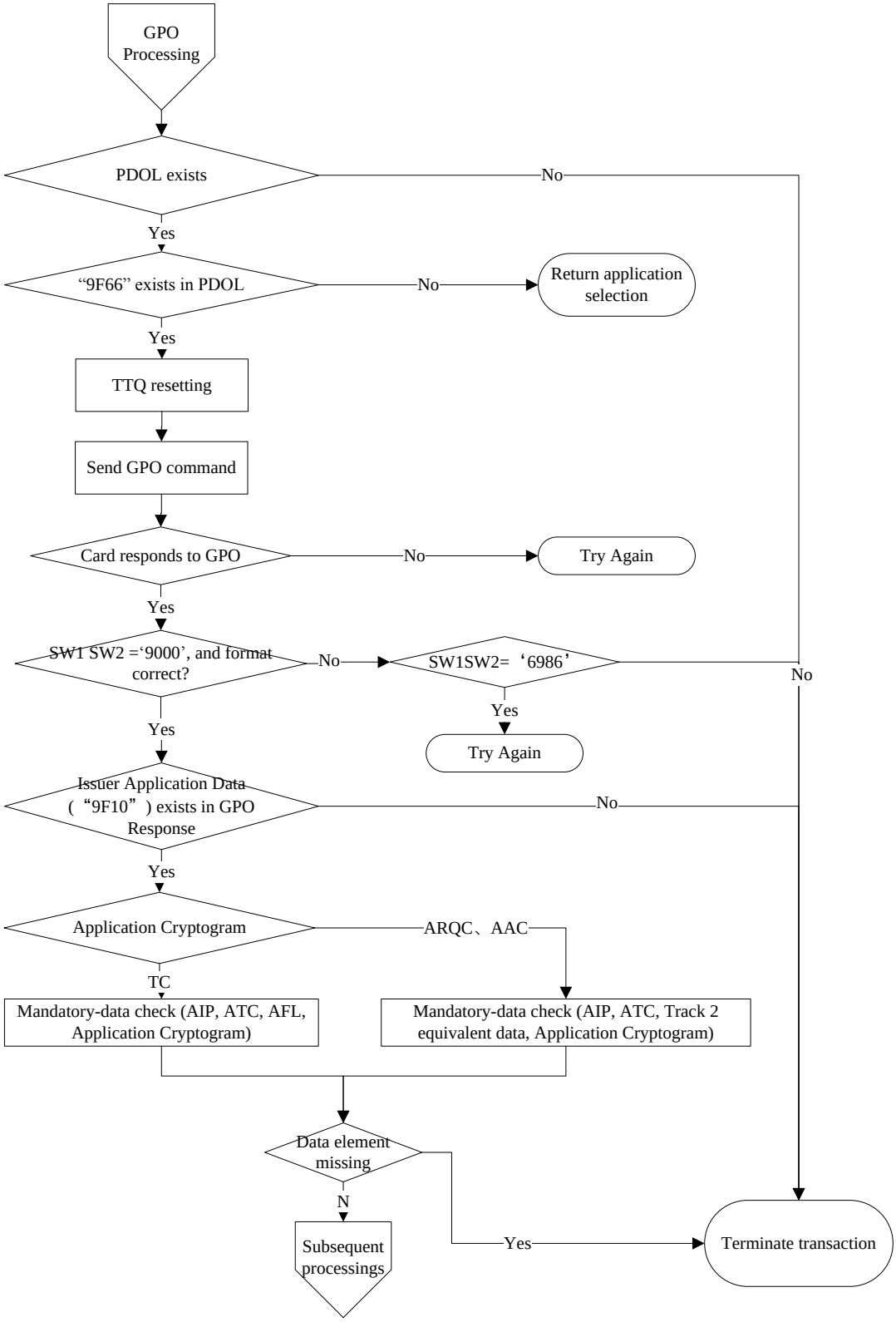

Figure 4-1 GPO Processing

4.1.4 Kernel Processing

Requirement – PDOL Check

4.1.4.1 The Kernel shall check the existence of PDOL and Terminal Transaction Qualifiers (Tag "9F66"):

If PDOL doesn't exist in the FCI of the selected application which is passed over from Entry Point,

Then the Kernel shall terminate the processing and select next application in the candidate list by providing a ***Select Next*** Outcome, see 4.5.6.

Else continue to check the existence of Terminal Transaction Qualifiers (Tag '9F66') in PDOL.

If exists,

Then the Kernel continues with TTQ resetting, see 4.1.4.2.

Else the Kernel shall terminate the processing and select next application in the candidate list by providing a ***Select Next*** Outcome, see 4.5.6.

Requirement – TTQ Resetting

4.1.4.2 The Kernel shall reset the copy of TTQ passed from Entry Point on Kernel Activation

: Set the copy of TTQ Byte 4 bit 8 to "1".

Requirement – Processing of Status Code

4.1.4.3 The Kernel shall check SW1 SW2 in GPO response:

If any L1 error[2] occurs,

Then the Kernel shall terminate the processing by providing a ***Try***

[2] L1 errors can be one of the following: timeout, protocol error or transmission error. See *[EMV Contactless Book D]* for details.

Requirement – Processing of Status Code

Again Outcome, which will remind the cardholder to represent the card, see 4.5.3.

If SW1 SW2 does NOT equal to "9000",

If SW1 SW2 equals to “6986”

Then the Kernel shall terminate the processing by providing a Try Again Outcome, which will remind the cardholder to represent the card, see 4.6.8.

Else If TTQ Bit 5 Byte 1 is "1", indicating Contact interface is supported,

or Magstripe is supported in the terminal,

Then the Kernel shall terminate the processing by providing a ***Try Another Interface*** Outcome, which will remind the cardholder to conduct the transaction in another interface. The Kernel sets the Parameters as a result of the following check:

If the terminal supports contact interface,

Then the Kernel sets *Alternate Interface Preference* to *Contact Chip*.

Else the Kernel sets *Alternate Interface Preference* to *Mag-stripe*.

Else the Kernel shall terminate the processing by providing an ***End Application*** Outcome, see 4.5.7.

If SW1 SW2 equals to "9000",

Then the Kernel checks the data format of GPO response. Only Format 2 of GPO response is supported, see [EMV 4.3 Book3] see 6.5.8.4..

If there is any format error,

Then the Kernel shall terminate the processing by providing an ***End Application*** Outcome, see 4.5.7

Else the Kernel continues with The Card’s Transaction Disposition, see 4.1.4.4.

Requirement –Transaction Disposition

4.1.4.4 The Card's transaction disposition is included in the Cryptogram Information Data (Tag "9F27"). If the card does not return CID, the Kernel shall:

- Initialise CID to '00'
- Set CID bits 8-7 to the value of Issuer Application Data(Tag "9F10") Byte 5 bit 6-5

If the card returns an ARQC (CID bit8-7 = '10'), indicting an online transaction is requested.

Then the Kernel continues with mandatory-data check for online request, see 4.1.4.5.

If the card returns an AAC(CID bit8-7='00'), indicting an offline decline.

Then the Kernel continues with mandatory-data check for offline decline, see 4.1.4.5.

If the card returns a TC (CID bit8-7='01'), indicting an offline approval.

Then the Kernel continues with mandatory-data check for offline approval, see 4.1.4.6.

Requirement – Mandatory-Data Check

4.1.4.5 Table 4-3 illustrate the data returned in GPO response if the card's transaction disposition is an ARQC or AAC, the Kernel shall perform the followings:

If the card's disposition is ARQC,

If AFL is not returned, **Then**

If any Mandatory data listed in Table 4-3 is absent in the GPO response,

Then the Kernel shall terminate the Kernel processing by providing an ***End Application*** Outcome, see 4.5.7.

Else the Kernel perform the Online Processing, see 3.2.5

Else (AFL is returned) the Kernel continues with AFL format check, see 4.1.4.7.

If the card's disposition is AAC.

If any Mandatory data listed in Table 4-3 is absent,

Then the Kernel shall terminate the processing by providing an ***End Application*** Outcome, see 4.5.7.

Else the Kernel declines the transaction by providing a ***Declined*** Outcome, see 4.5.4

4.1.4.6 Table 4-4 illustrate the data returned in GPO response if the card's transaction disposition is a TC, the Kernel shall perform the followings:

If the card's transaction disposition is TC,

If any Mandatory data elements listed in Table 4-4 is absent,

Then the Kernel shall terminate the processing by providing an ***End Application*** Outcome, see 4.5.7.

Else the Kernel continues with AFL format check, see 4.1.4.7.

Requirement – Mandatory-Data Check

4.1.4.7 Table 4-5 illustrates the definition and format of AFL. The Kernel shall perform the followings:

If No Entry exists in AFL,

OR any one or more of the following format errors occurs:

- An SFI of 0 or 31;
- A starting record number of 0;
- An ending record number less than the starting record number (byte 3 < byte 2).
- Number of records participating in offline data authentication greater than the number of records (byte 4 > byte 3 - byte 2 + 1).

Then the Kernel shall terminate the processing by providing an ***End Application*** Outcome, see 4.5.7.

Else the Kernel continues with Read Application Data, see 4.2.

Table 4-3 GPO Response Data if The Card's Transaction Disposition is an ARQC or AAC

Tag	Mandatory (M) Optional (O) Conditional (C)	Data Element Name
"82"	M	AIP
"94"	C If Offline Authentication Data is supported and requested.	AFL, this data shall not be returned in AAC
"9F36"	M	ATC
"57"	M	Track 2 Equivalent Data
"9F10"	M	Issuer Application Data.

Tag	Mandatory (M) Optional (O) Conditional (C)	Data Element Name
“9F26”	M	Application Cryptogram
“9F27”	M	Cryptogram Information Data
“9F4B”	C If Offline Authentication Data is supported and requested. If fDDA is supported and the length of ICC private key is less than or equal to 1024 bits.	Signed Dynamic Application Data, this data shall not be returned in AAC
“9F63”	C If exists in card	Product Identification Information
“5F34”	C If exists in card	Application PAN Sequence Number
“9F6C”	C If exists in card	Card Transaction Qualifiers
“9F5D”	C If the Available Offline Amount display is allowed	Available Offline Spending Amount. Only if the Available Offline Spending Amount("9F5D") is personalized to 1, the card returns this data in GPO response. Meanwhile, the Card Additional Processing (Byte 1 bit 1) shall be personalised to 1 to indicate that this amount will be calculated and included in all contactless transactions. Personalization of Available Offline Spending Amount to 1 also means that this data can be retrieved by GET DATA command.
“5F20” or “9F0B”	O	Cardholder Name If the length of Cardholder Name is less than or equal to 26 bytes, "5F20" will be returned; if the length of Cardholder Name is more than 26 bytes, "9F0B" will be returned.

Table 4-4 GPO Response Data if The Card's Transaction Disposition is a TC

Tag	Mandatory (M) Conditional (C)	Data element name
"82"	M	AIP
"94"	M	AFL
"9F36"	M	ATC
"9F26"	M	Application Cryptogram
"9F10"	M	Issuer application data
"9F27"	M	Cryptogram Information Data
"57"	C If Track 2 Equivalent Data is not part of Signed Static Application Data.	Track 2 Equivalent Data
"5F34"	C If exists in card	Application PAN Sequence Number.
"9F4B"	C If fDDA is supported and the length of ICC private key is less than or equal to 1024 bits.	Signed Dynamic Application Data
"9F6C"	C If exists in card	Card Transaction Qualifiers

Tag	Mandatory (M) Conditional (C)	Data element name
“9F5D”	C If the Available Offline Amount display is allowed and IC card private key length is shorter or equal to 1024 bits.	Available Offline Spending Amount. Only if the Available Offline Spending Amount("9F5D") is personalized to 1, the card returns this data in GPO response. Meanwhile, the Card Additional Processing (Byte 1 bit 1) shall be personalised to 1 to indicate that this amount will be calculated and included in all contactless transactions. Personalization of Available Offline Spending Amount to 1 also means that this data can be retrieved by GET DATA command.

4.2 Read Application Data

Read Application Data can be performed in both online and offline transactions.

According to AFL in the GPO response, the Kernel reads the data contained in the card to proceed fDDA verification and various functions including checking the Application Expiration date and Exception File if any.

4.2.1 Input

Table 4-5 lists the data returned from the card in Application Initialization which will be used in Read Application Data.

Table 4-5 Card Data used in Read Application Data

Data Object	Description
Application File Locator (AFL)	Indicates the file location and range of records which contain card data to be read by the Kernel. For each file to be read, the AFL contains the following information: • Byte 1 - Short File Identifier (a numeric file label) • Byte 2 - Record number of the first record to be read • Byte 3 - Record number of the last record to be read • Byte 4 - Number of consecutive records containing data to be used in Offline Data Authentication beginning with the first record to be read as indicated in Byte 2.

4.2.2 Commands

The Kernel uses READ RECORD command in Read Application Data. The format of READ RECORD command conforms to [*EMV 4.3 Book 3*].

4.2.3 Flow Chart

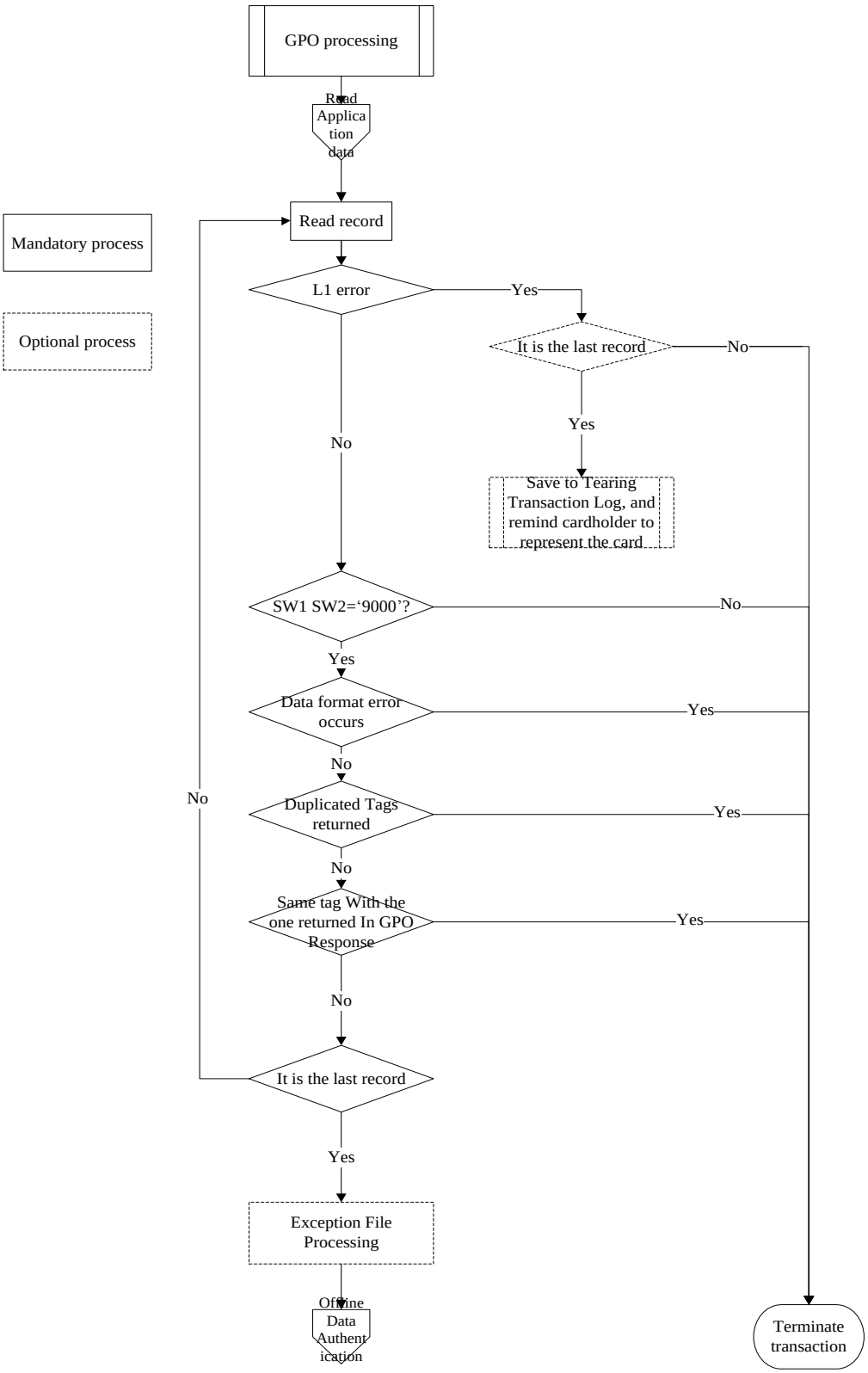

Figure 4-2 Read Application Data

4.2.4 Kernel Processing

Each file entry in AFL contains the information of consecutive records in the AEF. For each record, the Kernel shall send a READ RECORD command until the last record has been retrieved. If there are more than one file entry in AFL, the Kernel repeats the steps until all the entries have been process.

Requirement – L1 Errors in Reading Application Data

4.2.4.1 In case of L1 errors, the Kernel shall perform the followings:

If any L1 error occurs,

THEN the Kernel shall terminate the processing by providing a ***Try Again*** Outcome, which will remind the cardholder to represent the card, see 4.5.3.

Else the Kernel continues with Response Processing, see 4.2.4.2.

Requirement – Processing of READ RECORD Response

4.2.4.2 The Kernel shall process the READ RECORD Response:

If SW1 SW2 does NOT equal to '9000',

Then the Kernel shall terminate the processing by providing an ***End Application*** Outcome, see 4.5.7.

Else the Kernel performs Data Format check , see 4.2.4.3.

4.2.4.3 The Kernel shall check the data in READ RECORD response.

If any format error occurs in the response,

Then the Kernel shall terminate the processing by providing an ***End Application*** Outcome, see 4.5.7.

Else the Kernel checks the data duplication, see 4.2.4.4.

Requirement – Processing of READ RECORD Response

4.2.4.4 The Kernel shall check the data duplication[3] in response. Data Duplication could happen between one data object returned in Read Application Data and another returned in Application Initialization, or between two data objects returned in Read Application Data.

If data duplication occurs,

Then the Kernel shall terminate the processing by providing an ***End Application*** Outcome , see 4.5.7.

Else the Kernel continue with checking the application expiration date, see 4.2.4.5.

4.2.4.5 Once the Kernel retrieves the Application Expiration Date(“5F24”), it shall perform the followings:

If the current date obtained from the terminal is greater than the Application Expiration Date(“5F24”), meaning the application is expired. The Kernel shall do the following process:

If Card Transaction Qualifiers(“9F6C”), Byte 1 bit 4 value is “1”, meaning Go Online if Application Expires,

Then the Kernel performs the Online Process, see 3.2.5, and display "The application is expired, transaction is going online" on screen.

Else the Kernel shall decline the transaction by providing a ***Declined*** Outcome, see 4.5.4, and display "The application is expired, transaction declined" on screen.

Else the Kernel perform the next step, see 4.2.4.6

[3] The data objects with the same Tag value are considered as a duplication.

Requirement – Processing of READ RECORD Response

4.2.4.6 The Kernel shall see whether the current record is the last one to be read.

If the current record is the last record,

If exception file exists in the terminal,

Then the Kernel performs exception file check, see 4.2.4.7..

Else the Kernel continues with Offline Data Authentication, see 4.3.

Else the Kernel reads the next record.

Requirement – Exception File

4.2.4.7 This is an optional feature which is done only if the exception file exists in the terminal.

If the leftmost digits of PAN Number exists in the exception file[4],

Then the Kernel declines transaction by providing a ***Declined*** Outcome, see 4.5.4.

Requirement – Other Exception Handlings

4.2.4.8 **If** there exists any data object that in correct TLV format but undefined in this specification,

Then the Kernel shall store them in memory for further use instead of terminating the transaction.

[4] There is no requirement in this specification for an exception file.

Requirement – Other Exception Handlings

4.2.4.9 **If** any one or more of the followings occurs, the Kernel shall store the value and continue the transaction instead of terminating the transaction:

- The length of Cardholder Name (“5F20”) is not consistent with the requirements specified in *[EMV 4.3 Book3]* Table A1.
- The length of Cardholder Name Extension (“9F0B”) is not consistent with the requirements specified in *[EMV 4.3 Book3]* Table A1.
- Both Cardholder Name (“5F20”) and Cardholder Name Extension (“9F0B”) are returned.

4.3 Offline Data Authentication

The crucial requirement of a transaction over the contactless interface is the time that the cardholder has to hold the card in the field. Dynamic Data Authentication used in payment over the contactless interface is called fast Dynamic Data Authentication (fDDA) which can be performed without the card's presence in the field. This enables the cardholder to remove the card just after Read Application Data and guarantee the security of ICC transaction,

Only fDDA version 01 is supported in Kernel7.

4.3.1 Input

Table 4-6 lists the SDA-related data in the Kernel used for Offline Data Authentication.

Table 4-6 Offline Data Authentication--Terminal Data

Data element	Description
Public Key Index (PKI)	Identifies the Certificate Authority's public key in conjunction with the RID for use in offline static and dynamic data authentication..
CA Public Key	The Kernel uses CA Public Key to unlock the Issuer PK Certificate to recover the Issuer Public Key.
Registered Application Provider Identifier (RID)	Part of AID (first 5 bytes), used to identify payment systems. Identifies the application provider and the CA public key in conjunction with PKI.

Table 4-7 illustrates all Card data used for the Kernel to decide whether to perform SDA or fDDA.

Table 4-7 Offline Data Authentication--Card Data

Data element	Description
Application Interchange Profile (AIP)	Including indicators: • Byte 1 bit 7 indicates that the card supports SDA; • Byte 1 bit 6 indicates that the card supports fDDA.

Data element	Description
CA PKI	Used with the Registered Application Provider Identifier (RID) to identify which Private Key was used to encrypt the Issuer PK Certificate and which corresponding Public Key shall be used to recover the Issuer PK Certificate.
Issuer Public Key Certificate	Provided by the appropriate certification authority to the card issuer. When the Kernel verifies this data element, it authenticates the Issuer Public Key plus additional data.
Issuer Public Key Exponent	Provided by the issuer and used to retrieve signed static application data and ICC PK certificate.
Issuer Public Key Remainder	Includes the part of issuer public key which is not listed in Issuer public key certificate
Registered Application Provider Identifier (RID)	Part of AID (first 5 bytes), used to identify payment systems. Identifies the application provider and the CA public key in conjunction with PKI.
Signature Static Application Data (SAD)	A signature used in the validation of the card's static data. The SAD is signed with the Issuer Private Key a nd is placed on the card during the personalization process.
Static data authentication tag list	This is an optional data which contains the tag of the Application Interchange Profile (AIP) if it is to be signed. Tags other than the tag of the AIP shall not be present in the SDA Tag List. The AIP shall be included in the SDA Tag List if SDA, DDA, or CDA is supported. This data element is used for SDA, DDA and CDA.
ICC Dynamic Data	Data specified by Issuer and included in signed dynamic application data.
ICC Dynamic Number	ICC dynamic number is the first data element of the ICC dynamic data. The ICC dynamic number contains a time-variant generated by the ICC.
ICC PK Certificate	ICC PK Certificate is created using the Issuer Private Key and placed in the card during card personalization. ICC PK Certificate contains the ICC public key and a hash of static application data.
ICC PK Exponent	Used to recover the Signed Dynamic Application Data, with the value of 3 or 65537.

Data element	Description
IC Card Public Key Remainder	Part of the ICC PK which is not contained in the ICC PK Certificate (if any)
Signed Dynamic Application Data	The Signature generated by the card on receiving GPO command.

4.3.2 Kernel Processing

Requirement – fDDA Version Check

4.3.2.1 The Kernel shall support fDDA.

If Kernel7 supports fDDA,

Then the Kernel shall check the fDDA version in the Card. See 4.3.2.2;

Else fDDA verification failed, and the Kernel continues with the subsequent procedure in case that fDDA failed or not performed, see 4.3.2.5.

4.3.2.2 The card shall support fDDA.

If the Application Interchange Profile (AIP) indicates that the card supports DDA (AIP Byte 1 bit 6 is "1"),

Then the Kernel performs fDDA data check. See 4.3.2.3;

Else fDDA verification failed, and the Kernel continues with the subsequent procedure in case that fDDA failed or not performed, see 4.3.2.5.

Requirement – fDDA Data Check

4.3.2.3 The Kernel is responsible to ensure that all fDDA related data exist.

Table 4-8 illustrates all the dynamic Kernel data to be hashed.

Card Authentication Related Data includes a card unpredictable number and Card Transaction Qualifiers. As one of the Terminal Dynamic Data elements, the card shall generate a card unpredictable number and pad the Card Transaction Qualifiers into Card Authentication Related Data.

Note: If the Card Transaction Qualifiers is not placed in the card during personalization, then the card shall set the value as zero for use in Card Authentication Related Data.

Table 4-9 illustrates the data elements in ICC Dynamic Data.

If any data element listed in Table 4-8 and Table 4-9 is missing,

Then fDDA verification failed, and the Kernel continues with the subsequent procedure in case that fDDA failed or not performed, see 4.3.2.5.

Else, the Kernel performs fDDA verification, see 4.3.2.4.

Table 4-8 Dynamic Kernel Data to be Hashed

Data element	Tag	Length	Source
Unpredictable number	9F37	4 bytes	Kernel
Authorized amount	9F02	6 bytes	Kernel
Transaction currency code	5F2A	2 bytes	Kernel
Card Verification Related Data	9F69	Changeable	Card

Note: Card verification related data is variable-length data. Reader shall perform dynamic signature authentication using the entire card verification related data returned by card.

Table 4-9 IC Card Dynamic Data to be Hashed

Tag	Data element	Length	Source
9F36	Application Transaction Calculator (ATC)	2 bytes	Card

Requirement – fDDA Verification

4.3.2.4 The Kernel performs fDDA verification.

1. Obtain CA public key, this process conforms to Chapter 6.2 of *[EMV Book2]*.

2. Recover Issuer public key, this process conforms to Chapter 6.3 of EMV Book2.

3. Recover IC card public key, this process conforms to Chapter 6.4 of *[EMV Book2]*.

4. Verify dynamic signature, this process conforms to Chapter 6.5 of *[EMV Book2]*, except the following contents:

If the Card Verification Related Data (Tag "9F69") is returned,

And the length of the Card Verification Related Data (Tag "9F69") is more than or equal to 8 bytes and less than or equal to 16 bytes,

And the first byte of the Card Verification Related Data (Tag "9F69") is “01”,

then the Kernel continues with the following steps in fDDA verification;

Else fDDA verification failed, and the Kernel continues with the subsequent procedure in case that fDDA failed or not performed, see 4.3.2.5.

Dynamic Kernel data elements to be hashed shall not be specified in DDOL (DDOL is an unidentifiable data in the Kernel), instead they are coded in sequence as specified in Table 4-8.

If the card returns DDOL,

Then the Kernel ignores the data object, and shall not terminate the

Requirement – fDDA Verification

transaction.

If fDDA verification is successful,

Then the Kernel offline approve the transaction by providing an ***Approved*** Outcome, see 4.5.1.

Else fDDA verification failed, and the Kernel continues with the subsequent procedure in case that fDDA failed or not performed, see 4.3.2.5.

Note：In online transactions with offline data authorisation, the special terminals may perform specific actions based on the results of fDDA verification on online processing; this is out of the scope of this document.

Requirement – fDDA Failed or Not Performed

4.3.2.5 The issuer may request an Online processing if fDDA failed or not performed besides terminating the transaction. The Kernel needs to check with issuer's intention on the further process.

If the Card Transaction Quality Byte 1 bit 6='1' (Go Online if offline data authentication fails and terminal is online capable)

And the TTQ Byte 1 bit 4 = '0', indicating the terminal is online capable,

Then the Kernel goes for Online Processing, see 3.2.5 and the terminal shall notify cardholder that the transaction is processing as well as generate online message.

Else if the Card Transaction Quality Byte 1 bit 5 ='1' (Terminate Transaction and swith interface if full transaction flow in contact interface supported.)

And the TTQ Byte 1 bit 5 = '1' (Full transaction flow in contact interface Supported)

Then the Kernel shall terminate the transaction by providing a ***Try Another Interface*** Outcome.

Else, fDDA verification fails, the Kernel declines transaction by providing a ***Declined*** Outcome, see 4.5.4.

Note：In online transactions with offline data authorisation, the special terminals may perform specific actions based on the results of fDDA verification on online processing; this is out of the scope of this document.

4.4 Cardholder Verification

The kernel determines if a Cardholder Verification Method (CVM) is to be performed. The CVMs that may be supported for Kernel 7 are Online PIN, Consumer Device CVM, and Signature.

Note: A Consumer Device CVM is a CVM performed on, and validated by, the consumer's payment device, independent of the reader.

4.4.1 General Requirements

Terminal Implementation Requirements: With the exception of ATMs, Cardholder Verification shall be implemented for Kernel7. Note: ATMs may need to support an appropriate minimum level of cardholder verification, as determined by the payment system or local law, regardless of the CVMs supported by the card. As a consequence, ATMs are not subject to the Cardholder Verification processing requirements of this specification.

Acquirer-Merchant Configure Requirements: The acquirer-merchant shall be able to enable and disable the supported CVMs. However, support for the Consumer Device CVM shall be enabled (TTQ byte 3 bit 7 is 1b).

4.4.2 CVM Processing

4.4.2.1 CTQ not returned by card

Requirement – CVM Check

The kernel checks CVM for the transaction

If the kernel requires a CVM, and the payment application does not return the Card Transaction Qualifiers (CTQ, Tag '9F6C'),

Then the kernel shall

If the reader supports Signature

Then the kernel shall request a signature in the Outcome.

If the reader supports only the Consumer Device CVM and Online PIN

Then the kernel shall request Online PIN in the Outcome.

If the reader supports only the Consumer Device CVM,

Then the kernel shall provide ***Declined*** Outcome with *CVM* parameter set to *N/A*

4.4.2.2 CTQ returned by card

Requirement – CVM Check

If the card returns Card Transaction Qualifiers (“9F6C”), the kernel shall examine the CTQ to determine the CVM to be performed:

If Online PIN Required by card (CTQ byte 1 bit 8 is 1)and Online PIN supported by reader, then the kernel shall:

Then the Kernel shall provide an ***Online Request*** Outcome with *CVM* parameter set to *Online PIN.*

Else if (Online PIN not required or not supported) **and** Consumer Device CVM Performed by card (CTQ byte 2 bit 8 is 1),

If the Card Authentication Related Data(“9F69”) was returned during the transaction, **then**

If Card Authentication Related Data bytes 6-7 match CTQ bytes 1-2 (respectively),

Then the kernel shall set the CVM parameter in the Outcome to Confirmation Code Verified.

Else the kernel shall provide ***Declined*** Outcome with *CVM* parameter set to *N/A*;

Else (the Card Authentication Related Data(“9F69”) was not returned during the transaction),**then**

If the cryptogram type is an ARQC,

Then the kernel shall provide an ***Online Request*** Outcome with *CVM* parameter set to Confirmation Code Verified.

Else the kernel shall provide ***Declined*** Outcome with *CVM* parameter set to *N/A*;

Else if Signature Required (CTQ byte 1 bit 7 is 1) **and** the reader supports Signature (TTQ byte 1 bit 2 is 1),

Then the kernel shall set the CVM parameter in the Outcome to Obtain Signature.

Else (No CVM is indicated in the CTQ)

If the reader requires a CVM

Then the kernel shall set the Decline Required by Reader Indicator to 1,

Else the kernel shall set the *CVM* parameter set to *N/A*;

4.5 Outcome

Outcomes are the important information transferred from the Kernel to Entry Point, with parameters set, instructing the terminal for further processing. An Outcome indicates the Kernel's transaction disposition, and the parameters are additional instructions for terminal. The parameters include information to be displayed on the screen, clearing data for online transactions and Approved transactions, cardholder verification method, power supply for the field, etc.

The value and meaning of Outcomes and parameters can be found in Chapter6 of *[EMV Contactless BookA].*

This chapter describes specific Outcomes and corresponding parameters used in Kernel7.

4.5.1 Approved

If the transaction is approved offline, then Kernel returns an ***Approved*** Outcome.

Requirement – APPROVED Outcome

4.5.1.1 The kernel shall provide an ***Approved*** Outcome with the following parameters:

- **Start:** N/A
- **Online Response Data:** N/A
- CVM:
  - If the transaction amount is less than the CVM limit, then No CVM;
  - If transaction amount is greater than or equal to the CVM limit, then *Obtain Signature* or *Online PIN respectively*.
- UI Request on Outcome Present: Yes
  - o Message Identifier: '03' ("Approved")
  - o Status: Card Read Successfully
  - o [5](Value Qualifier: "Balance"

[5]If the Available Offline Spending Amount is returned, this will be displayed in Value Qualifier. This applies to all Outcomes.

Requirement – APPROVED Outcome

o Value: Available Offline Transaction Amount('9F5D')

o Currency Code: Transaction Currency Code)

- **UI Request on Restart Present:** No
- **Data Record Present:** Yes[6]
- **Discretionary Data Present:** No
- **Alternate Interface Preference:** N/A
- **Receipt:** Yes[7]
- **Field Off Request:** N/A
- **Removal Timeout:** zero

The Kernel shall provide clearing data in the Approved outcome, see Annex C for the data elements:

4.5.2 Online Request

If it is an online transaction, then the Kernel returns an *Online Request* Outcome.

[6]If Data Record Present is "Yes", then Kernel shall provide clearing data. See Annex C for clearing data. This applies to all Outcomes.

[7]If card returns Available Offline Spending Amount, the value shall be printed on the receipt. This applies to all Outcomes.

Requirement – ONLINE REQUEST Outcome

4.5.2.1 The kernel shall provide an ***Online Request*** Outcome with the following parameters:

- **Start:** N/A
- **Online Response Data:** N/A
- **CVM:**
  - If the transaction amount is less than the CVM limit, then No CVM;
  - If transaction amount is greater than or equal to the CVM limit, then *Obtain Signature* or *Online PIN respectively*.
- **UI Request on Outcome Present:** Yes

  o Message Identifier: '1B' ("Authorizing, Please Wait")

  o Status: Card Read Successfully

  o (Value Qualifier: "Balance"

  o Value: Available Offline Transaction Amount('9F5D')

  o Currency Code: Transaction Currency Code)
- **UI Request on Restart Present:** No
- **Data Record Present:** Yes
- **Discretionary Data Present:** No
- **Alternate Interface Preference:** N/A
- **Receipt:** N/A
- **Field Off Request:** N/A
- **Removal Timeout:** zero

The terminal goes online instructed by Entry Point on receiving ***Online Request*** Outcome.

On receiving the response from the host, the Kernel will not be invoked again. Instead, the terminal will process the transaction disposition based on the online authorization to complete the transaction and display the corresponding information to the cardholder.

If the Kernel requests an online transaction and the terminal failed to go online, then the terminal declines transaction. This won't restart the Kernel, however the terminal will interact with the cardholder on the transaction disposition.

If the final disposition is Approved, the terminal will use data transmitted from the Kernel in Outcome parameters to prepare clearing document, clearing data is shown in Annex C:

4.5.3 Try again (1)

If Kernel requests the cardholder to re-present the card in any case, the Kernel returns a ***Try Again*** Outcome.

Requirement – TRY AGAIN Outcome

4.5.3.1 The kernel shall provide a ***Try Again*** Outcome with the following parameters:

- **Start:** B
- **Online Response Data:** N/A
- **CVM:** N/A
- **UI Request on Outcome Present:** Yes
  - Message Identifier: '21' ("Present Card Again ")
  - Status: Processing Error
  - Hold Time: 13
  - Language Preference: 'en'
- **UI Request on Restart Present:** Yes
  - Status: Ready to Read
- **Data Record Present:** No
- **Discretionary Data Present:** No
- **Alternate Interface Preference:** N/A
- **Receipt:** No
- **Field Off Request:** 13
- **Removal Timeout:** zero

The cardholder is requested to present the card again if the Kernel provides a ***Try Again*** Outcome.

4.5.4 Declined

If transaction is declined offline, then the Kernel returns a ***Decline*** Outcome:

Requirement – DECLINED Outcome

4.5.4.1 The kernel shall provide a ***Declined*** Outcome with the following parameters:

- **Start:** N/A
- **Online Response Data:** N/A
- **CVM:** N/A
- **UI Request on Outcome Present:** Yes
  - Message Identifier: '07' ("Not Authorised")
  - Status: Card Read Successfully
- **UI Request on Restart Present:** No
- **Data Record Present:** No
- **Discretionary Data Present:** No
- **Alternate Interface Preference:** N/A
- **Receipt:** No
- **Field Off Request:** N/A
- **Removal Timeout:** zero

4.5.5 Try Another Interface

If the Kernel requests to try another interface, is shall return a ***Try Another Interface*** Outcome.

Requirement – TRY ANOTHER INTERFACE Outcome

4.5.5.1 The kernel shall provide a ***Try Another Interface*** Outcome with the following parameters:

- **Start:** N/A
- **Online Response Data:** N/A
- **CVM:** N/A
- **UI Request on Outcome Present:** Yes
  - Message Identifier: '18' ("Please insert or swipe card")
  - Status: Ready to Read
- **UI Request on Restart Present:** No
- **Data Record Present:** No
- **Discretionary Data Present:** No
- **Alternate Interface Preference:** Can be Contact Chip or Magstripe based on terminal capabilities.
- **Receipt:** N/A
- **Field Off Request:** N/A
- **Removal Timeout:** zero

4.5.6 Select Next

In any case that there is exception which requires go back to application selection for another application, the Kernel shall return a ***Select Next*** Outcome to start another combination selection.

Requirement – SELECT NEXT Outcome

4.5.6.1 The kernel shall provide a ***Select Next*** Outcome with the following parameters:

- **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:** zero

4.5.7 End Application

If the transaction is to be terminated, the Kernel returns an ***End Application*** Outcome, with the parameter *Start* to *N/A*.

Requirement – END APPLICATION Outcome

4.5.7.1 The kernel shall provide a ***Select Next*** Outcome with the following parameters:

- **Start**: N/A
- **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:** zero

4.5.8 Try again (2)

If the kernel receives SW1 SW2 = '6986' in response to the GPO command then the kernel shall provide a Try Again Outcome with the following parameters.

Requirement – TRY AGAIN Outcome

4.5.8.1 The kernel shall provide a ***Try Again*** Outcome with the following parameters:

- **Start:** B
- **Online Response Data:** N/A
- **CVM:** N/A
- **UI Request on Outcome Present:** Yes
  - Message Identifier: '20' ("See your mobile device for instructions")
  - Status: Processing Error
  - Hold Time: 132
  - Language Preference: 'en'
- **UI Request on Restart Present:** Yes
  - Status: Ready to Read
- **Data Record Present:** No
- **Discretionary Data Present:** No
- **Alternate Interface Preference:** N/A
- **Receipt:** No
- **Field Off Request:** 132
- **Removal Timeout:** zero

Annex A Data Dictionary

This annex lists all the data elements used in Kernel7's processing. The column definition can be found below.

Name
Name of this data element;

Format Tag Length

- Data element's format complies with *[EMV Book3]* Annex B;
- Data element's tag is in sexadecima;
- Data element's length is decimal indicating the length of value field.

Requirement

The Requirement column represents whether the data element's existence in the processing is **M**andatory, **C**onditional or **O**ptional, and indicates the source of the data element including the Kernel and the Card.

Retrieval

Retrieval column indicates the capability of the Kernel to retrieve the data in the card. If the Kernel is capable to retrieve the data, the table shall list the command to be used.

Value

The definition of the data value.

Table A-1 Data Dictionary

Name	Format Tag Length	Requirement	Description	Retrieval	Value
Available Offline Spending Amount	F: n 12 T: "9F5D" L: 6	S: Card R: Optional	Only if the Available Offline Spending Amount ("9F5D") is personalized to 1, the card returns this data in GPO response. Meanwhile, the Card Additional Processing (Byte 1 bit 1) shall be personalised to 1 to indicate that this amount will be calculated and included in all contactless transactions. Personalization of Available Offline Spending Amount to 1 also means that this data can be retrieved by GET DATA command.	GET DATA GPO READ RECORD	If the personalized value is greater than zero, GET DATA command is allowed to use retrieve the Available Offline Spending Amount; If this data element is personalized to "1" and Card Application Process Byte 1 bit 1 is "1", Available Offline Spending Amount is returned in GPO and READ RECORD is allowed. If the length of ICC private key is more than 1024 bits, Available Offline Spending Amount is retrieved via READ RECORD command instead of GPO.

Name	Format Tag Length	Requirement	Description	Retrieval	Value
Card Transaction Qualifiers	F: b 16 T: “9F6C” L: 2	S: Card R: Conditional If CVM supported or Card Transaction Qualifiers perference is supported.	The requirement used to indicate to device the CVM, card capabilities and Issuer requested by card.	GPO	Byte 1 Bit 8 1= Request Online PIN Bit 7 1= Request Signature Bit 6 1= If Offline Data Authentication fails and terminal is online-capable, go online Bit 5 1= If Offline Data Authentication fails and terminal supports standard debit/credit procedures, terminate the transaction. Bit 4 1= If application is expired, go online Bit 3~1= RFU Byte 2 Bit8:1=Consumer Device CVM Performed Note: Bit 8 is not used by cards compliant to this specification, and is set to 0b Bit 7-1= RFU

Name	Format Tag Length	Requirement	Description	Retrieval	Value
Application Interchange Profile (AIP)	F: b 16 T: “82” L: 2	S: Card R: Mandatory	Describes the card capability	GPO	Byte 1 Bit 8 RFU Bit 7 1= Support SDA Bit 6 1= Support DDA Bit 5 1= Support cardholder verification method Bit 4 1= Support terminal risk management Bit 3 1= Support Issuer verification Bit 2 1= RFU Bit 1 1= Support CDA Byte 2 Bit 8 = 01 Bit 7~1 RFU
Terminal Transaction Qualifiers	F: b 32 T: “9F66” L: 4	S: Kernel R: Mandatory	Indicates terminal capabilities, requirements and results of Pre-processing	N/A	See Table 3-1 Terminal Transaction Qualifiers (Tag “9F66”)

Name	Format Tag Length	Requirement	Description	Retrieval	Value
Card Authentication Related Data	F: b T: “9F69” L: var 8-16	S:Card R:Mandatory	Shall be returned in the last record retrieval.	READ RECORD GPO	Byte 1: fDDA Version No. (“01” in this version of specification) Byte 2-5: Card Unpredictable Number Byte 6-7: Card Transaction Qualifiers Byte 8: RFU(00) implementation is out of this specification Note: Card Verification Related Data is a 8-byte data and is reserved in the card during personalization.
Issuer Application Data	F: b T: “9F10” L: var. up to 32	S:Card R:Mandatory	Included in Issuer Application Data	GPO	
Application Currency Code	F: n 3 T: “9F5D” L: 2	S:Card R:Mandatory	Included in Issuer Application Data	GET DATA	
Application Transaction Counter	F: b 16 T: “9F36” L: 2	S:Card R:Mandatory		GET DATA GPO	

Annex B Fast Dynamic Data Authentication (fDDA)

There are Version 00 and Version 01 and only Version 01 is supported in Kernel7. A fDDA V01 – supported card shall contain Card Verification Related Data (Tag “9F69”), in which the Byte 1 indicating the lastest version of fDDA supported by card. If byte 1 of Card Verification Related Data (Tag “9F69”) specifies the version supported by card is 00, the Kernel shall treat the fDDA verification as a failure followed by the process in 4.3.2.5.

Dynamic data signature includes data from both the Kernel and the card. The interaction starts from the combination selection in Pre-processing during the period the Card returns PDOL containing the Kernel data to be signed in Dynamic Data Generation. PDOL is transferred from the Entry Point to the Kernel for reference on data to be passed to the card in GPO command. The data listed in PDOl include but not limited to Terminal Unpredictable Number, Authorized Amount and Transaction Currency Code. The Card uses the data to generate a Signature to be returned with other data in GPO response to the Kernel. The Kernel starts fDDA verification once the last record is read.

Figure B-1 describes a sample flow of fDDA-related data exchange.

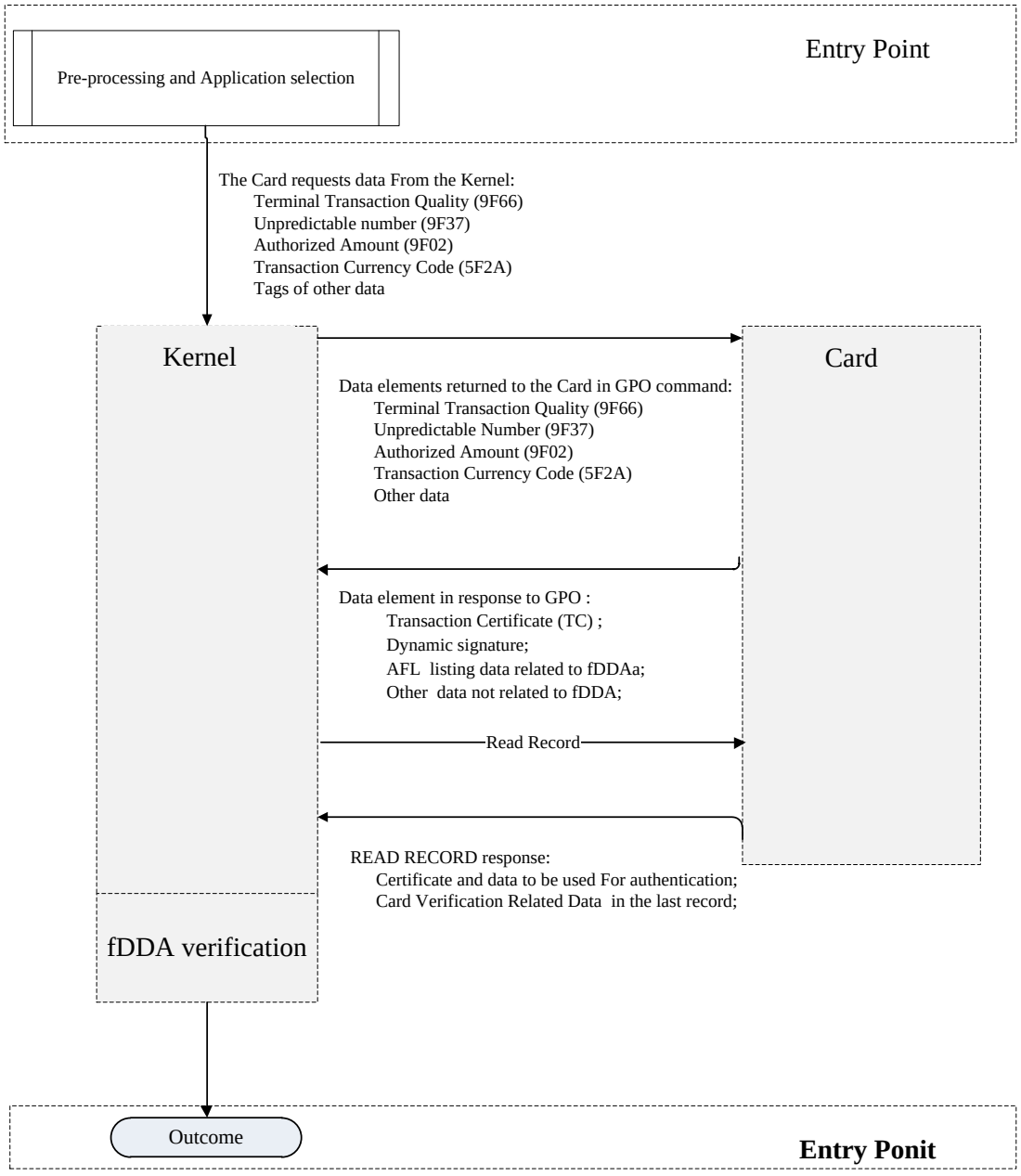

Figure B-1 Sample of Data Exchange Related to fDDA

Annex C Data Elements

Table C-1 lists the data elements provided by the Kernel in Outcomes for online messages and clearing records. The terminal shall be able to assemble the data elements into appropriet messages for the acquirers.

The network message is requested in Online transactions and offline approved transactions, however the data elements required can be different which is defined in the *Authorization Mode* column. The value of the column can be *Offline* and *Online* standing for Offline approved transactions and Online transactions respectively.

Table C-1 Clearing Data Element

Data element	Tag	Source	Authorization Mode	Requirement
Amount, Authorised (Numeric)	*'9F02'*	*Terminal*	*Offline and online*	Mandatory
Amount, Other (Numeric)	*'9F03'*	*Terminal*	*Offline and online*	Mandatory
Application Cryptogram (AC)	*'9F26'*	*Card*	*Offline and online*	Mandatory
Application Interchange Profile (AIP)	*'82'*	*Card*	*Offline and online*	Mandatory
Application PAN	*'5A'*	*Card*	*Offline and online*	Mandatory
Application PAN Sequence Number	*'5F34'*	*Card*	*Offline and online*	Conditional, only if the card returns it back to kernel.

Data element	Tag	Source	Authorization Mode	Requirement
Application Transaction Counter (ATC)	*'9F36'*	*Card*	*Offline and online*	Mandatory
Cryptogram Information Data (CID)	*'9F27'*	*Card*	*Offline and online*	Mandatory
Issuer Application Data (IAD)	*'9F10'*	*Card*	*Offline and online*	Mandatory
Terminal Capabilities	*'9F33'*	*Terminal*	*Offline and online*	Mandatory
Terminal Country Code	*'9F1A'*	*Terminal*	*Offline and online*	Mandatory
Terminal Verification Results (TVR)	*'95'*	*Kernel*	*Offline and online*	'00 00 00 00 00'
Track 2 Equivalent Data	*'57'*	*Card*	*Online*	Mandatory
Transaction Currency Code	*'5F2A'*	*Terminal*	*Offline and online*	Mandatory
Transaction Date	*'9A'*	*Terminal*	*Offline and online*	Mandatory
Transaction Type	*'9C'*	*Terminal*	*Offline and online*	Mandatory
Unpredictable Number (UN)	*'9F37'*	*Terminal*	*Offline and online*	Mandatory
Product Identification Information	*'9F63'*	*Card*	*Offline and online*	Conditional, only if the card returns it back to kernel.
Application Version No.	*'9F09'*	*Terminal*	*Offline and online*	Optional

Data element	Tag	Source	Authorization Mode	Requirement
Track 1 Discretionary Data	*'9F1F'*	*Card*	*Online*	Conditional, only if the card returns it back to kernel.

Annex D Transaction Log Retrieval

Transaction Log retrieval is an optional feature.

Transaction Log is a fixed-length cyclic file tracking all transaction sucessfully commited by the card. The records in the file shall not contain the Application Elementary File (AEF) Data Template (tag '70'). The SFI of the Transaction Log and record number are specified in Log Entry data element (Tag “9F4D”). The SFI of Transaction Log shall range from 11 to 30. Transaction logging can be enabled or disabled in the card during personalization, if the feature is enable, information of all online transaction and offline approved transactions will be saved. Each transaction only generates one record in the log and the maximum number of the records in a log is 10.

If the card supports transaction logging, Log Entry (Tag “9F4D”) data element shall be included in Issuer Discretionary data (Template “BFOC”) (see Table 4-1).

To retrieve Transaction Log, the Kernel shall go through the following steps in sequence:

- Perform Application Selection and retrieve the Log Entry data element located in the FCI Issuer Discretionary Data. If the Log Entry data element is not present, the application does not support the Transaction Log function.
- Send GET DATA command to read the Log Format(Tag “9F4F”) which lists (in tag and length format) of data objects representing the logged data elements.
- Issue READ RECORD command to read the Transaction Log.

Log Format and Transaction Log remains accessible when the application is blocked. And the FCI of the selected application shall be returned in response to SELECT command during pre-processing for log retrieval.

Annex E Glossary

This is a glossary of terms and abbreviations used in this specification. For descriptions of data elements, see Annex A.

a	Alphabetic
AAC	Application Authentication Cryptogram
AC	Application Cryptogram
Acquirer	A financial institution that signs a merchant (or disburses currency to a cardholder in a cash disbursement) and directly or indirectly enters the resulting transaction into interchange.
AFL	Application File Locator
AID	Application Identifier
AIP	Application Interchange Profile
Application Cryptogram	Cryptogram returned by the card; one of the following cryptogram types: AAC Application Authentication Cryptogram ARQC Authorisation Request Cryptogram TC Transaction Certificate
Approved	A Final Outcome
ARQC	Authorisation Request Cryptogram
ATC	Application Transaction Counter
C	Conditional
Card	As used in these specifications, a consumer device supporting contactless transactions.
Cardholder	An individual to whom a card is issued or who is authorised to use that card.

Cardholder Verification Method (CVM)	A method used to confirm the identity of a cardholder.
CDOL	Card Risk Management Data Object List
CID	Cryptogram Information Data
CVM	Cardholder Verification Method
DDA	Dynamic Data Authentication
DDOL	Dynamic Data Authentication Data Object List
Declined	A Final Outcome
DOL	Data Object List
EMV®	A global standard for credit and debit payment cards based on chip card technology. The EMV Integrated Circuit Card Specifications for Payment Systems are developed and maintained by EMVCo.
EMV mode	An operating mode of the POS System that indicates that this particular acceptance environment and acceptance rules supports chip infrastructure. Especially indicate contactless payment utilising a full chip infrastructure carrying EMV minimum data.
EMVCo	EMVCo LLC is the organisation of payment systems that manages, maintains, and enhances the EMV specifications. EMVCo is currently operated by American Express, JCB, MasterCard, and Visa.
End Application	A Final Outcome
F	Format
fDDA	Fast DDA. Leverages DDA as defined in *[EMV 4.3]* specifications. Used in EMV mode transactions to allow the reader to issue READ RECORD commands to obtain Dynamic Data Authentication (DDA) related data from the card and perform the DDA calculations after the card has left the field.

Final Outcome	Result provided to the reader as a result of Entry Point processing the Outcome from the kernel, or provided directly by Entry Point under exception conditions.
GPO	GET PROCESSING OPTIONS command
IAD	Issuer Application Data
ICC	Integrated Circuit Card
Issuer	A financial institution that issues contactless cards or contactless payment applications that reside in consumer devices.
Kernel	The kernel contains interface routines, security and control functions, and logic to manage a set of commands and responses to retrieve the necessary data from a card to complete a transaction. The kernel processing covers the interaction with the card between the Final Combination Selection (excluded) and the Outcome Processing (excluded).
Kernel ID	Identifier to distinguish between different kernels that may be supported by the reader.
L	Length
M	Mandatory
n	Numeric
N/A	Not Applicable; a possible value for several Outcome and Final Outcome parameters
O	Optional
Online PIN	A method of PIN verification where the PIN entered by the cardholder into the terminal PIN pad is encrypted and included in the online authorisation request message sent to the issuer.
Online Request	A Final Outcome
Outcome	Result from the kernel processing, provided to Entry Point, or under exception conditions, result of Entry Point processing. In either case, a primary value with a parameter set.
PAN	Primary Account Number

PDOL	Processing Options Data Object List
PICC	Proximity IC Card
PIN	Personal Identification Number
POS	Point of Sale
Select Next	An Outcome
SDA	Static Data Authentication
SFI	Short File Identifier
T	Tag
TC	Transaction Certificate
Terminal	A component of the POS System; described in detail in Chapter 2.
TLV	Tag Length Value
Transaction	The reader-card interaction between the first presentment of the card and the decision on whether the transaction is approved or declined. If the transaction is authorised online, this may involve multiple presentments of the card on the reader.
Try Again	An Outcome
Try Another Interface	A Final Outcome
TVR	Terminal Verification Results
UN	Unpredictable Number
VLP	Visa Low-Value Payment

*** END OF DOCUMENT ***