MasterCard International

Integrated Circuit Card Application Specification

For Debit and Credit on Chip

Version 2.0

Version 2.0
Published November 1998

Table of Contents

USING THIS MANUAL

SECTION 1 FUNCTIONAL SPECIFICATION OF EMV '96, VERSION 3.1.1 SPECIFICATION FOR PAYMENT SYSTEMS TRANSACTIONS

Table of Contents

SECTION 2 SECURITY SPECIFICATION OF EMV '96, VERSION 3.1.1 ICC SPECIFICATION FOR PAYMENT SYSTEMS TRANSACTIONS

USING THIS MANUAL

PURPOSE OF THIS MANUAL

The *ICC Application Specification* is provided to the payment system members and the card manufacturers to assist in the development of chip card applications to support EMV compliant debit and credit products. The usage of this specification is not mandated to achieve Type Approval of the developed products.

Members are encouraged to contact MasterCard to discuss any implementation issues that are not addressed by this publication.

Any comments or questions regarding this publication should be addressed to:

chip_help@mastercard.com

INTENDED AUDIENCE

This manual is intended for MasterCard members planning debit/credit/ATM product implementation on chip, ICC application developers, and terminal vendors.

This publication assumes that the reader is familiar with the *EMV '96, Version 3.1.1* specifications (see "Related Publications," below).

Related Publications

The following MasterCard publications contain material directly related to the contents of this publication.

- *Minimum Card Requirements for Debit and Credit on Chip, Version 2.0*
- *EMV '96 Integrated Circuit Card Application Specification for Payment Systems, Version 3.1.1*
- *EMV '96 Integrated Circuit Card Specification for Payment Systems, Version 3.1.1*

ORGANIZATION OF THIS MANUAL

This manual is organized into sections presenting information in the following manner:

- **Table of Contents**
  A list of the manual's sections and subsections. Each entry references a section and page number.

- **Using this Manual**
  A description of this manual and its contents.

- **Functional Specification**
  This section describes how the necessary functionality described in *EMV'96* (version 3.1.1) should be managed in the ICC to be compliant with the MasterCard business rules (set out in the Minimum Card Requirements). This part is fully platform independent. Any MasterCard debit or credit application can be constructed using the different building blocks described in this part.

  This section contains:

  - The description of Payment and Post-Issuance commands
  - The internal card processing
  - The description of all ICC related specifications to perform all transaction types supported by business rules for chip cards

- **Security Specification**
  This section describes how the security should be managed in the ICC to be compliant with the MasterCard security requirements. Because security is often an end-to-end requirement, issues related to the terminal and the issuer host system are also addressed, where appropriate. The different security aspects considered are:

  - Static Data Authentication
  - Dynamic Data Authentication
  - Off-line PIN Encipherment
  - Application Cryptogram generation
  - Issuer Authentication

- Secure Messaging

Furthermore, this section contains the security mechanisms, cryptographic algorithms and key management used to realize the six topics specified above:

- Issuer derivation of a unique triple-DES ICC Master Key for generation of the Application Cryptogram and for Secure Messaging

- Issuer derivation of the Data Authentication Codes

- ICC and Issuer derivation of the ICC Dynamic Numbers

- **Data Specification**
  This section lists and describes the data elements, which are required to support financial transactions. These data elements include:

  - ICC related data objects defined in *EMV'96 Integrated Circuit Card Specification for Payment Systems* - Part 2 and *EMV'96 Integrated Circuit Card Application Specification for Payment Systems*

  - ICC internal data objects which are MasterCard and Europay proprietary

  - Authorization and clearing data elements

  This document does not describe ICC internal data objects that are Issuer proprietary.

- **Appendix A: Network Data Element Requirements**
  This section summarizes by message the list of new data elements that the acquirer is to pass to the issuer. Some of these data elements may not be provided/requested by the ICC or the issuer, however the acquirer should support them and pass them when present. Data Element 55 has been established for the chip data in the Authorization Messages.

- **Appendix B: Digital Signature Scheme Giving Message Recovery**
  This section describes the digital signature scheme giving message recovery using a hash function according to ISO/IEC CD 9796-2. The main features of the scheme are the following:

  - Adding of redundancy in the signature by applying a hash function to the data to be signed

  - Adding of a header and trailer byte to obtain a unique recovery procedure and to prevent certain attacks

- **Appendix C: Cryptographic Algorithms**
  This section explains the following cryptographic algorithms:

  - DES and Triple-DES

  - RSA/Rabin

  - SHA-1

- **Index**
  An alphabetical listing by topic, subject, or title where items can be located in this manual.

USING THE SECTIONS IN THIS MANUAL

You do not have to read the contents of any section in any particular order. The following elements used in the manual help you locate the information you need more quickly:

The hand points at important information that you need to know. The text is in **boldface type**. Be sure to read everything that is labeled this way.

ENTER — Terms in UPPER CASE within a paragraph are acronyms, keyboard function keys, or computer commands.

Procedures — Numbered steps describe the sequence of actions you must take to perform a task.

Notations — Values surrounded by single quotes are hexadecimal values. For example, a binary field that is one byte in length and has a value of zero would be represented as ‘00’.

ABBREVIATIONS

The following abbreviations are used in this specification:

a	Alpha
AAC	Application Authentication Cryptogram
ABF	Application Blocked Flag
AC	Application Cryptogram
ADF	Application Definition File
AEF	Application Elementary File
AFL	Application File Locator
AID	Application Identifier
AIP	Application Interchange Profile
an	Alphanumeric
ans	Alphanumeric Special
APDU	Application Protocol Data Unit
ARC	ARPC Response Code
ARPC	Authorization Response Cryptogram
ARQC	Authorization Request Cryptogram
ASN	Abstract Syntax Notation
ATC	Application Transaction Counter
ATM	Automated Teller Machine
b	Binary
BER	Basic Encoding Rules
BIN	BASE Identification Number
CA	Certification Authority
CAM	Card Authentication Method
CBC	Cipher Block Chaining
CCPS	Chip Card Payment Service
CDOL	Card Risk Management Data Object List
CD	Committee Draft
CIAC	Card Issuer Action Code
CID	Cryptogram Information Data
CLCD	Card Life Cycle Data
CLA	Class Byte of the Command Message
cn	Compressed Numeric

CRM	Card Risk Management
CSI	Card Status Information
CTVR	Card Terminal Verification Results
CVM	Cardholder Verification Method
CVC	Card Verification Code
CVR	Card Verification Results
CVV	Card Verification Value
CVM	Cardholder Verification Method
DAC	Data Authentication Code
DDF	Directory Definition File
DDOL	Dynamic Data Object List
DEA	Data Encryption Algorithm
DES	Data Encryption Standard
DES3	Triple DES
DF	Dedicated File
DIS	Draft International Standard
DK	Derivation (DEA) Key
EMV	Europay/MasterCard/Visa
ECB	Electronic Code Book
EPI	EUROPAY International
FCI	File Control Information
Hex.	Hexadecimal
HHMMSS	Hours, Minutes, Seconds
IAD	Issuer Application Data
IARC	Issuer Authentication Response Code
IC	Integrated Circuit
ICC	Integrated Circuit Card
ID	Identifier
IDN	ICC Dynamic Number
IEC	International Electrotechnical Commission
IFD	Interface Device
IMK	Issuer Master Keys
IMK_{DAC}	Issuer Master Keys for Data Authentication Code
INS	Instruction
ISO	International Organization for Standardization

Lc	Length command data
Lcm	Least Common Multiple
L_{DD}	Length of the ICC Dynamic Data
LATC	Last on-line Application Transaction Counter
LCOLL	Lower Consecutive Offline Limit
LRC	Longitudinal Redundancy Check
M	Mandatory
MAC	Message Authentication Code
MCI	MasterCard International
MF	Master File
MK	Master Key
MK_{AC}	ICC Master Key Application Cryptogram
MK_{IDN}	ICC Master Key for ICC Dynamic Number generation
MK_{SMI}	ICC Master Key for Secure Messaging for Integrity
MK_{SMC}	ICC Master Key for Secure Messaging for Confidentiality
n	Numeric
N_{CA}	Length of the Certification Authority Public Key Modulus
N_I	Length of the Issuer Public Key Modulus
N_{IC}	Length of the ICC Public Key Modulus
N_{PE}	ICC PIN Encipherment Public Key Modulus
O	Optional
OLCTA	Offline Cumulative Transaction Amount
P1	Parameter 1
P2	Parameter 2
PAN	Primary Account Number
P_{CA}	Certification Authority Public Key
P_I	Issuer Public Key
P_{IC}	ICC Public Key
PDOL	Processing Options Data Object List
PIN	Personal Identification Number
PIX	Proprietary Application Identifier Extension
POS	Point of Service
PSE	Payment System Environment
PTC	PIN Try Counter
PTL	PIN Try Limit

PVV	PIN Verification Value
RFU	Reserved for Future Use (see next table)
RID	Registered Application Provider Identifier
RIP	Reset Internal Parameters
ROM	Read-Only Memory
RSA	Rivest Shamir Adleman (a Public Key algorithm)
S_{CA}	Certification Authority Private Key
SHA	Secure Hash Algorithm
S_I	Issuer Private Key
S_{IC}	ICC Private Key
SK	Session Key
SKAC	Session Key Application Cryptogram
SFI	Short File Identifier
SM	Secure Messaging
SW1	Status byte 1
SW2	Status byte 2
T-DES	Triple DES
TC	Transaction Certificate
TDOL	Transaction Certificate Data Object List
TLV	Tag Length Value
TSI	Transaction Status Information
TVR	Terminal Verification Results
UCOLL	Upper Consecutive Off-Line Limit
UDK	Unique DEA Key
var.	Variable
YDDD	Year, Day "Y" = Rightmost digit of the year. "D" = Day of the year (1-366)
YYMMDD	Year, Month, Day

NOTATIONS

The following notations are used in this specification:

Notation	Meaning
'0' to '9' and 'A' to 'F'	16 hexadecimal digits
#	Number
[...]	Optional part
xx	Any value
n/a	not applicable

REVISIONS TO THIS MANUAL

MasterCard periodically will issue revisions to this document as enhancements and changes are implemented, or as corrections to the manual are required.

All changes to this document are distributed as either (1) a full replacement of the entire manual, or (2) a partial update with new pages and replacement pages to be inserted throughout the document. Each revision is issued with the revision date printed in the page footer (replacing the release date), along with the page number, to ensure that all changes are accurately documented.

When issuing a partial update, in addition to the new document revision date, each revised page is annotated with "revision markers" (vertical lines in the right margin) to indicate where deletions, additions, or changes occurred.

At times, revisions or additions to this document initially will be published in an *Operations Bulletin* or other bulletin. A revision announced in a MasterCard bulletin or release document is effective as of the date indicated in that publication.

RELATED INFORMATION

In addition to this *ICC Application Specification*, the following information is available for this product:

Contact the ISO Web site at **www.iso.ch** for more information.

MASTERCARD CONTACTS

For inquiries on programs and services, contact your regional Member Services representative or your representative in St. Louis at the following addresses and numbers:

Attn: (your Member Services representative)
MasterCard International Incorporated
12115 Lackland Road
St. Louis, MO 63146-4071
USA

Phone: 314-275-6100
Fax: 314-542-7192
Telex: 434800 *answerback:* 434800 ITAC UI
E-mail: firstname_lastname@mastercard.com

For a list of member representatives, please see the *Member Information Manual*, Headquarters and Regional Offices section.

COMMENTS AND SUGGESTIONS

MasterCard is listening...

Please take a moment to provide us with your feedback on the material and usefulness of the *ICC Application Specification* using the Comment Form found in the back of the manual.

Or send us your comments on this manual via e-mail:

☞ **publications@mastercard.com**

We continually strive to improve our publications. Your input will help us accomplish our goal of providing you with the information you need.

Table of Contents

SECTION 1 FUNCTIONAL SPECIFICATION OF EMV '96, VERSION 3.1.1 SPECIFICATION FOR PAYMENT SYSTEMS TRANSACTIONS

Table of Contents

1.1 OVERVIEW

When a card that contains an Integrated Circuit Card (ICC) is presented to a terminal, the terminal determines which applications are mutually supported and the appropriate application is selected. If a MasterCard application is selected, the terminal reads the data from the ICC and determines whether to perform static or dynamic data authentication.

If supported, the appropriate cardholder verification method is performed.

Both the terminal and ICC perform offline risk management (for example, floor limit checking, transaction velocity checking) to determine whether the transaction should be approved, transmitted online for authorization, or declined. If both the ICC and the terminal indicate that the transaction satisfied the criteria for offline authorization, the transaction may be approved offline, and the ICC generates a Transaction Certificate (TC). The TC is stored by acquirers for possible use in the resolution of any future cardholder disputes and/or issuer chargebacks.

If the criteria for offline authorization are not satisfied, the terminal transmits an online request to the issuer indicating why the transaction was transmitted online (if the terminal has online capability). During online processing, the issuer verifies that the ICC is genuine and the ICC may verify that the issuer is genuine. The issuer may choose to transmit cardholder data updates to the ICC in the response message. To successfully complete the transaction, the ICC generates a Transaction Certificate.

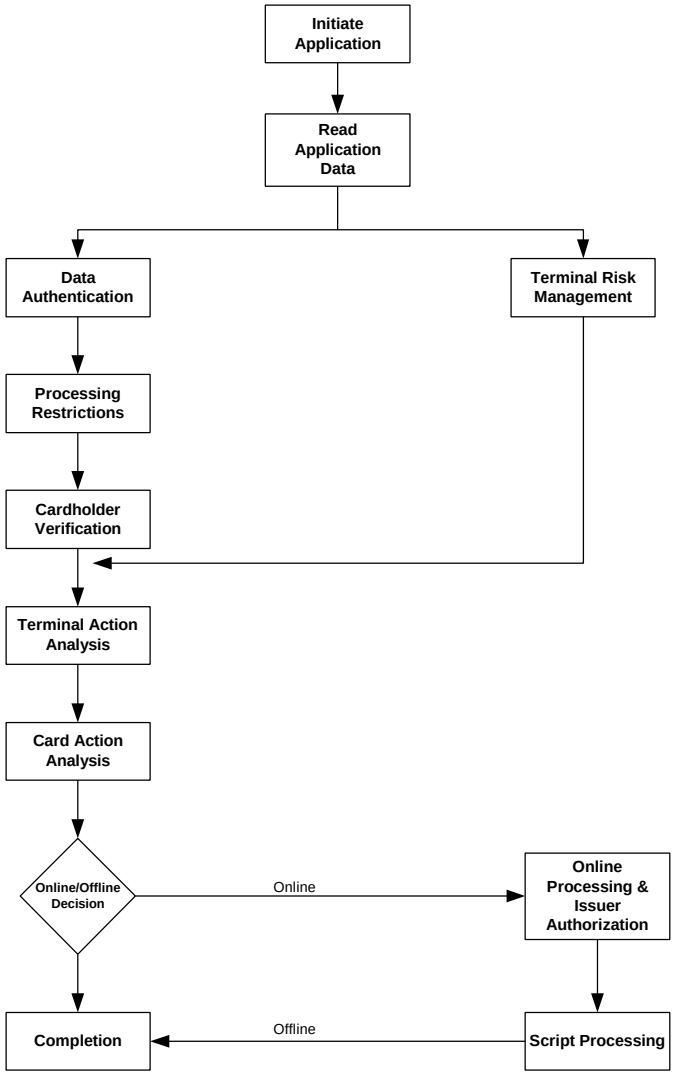
To decline a transaction, the ICC will generate, or be asked to generate by the terminal, an Application Authentication Cryptogram (AAC).

1.2 TRANSACTION FLOW

Exception handling shall be performed as described in the *EMV"96 ICC Specification for Payment Systems.*

An example of the transaction flow (without exceptions) is as:

FIGURE 1.1: TRANSACTION FLOW



1.2.1 Card Transaction Flow Flags

The following data elements are defined in ICC to manage the transaction flow. The "Sequence Flag" is a data object grouping flags or indicators managed in the ICC volatile memory to keep track of events in the current transaction. Table 1.1 lists the recommended number of indicators to be used.

TABLE 1.1: INTERNAL FLAGS—VOLATILE MEMORY ("SEQUENCE FLAG")

Flag name	Meaning
"PIN Verification Performed"	A VERIFY command has been performed during the current transaction
"PIN Verification Failed"	A PIN Verification has failed during the current transaction
"Issuer Authentication Performed"	Issuer Authentication has been performed
"Issuer Authentication successful"	Issuer Authentication has been performed successfully
"Script performed"	A script command has been performed during the current transaction
"Script failed"	A script command has failed during the script processing

The " Application Flag" is a data object grouping flags or indicators managed in the ICC non-volatile memory to keep track of events in previous transactions. Table 1.2 lists the recommended number of indicators to be used.

TABLE 1.2: INTERNAL FLAGS - NON-VOLATILE MEMORY ("APPLICATION FLAG")

Flag name	Meaning
Script Status Flag(SSF)	The setting of this flag to 1 indicates that the script processing is "pending", the setting of this flag to 0 indicates that no script processing is " pending" (the previous script processing has been normally terminated)
Application Blocked Flag (ABF)	The setting of this flag to 1 indicates that the selected Application is blocked
Last online transaction not completed	The ICC issued an ARQC on the 1st Generate AC but a 2nd Generate AC was not received
Issuer Authentication failure on last online authorization	Issuer Authentication Data was present but failed on the last online transaction.
Offline Static data authentication failed on last transaction	Offline Static CAM failed in a previous transaction
Offline Dynamic data authentication failed on last transaction	Offline Dynamic CAM failed in a previous transaction

1.3 STANDARD PAYMENT FUNCTIONS

1.3.1 Application Selection

Application selection is performed as described in the *EMV"96 ICC Specification for Payment Systems*:

The ICC has different options in supporting the application selection:

- It can support the Payment System Environment (PSE)
- It can support selection by partial name
- It must support direct selection by full name

Application selection is always the first application function performed. A successful execution of the function sets the path to the Application Definition File (ADF). Subsequent commands apply to Application Elementary Files (AEFs) associated to the selected ADF using Short File Identifiers (SFIs).

In response to the selection, the ICC returns the File Control Information (FCI) of the file.

1.3.1.1 Data Field Returned in the Response message

The data objects returned after the successful selection of the PSE, with file name "1PAY.SYS.DDF01," are described in Table 1.3.

TABLE 1.3: SELECT RESPONSE MESSAGE DATA FIELD (FCI) OF THE PSE

Name	Presence	Tag	Length	Format
FCI Template	M	'6F'	var. up to 252	var.
DF Name '1PAY.SYS.DDF01'	M	'84'	14	b
FCI Proprietary Template	M	'A5'	var.	var.
SFI of the directory elementary file	M	'88'	1	b
Language Preference	O	'5F2D'	2-8	an 2
Issuer Code Table Index	O	'9F11'	1	n 2
FCI Issuer Discretionary Data	O	'BF0C'	var. up to 222	var.

The data objects returned after the successful selection of an ADF are described in Table 1.4

TABLE 1.4: SELECT RESPONSE MESSAGE DATA FIELDS OF AN ADF

Value	Presence	Tag	Length	Format
FCI Template	M	'6F'	var. up to 252	var.
DF Name	M	'84'	5-16	b
FCI Proprietary Template	M	'A5'	var.	var..
Application Priority Indicator	O	'87'	1	b
Application Label	O	'50'	1-16	an 1-16
Language Preference	O	'5F2D'	2-8	an 2
Issuer Code Table Index	O	'9F11'	1	n 2
PDOL	O	'9F38'	var.	b
Application Preferred Name	O	'9F12'	1-16	an 1-16
FCI Issuer Discretionary Data	O	'BF0C'	var.	var.

For an ADF the Dedicated File (DF) Name and Application Identifiers (AIDs) are identical.

If any of the optional data elements above have length 0 (meaning they are empty), they are not present in the FCI.

1.3.1.2 Coding of Payment System Application Identifiers

Application Identifiers (AIDs) which conform to ISO/IEC 7816-5 standard uniquely identify applications. An AID consists of two parts:

- A Registered Application Provider Identifier (RID) of 5 bytes, unique to an application provider and assigned according to ISO/IEC 7816-5.
- An optional field assigned by the application provider of up to 11 bytes. This field is known as a Proprietary Application Identifier Extension (PIX) and may contain any 0 - 11 byte value specified by the provider. The meaning of this field is defined only for the specific RID; it needs not be unique across different RIDs.

Additional ADFs defined under the control of other application providers may be present in the ICC but must not duplicate the range of RIDs assigned to Payment Systems (which is an application provider according to ISO/IEC 7816-5). Compliance with ISO/IEC 7816-5 will assure this avoidance.

Refer to the Minimum Card Requirements for the MasterCard products (MasterCard, Cirrus).

1.3.1.3 Structure of the Payment Systems Environment

If present, the Structure of the PSE is compliant with *EMV '96 ICC Specification for Payment Systems*—Part 3.

The PSE begins with a Directory Definition File (DDF) called '1PAY.SYS.DDF01' which contains a Directory file referencing the Payment Applications which are stored within the ICC. The presence of the PSE '1PAY.SYS.DDF01' is optional.

1.3.1.4 Coding of a Payment System's Directory

A Payment System's Directory is a linear file identified by a SFI in the range 1 to 10. The SFI for the directory is contained in the FCI of the PSE.

Each entry in a Payment Systems Directory is an Application Template (tag '61') and contains the information according to Table 1.5:

TABLE 1.5: ADF DIRECTORY ENTRY FORMAT

Tag	Length	Value	Presence
'4F'	5-16	ADF Name (AID)	M
'50'	1-16	Application Label	M
'9F12'	1-16	Application Preferred Name	O
'73'	var.	Directory Discretionary Template	O

1.3.1.5 Application Selection with or without cardholder confirmation

The applications can be ranked by priority order according to Table 1.6:

The applications that are intended to be selected after the confirmation of the cardholder will have b8 set.

TABLE 1.6: CODING OF APPLICATION PRIORITY WITH OR WITHOUT CARDHOLDER CONFIRMATION

b8	b7-b5	b4-b1	Definition
0/1			0 = Application may be selected without confirmation of cardholder 1 = Application cannot be selected without confirmation of cardholder
	xxx		RESERVED
		0000	No priority assigned
		xxxx (except 0000)	Order in which the application is to be listed or selected, ranging from 1 to 15, with 1 being highest priority.

1.3.2 Initiate Application Processing

Initiate application processing is performed as described in the *EMV '96' ICC Specification for Payment Systems*.

In the Get Processing Options command data, the ICC may receive the PDOL which contains the terminal data object values as required in the response to the Select command. According to this PDOL data object values the ICC shall be able to return the appropriate Application File Locator (AFL) and Application Interchange Profile (AIP). The ICC can manage several sets of AIP/AFL and chooses the appropriate AIP and AFL depending on the PDOL data. An AFL defines the set of data records that the terminal must read.

In order to differentiate several behaviors regarding CVM list, CDOLs, IACs … depending on the terminal environment where the ICC is used, the ICC may contain several AFLs.

If a single couple (AIP, AFL) is defined for the application, an empty PDOL can be used (PDOL not present in the response to the Select command).

A single couple (AIP, AFL) may be used with a non empty PDOL to test additional constraints of the selected application (e.g. to rejecting a transaction based on non supported terminal capabilities).

The ATC shall be incremented during the Initiate application processing.

1.3.2.1 Optional Data Objects Supplied With The GET PROCESSING OPTIONS command

If the Processing Option Data Object List (PDOL) is used in the GET PROCESSING OPTIONS command, it is recommended that it includes one or more data objects defined Table 1.7:

TABLE 1.7: PDOL DATA

Value	Presence	Tag	Length
Terminal Type	O	'9F35'	1
Terminal Capabilities	O	'9F33'	3
Merchant Category Code	O	'9F15'	3
Transaction Type	O	'9C'	1
Terminal Country Code	O	'9F1A'	2
Transaction Category Code	O	'9F53'	1

1.3.2.2 AIP and AFL

The AIP and AFL information are defined at card personalization stage.

The AFL value is a list of objects defined in Table 1.8:

TABLE 1.8: AFL DESCRIPTION

	B8	b7	b6	b5	b4	b3	b2	b1	Meaning
Byte 1	x	x	x	x	x	0	0	0	coding of the SFI (five most significant bits)
Byte 2	x	x	x	x	x	x	x	x	first record number to be read for that SFI
Byte 3	x	x	x	x	x	x	x	x	last record number to be read for that SFI
Byte 4	x	x	x	x	x	x	x	x	number of records involved in offline data authentication

1.3.2.3 Mandatory Data Objects Supplied With The GET PROCESSING OPTIONS command

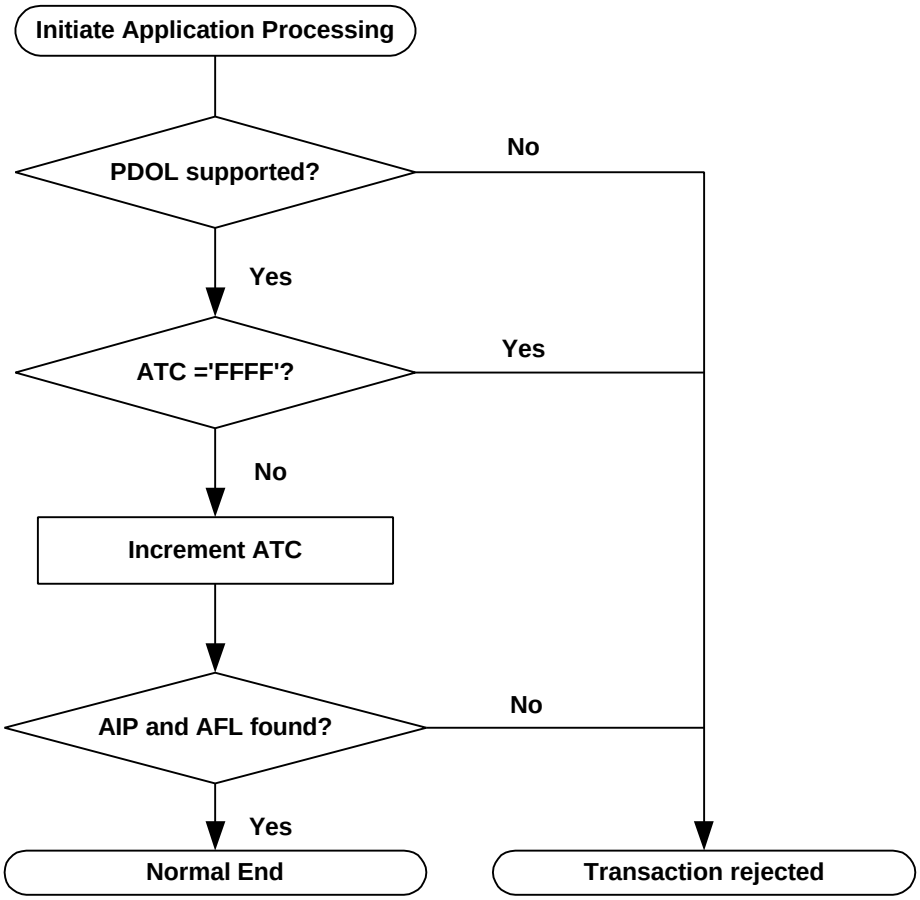
TABLE 1.9: DATA RETRIEVABLE BY GET PROCESSING OPTIONS COMMAND

Value	Presence	Tag	Length
Application Interchange Profile	M	'82'	2
Application File Locator	M	'94'	4-252

1.3.2.4 Initiate Application Processing flow

Figure 1.2 describes all the functions performed by the ICC during the Get Processing Options command.

FIGURE 1.2: INITIATE APPLICATION PROCESSING



1.3.2.4.1 PDOL supported?

The ICC verifies that it supports the PDOL data objects values provided in the Get Processing Options command. If one data object value is not supported, the transaction is rejected.

1.3.2.4.2 ATC='FFFF' ?

The ICC verifies that ATC equals 'FFFF'. If ATC equals 'FFFF', the transaction is rejected, else the ATC is incremented by 1.

1.3.2.4.3 AIP and AFL found?

The ICC searches for an appropriate AIP and an AFL corresponding to the PDOL data. If the couple (AIP, AFL) is not found the transaction is rejected; else the Initiate Application Processing is successful and the couple (AIP, AFL) is returned to the terminal.

1.3.3 Read Application Data

Read application data is performed as described in the *EMV"96 ICC Specification for Payment Systems*. Data Objects are read through the command READ RECORD according to the *EMV"96 ICC Specification for Payment Systems*.

The complete list of the Data Objects will be described in Section III and Appendix A of this specification.

1.3.4 Offline Card Authentication

Data authentication is performed by the terminal as described in the *EMV"96 ICC Specification for Payment Systems*. Using digital signatures based on public key techniques the terminal can verify the legitimacy of the card and of critical ICC data; the ICC contains certificates the terminal can verify.

1.3.4.1 Static offline card authentication

Data objects that are signed are "read only". Issuers may choose other data objects to be signed but such data must be static (i.e. never change) and for which the detection of fraudulent change is beneficial. With static offline card authentication the ICC doesn't play an active role.

1.3.4.2 Dynamic offline card authentication

For dynamic offline card authentication, the ICC plays an active role. As a response to the INTERNAL AUTHENTICATE command, it signs data the terminal can verify with the ICC public key.

1.3.5 Cardholder Verification

Cardholder verification is performed as described in the *EMV"96 ICC Specification for Payment Systems.*

The following Cardholder Verification Methods (CVMs) can be supported by the ICC application and may be included in the ICC applications CVM List:

- Paper signature
- No CVM required
- Online PIN
- Offline PIN (plain text or enciphered)

If offline plain text PIN is used, the VERIFY command initiates in the ICC the comparison of the Transaction PIN Data (tag '99') sent in the data field of the command with the Reference PIN stored in the ICC.

If offline enciphered PIN is used, the ICC first deciphers the enciphered PIN, before comparing it with the Reference PIN.

1.3.5.1 Reference PIN definition

The total length of the PIN may vary from 4 to 12 numeric digits. The initial length determined by the Issuer at the personalization stage of the ICC could be updated during the further life cycle of the ICC.

The PIN Try Limit (PTL), may vary from Issuer to Issuer. The PIN Try Counter (PTC) is linked to a single Reference PIN. The PTL value is determined by the Issuer during the personalization stage of the ICC, and is not changed during the further life cycle of the ICC. The plain text and enciphered PIN are the only supported offline PIN formats.

It must be impossible to extract the Reference PIN from the ICC.

For a PIN, the Offline PIN Verification status "PIN Verification Performed," "PIN Verification Failed" should be valid until a further ADF selection is performed.

1.3.5.2 Enciphered offline PIN definition

The complete process of PIN Encipherment by the terminal is described in Section 2. An ICC with dynamic RSA is required to perform the functionality:

The key involved to encipher the PIN by the terminal is the ICC PIN Encipherment Public key if present or the ICC Application Public key in the other case.

Unpredictable data are provided by the terminal and by the ICC, to complete the PIN data to be enciphered. The unpredictable data provided by the ICC are obtained by means of the command "GET CHALLENGE". The random number stays only valid for the next command, which should be a "VERIFY" command. If an error occurs or if the command is a command different from the "VERIFY" command, the random number is lost.

1.3.5.3 General processing

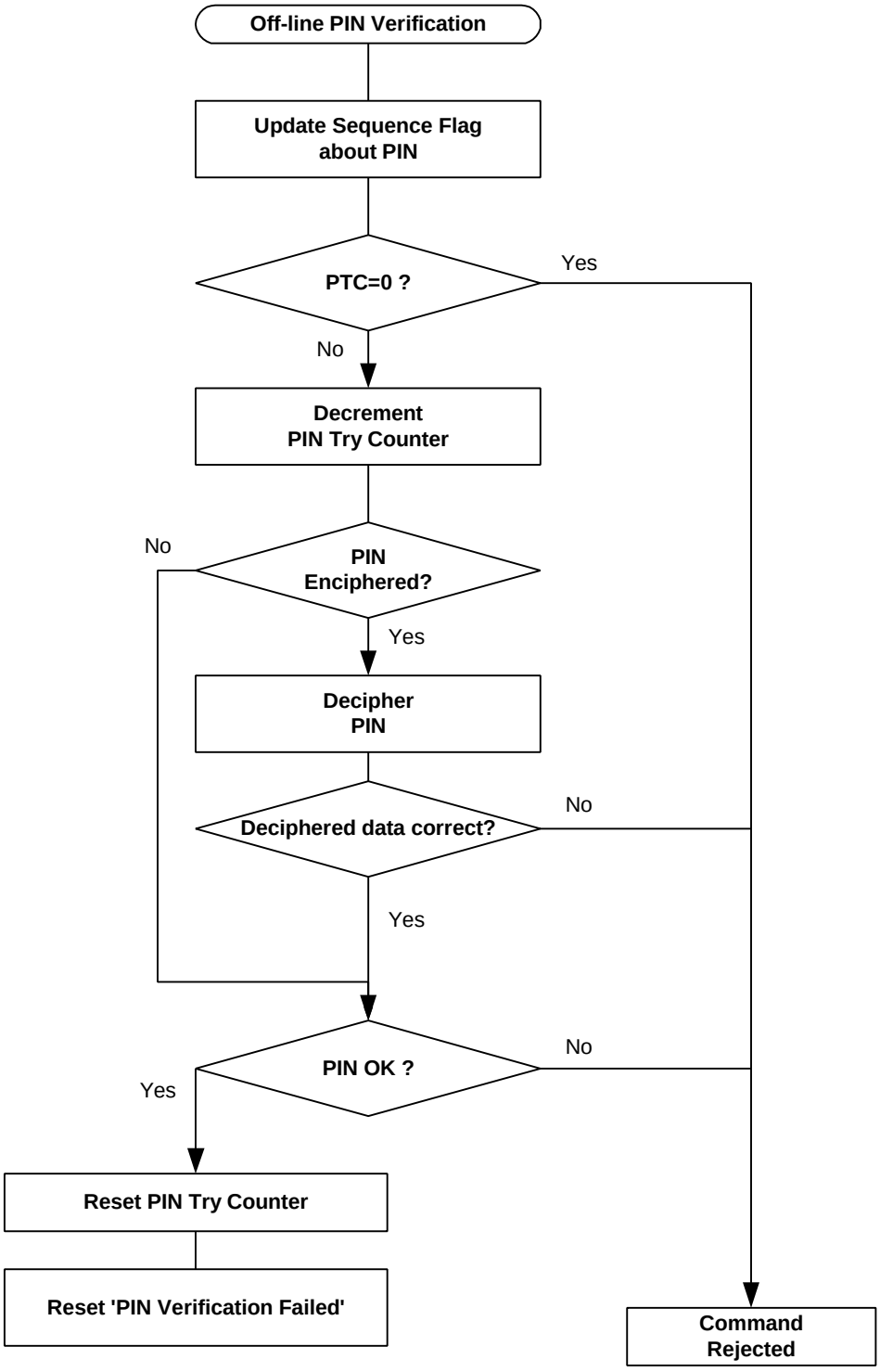
When the terminal determines that an offline PIN is to be used, the terminal should send a GET DATA command to the ICC to obtain the PTC. If the GET DATA command is supported by the ICC, the ICC replies with the PTC. If the PTC is "0", indicating no more PIN tries, the ICC does not allow offline PIN check.

The application decrements the PTC and then compares the Transaction and Reference PINs. If they match, the ICC indicates this in the response to the VERIFY command, and the ICC resets the PTC to the value of the PTL.

If the PTC was "0" before the beginning of the command, the ICC does not compare the Transaction and Reference PINs, it indicates in the response to the VERIFY command that the PTL is exceeded (the PIN is blocked).

If the PIN Verification has failed, the resulting value of the PTC is sent in the VERIFY response indicating the number of remaining PIN tries.

FIGURE 1.3: OFFLINE PIN VERIFICATION



1.3.5.4 Verify description

1.3.5.4.1 Update Sequence Flag about PIN

The ICC updates the PIN related flags in Sequence flag:

- The "PIN Verification Performed" bit is set
- The "PIN Verification Failed" bit is set

1.3.5.4.2 PTC = 0 ?

The ICC checks if the PTC has reached the value 0. In this case, the ICC cannot perform the PIN verification and returns an error code in the processing state SW1 SW2: '6983' Authentication method blocked.

1.3.5.4.3 Decrement PIN Try Counter (PTC)

Before comparing the Transaction PIN with the Reference PIN, the ICC decrements the PTC.

1.3.5.4.4 PIN Enciphered?

The ICC tests if the Transaction PIN data is sent in plain text or enciphered.

1.3.5.4.5 Decipher PIN

The ICC deciphers the Transaction PIN data with the ICC private key.

1.3.5.4.6 Deciphered data correct?

The ICC verifies the format of the deciphered data and its header (value '7F'), and checks with the challenge provided by ICC. In case the comparison fails, the command is rejected.

1.3.5.4.7 PIN OK?

The ICC verifies the PIN header and compares the Transaction PIN data sent in the data field of the command with the Reference PIN. In case the comparison fails, the ICC returns a warning in the processing state SW1 SW2: '63 Cx' (Verification Failed) where "x" designates the number of further retries allowed.

1.3.5.4.8 Reset PIN Try Counter

When the comparison of the Transaction PIN Data with the Reference PIN Data has been successful, the ICC resets the PTC to the PTL value.

1.3.5.4.9 Reset "PIN Verification Failed"

When the comparison of the Transaction PIN Data with the Reference PIN Data has been successful, the ICC resets the "PIN Verification Failed" flag in Sequence flag.

1.3.6 First GENERATE AC Processing

The GENERATE AC command is used to transmit transaction-related data objects from the terminal to the ICC and to demand an Application Cryptogram. The terminal demands one of three different types of Application Cryptogram.

1. Request for the generation of a TC:

   The ICC will respond with a Transaction Certificate (TC) when the terminal requests a TC and none of the results of the card risk management checks performed indicate that the ICC should respond with an Authorization Request Cryptogram (ARQC) or Application Authentication Cryptogram (AAC).

   Prior to responding with a TC, the ICC performs any necessary card risk management functions:

   - Set the bits in the Card Verification Results (CVR) to indicate that a TC was returned in the first GENERATE AC response and a second GENERATE AC command was not requested.
   - Increment the Cumulative Offline Transaction Amount by the value of the transaction amount where application currency is the same as transaction currency.

☞ **When the ICC responds with a TC or an AAC in response to the first GENERATE AC command, the terminal completes the transaction. The terminal should display a message to indicate the action taken (approved, declined).**

2. Request for the generation of an ARQC

   If the terminal requests an ARQC and the results of the card risk management allow the ICC to respond with an ARQC, the ICC responds with an ARQC. Upon receipt of the response, the terminal proceeds to online processing. Prior to responding with an ARQC, the ICC performs any necessary card risk management functions:

   - Set the bits in the CVR to indicate that an ARQC was returned in the first GENERATE AC response and a second GENERATE AC command was not yet requested
   - Set the "Last Online Transaction not completed" flag

   If the terminal requests an ARQC but the results of the card risk management do not allow the ICC to respond with an ARQC, the ICC responds with an AAC.

3. Request for the generation of an AAC

   The ICC will respond with an AAC when:

   - The terminal requests an AAC, or
   - The results of the card risk management checks indicate at least once that the ICC should respond with an AAC

   Prior to responding with an AAC, the ICC performs the following function:

   - Set the bits in the CVR to indicate that an AAC was returned in the first GENERATE AC response and a second GENERATE AC command was not requested;

The GENERATE AC command is coded as described in Part II of the *EMV '96 ICC Specification for Payment Systems*.

When a data object is referenced in CDOL1 or CDOL2, the length of the data objects is as specified for this specification.

When the "Amount, Authorized" and "Amount, Other" are referenced in CDOL1 and CDOL2 for input to the TC, AAC or ARQC, the tags and lengths for the numeric versions are used (tags "9F02" and "9F03"), not the tags and lengths for the binary versions.

The data field of the GENERATE AC command is not TLV encoded, so it is imperative for the ICC to know the format of this data when the command is received. This is achieved by having specified the format of the data to be included in the command, using the Card Risk Management Data Object List CDOL1 (tag '8C').

As a result of the GENERATE AC command, the ICC response consists of returning the following data objects to the terminal:

TABLE 1.10: DATA ELEMENTS SENT TO THE TERMINAL

Name	Tag	Length	Format
Cryptogram Information Data	'9F27'	1	b
ATC (Application Transaction Counter)	'9F36'	2	b
Application Cryptogram	'9F26'	8	b
Issuer Application Data	'9F10'	8-32[1]	b

1 Due to network restrictions in the U.K, the Issuer is only guaranteed to receive the first seven bytes of the Issuer Application Data in the authorization.

- **Cryptogram Information Data**: see Table 1.11.

TABLE 1.11: CRYPTOGRAM INFORMATION DATA

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0							AAC
0	1							TC
1	0							ARQC
1	1							Not used
		x	x					RESERVED
				0				No advice required
				1				Advice required
					x	x	x	Reason/advice/referral code
					0	0	0	No information given
					0	1	0	PIN try limit exceeded
					0	1	1	Issuer Authentication Failed
					x	x	x	Other values RESERVED

* An advice can be required (bit b4 = 1) in case the PIN is blocked (reason: PIN Try Limit exceeded) and the ICC does not demand an ARQC. During the First Generate AC the case "Issuer Authentication has failed" is not supported.

- **ATC:** This is the Application Transaction Counter stored in the ICC. It is incremented by one each time the GET PROCESSING OPTIONS command is executed. The ATC is used to generate the application cryptogram.

- **Application Cryptogram:** Section 2 describes in detail how the cryptogram is generated (using the appropriate ICC Master Key...), and which transaction data need to be taken into account. The result is a TC, an ARQC, or an AAC. The cryptogram returned by the ICC may differ from that requested by the terminal in the command message according to the internal ICC processing.

☞ *** Advice messages will not be supported across MasterCard networks, but may be supported under other circumstances.**

- **Issuer Application Data** contains the following data objects ordered as follows:
  - Key Derivation Index
  - Cryptogram Version Number
  - CVR: information about the ICC decisions taken during the Card Risk Management (CRM) processing
  - Data Authentication Code (DAC) or 2 leftmost bytes of ICC Dynamic Number: proving that the terminal correctly performed Static or Dynamic Data Authentication
  - Other data

1.3.6.1 Card Risk Management Data (CDOL1)

The Card Risk Management Data Object List (CDOL1) is an EMV ICC data object, which provides the terminal with a list of data objects required for the CRM processing and the certificate generation.

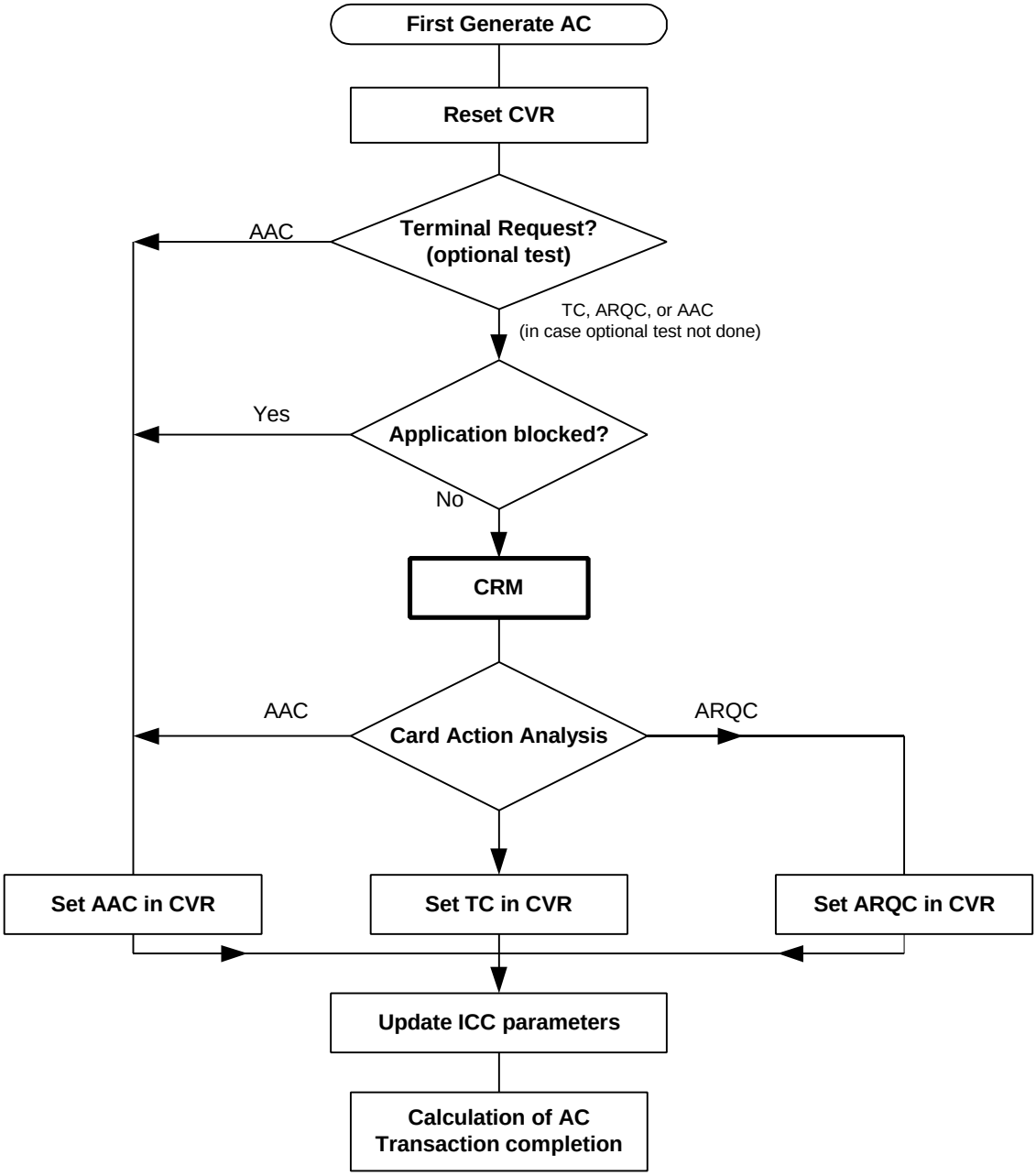
The CDOL1 is used with the first GENERATE AC command. The coding of CDOL1 is defined in Section 3.

The value of each data objects of the CDOL1 has to be passed from the terminal to the ICC within the data field of the first GENERATE AC command.

1.3.6.2 Internal Process

Figure 1.4 describes all the functions performed by the ICC during the first GENERATE AC command.

FIGURE 1.4: FIRST GENERATE AC OVERALL DIAGRAM



A detailed analysis of the Card Risk Management (CRM) function is described in section.1.3.6.3 CRM.

A detailed analysis of the Card Action Analysis function is described in section 1.3.6.4 Card Action Analysis.

A detailed analysis of the Update ICC Parameters function is described in section 1.3.6.5 Update ICC Parameters.

A detailed analysis of the Transaction Completion function is described in section 1.3.6.6 Transaction Completion/Online Processing.

1.3.6.2.1 Reset CVR (Card Verification Result)

The CVR are ICC internal registers storing information concerning the ICC functions performed during a Payment Transaction. The major ICC functions reflected in these registers are the PIN verification, the card risk management checks and the status of the previous transaction:

- Type of Application Cryptogram returned in 1st Generate AC (AAC, TC, ARQC)
- Type of Application Cryptogram returned in 2nd Generate AC (AAC, TC, 2nd Generate AC not requested)
- Issuer Authentication Failed
- Offline PIN Verification performed
- Offline PIN Verification failed
- Unable to go online
- Last online transaction not completed
- Pin try limit exceeded
- Exceeded velocity checking
- New card
- Issuer Authentication failure on last online transaction
- Issuer Authentication not performed after online authorization
- Application blocked by card because PIN Try Limit exceeded
- Offline Static data authentication failed on last transaction
- Number of script commands processed successfully
- Offline Dynamic data authentication failed on last transaction
- Issuer script processing failed on last or current transaction
- Lower consecutive Offline limit or Lower Cumulative Offline Transaction Amount Exceeded
- Upper consecutive Offline limit or Upper Cumulative Offline Transaction Amount Exceeded
- Maximum Offline Transaction Amount exceeded

The CVR is reset to '0' at the beginning of the first Generate AC.

1.3.6.3 CRM

This section describes the MasterCard recommended card risk management functions that are supported for international transactions and the setting of the bits in the CVR to record the outcome of the execution of the card risk management functions. Issuers can implement further non-EMV card risk management functions on the cards that they issue. As a general guideline, if the issuer uses EMV data elements and the risk management functionality is entirely within the ICC, such functionality should be supported for international transactions. If terminals are required to carry out any additional processing then that functionality may only be supported domestically. In all cases, implementers must seek MasterCard approval for issuer/country/region specific card risk management functionality before assuming anything other than local support.

The Issuer is free to pick among the different card risk management functions:

- PIN verification status
- Previous transaction status
- New card
- Maximum offline transaction Amount
- Velocity checking of Offline Consecutive transactions
- Offline cumulative amount

"Offline cumulative amount" and "Velocity checking of Offline Consecutive transactions" are strongly recommended as a minimum. Once the card risk management functions have been selected, those functions are to be performed as indicated in this specification.

☞ **Velocity checking is mandatory. It can be either performed by the terminal or by the card internally. The velocity checking described in this document concerns the velocity checking done by the card internally.**

To determine whether the ICC responds with a TC, ARQC, or AAC, the ICC may perform one or more of card risk management functions. Even if one of the card risk management functions determines that an AAC will be returned in the response to the GENERATE AC command, the ICC must proceed through all functions in card risk management prior to responding with an AAC. Since the ICC must perform all functions, the order in which the functions are performed need not conform to the order described in this section.

If a data object requested from the terminal is not available (the data object returned in the GENERATE AC command data field is zero filled), the ICC proceeds to the next function in card risk management.

1.3.6.3.1 PIN verification status

The ICC updates the CVR Offline PIN Verification bits:

- The CVR bit "Offline PIN Verification was performed" is set if a VERIFY command was received in the current transaction
- The CVR bit "Offline PIN Verification failed" is set if a VERIFY command has failed (PIN failed or blocked during the current transaction)
- The CVR bit "PIN Try Limit exceeded" is set if the PIN Try Counter became 0 before or during the current transaction

1.3.6.3.2 Previous transaction status

The ICC updates the previous transaction CVR bits using Application Flag and Script Counter:

- Last online transaction not completed
- Issuer Authentication failure on last online transaction
- Offline Static Data Authentication failed on last transaction
- Offline Dynamic Data Authentication failed on last transaction

This information is copied in the CVR for the current transaction.

The Application Flag data object also contains the status of the previous script processing. The CVR bit "Issuer script processing failed on last or current transaction" is set if the script processing status has the value "script processing under-process"; else it is reset.

☞ **The three least significant bits (b3-b1) of the Script Counter are copied in CVR (byte 4, b8-b6).**

1.3.6.3.3 New Card

If the Last Online Application Transaction Counter (LATC: tag '9F13') is zero, the ICC performs the following function:

- Set the "New card" bit in the CVR (the ICC has never performed a successful Issuer Authentication followed by a TC computation);

1.3.6.3.4 Maximum Offline Transaction Amount

This function determines if the current transaction exceeds a pre-determined maximum offline transaction amount.

As a first step in applying this function the ICC verifies that the Terminal Country Code (tag '9F1A') is the same as the Issuer Country Code (tag '5F28'), and the Transaction Currency Code (Tag'5F2A') supplied in the first GENERATE AC command, is the same as the Application Currency Code (Tag'9F42') stored in the ICC application. If these values are different or if either value is not available, the ICC proceeds to the next function in card risk management.

The ICC determines if the value of the Amount Authorized (tag '9F02') exceeds the parameter "Maximum Domestic Offline Transaction Amount." If the value is exceeded, the ICC sets the "Maximum Offline Transaction Amount exceeded" bit in the CVR.

1.3.6.3.5 Velocity Checking of Offline Consecutive Transactions

The "Velocity Checking of Offline Consecutive Transactions" function determines if the limit set for the maximum number of consecutive offline transactions has been exceeded.

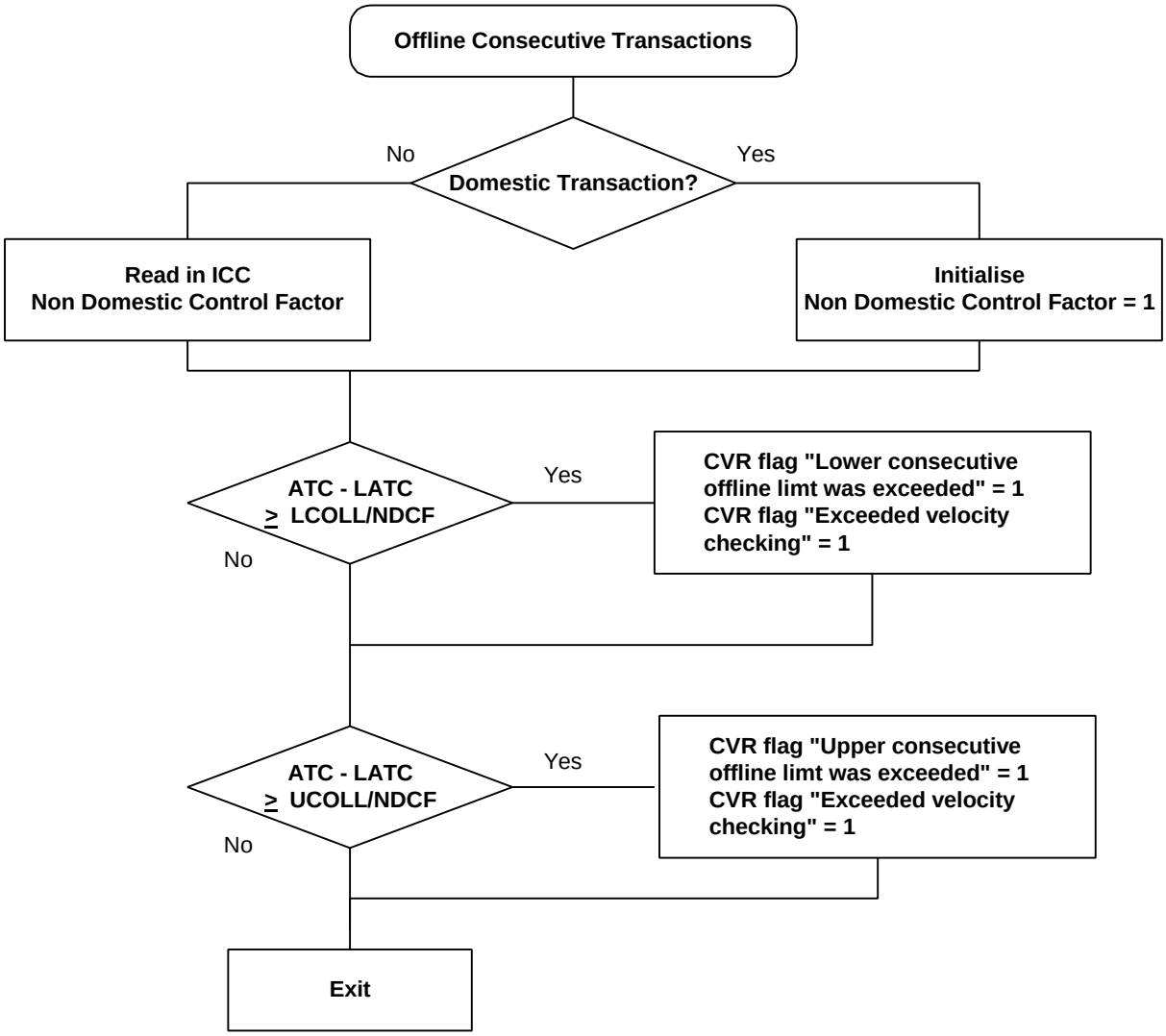
If the Terminal Country Code (tag '9F1A') is NOT the same as the Issuer Country Code (tag '5F28'), or the Transaction Currency Code (Tag '5F2A') supplied in the first GENERATE AC command is NOT the same as the Application Currency Code (Tag '9F42') stored in the ICC application, then the Upper and Lower Consecutive Offline Limits (tag '9F23' and '9F14') are adjusted in accordance with the Non-Domestic Control Factor (NDCF) in Table 1.12 and Figure 1.5. The adjustment will either force all transactions online or increase the frequency of online transactions.

The ICC determines if the difference between the ATC (tag '9F36') and the LATC (tag '9F13') is greater than the Lower Consecutive Offline Limit (tag '9F14'). If it is, the ICC sets the "Lower Consecutive Offline Limit exceeded" bit and the "Exceeded Velocity Checking" bit in the CVR.

The ICC determines if the difference between the ATC (tag '9F36') and the LATC (tag '9F13') is greater than the Upper Consecutive Offline Limit (tag '9F23'). If it is, the ICC sets the "Upper Consecutive Offline Limit" bit and the "Exceeded Velocity Checking" bit in the CVR.

The "Offline Consecutive Transaction Number" function is illustrated in Figure 1.5.

FIGURE 1.5: OFFLINE CONSECUTIVE TRANSACTION FUNCTION PROCESSING



The Issuer can determine its maximum offline risk by using the Lower Consecutive Offline Limit (tag '9F14') and the Upper Consecutive Offline Limit (tag '9F23') that correspond to the maximum value (ATC - LATC). The Issuer can update these parameters through script processing.

☞ **The division LCOLL/NDCF and UCOLL/NDCF can be truncated to integer value**

TABLE 1.12: DATA OBJECTS INVOLVED IN MAXIMUM OFFLINE CONSECUTIVE TRANSACTION NUMBER CHECK

Name	Source	Tag	Length	Format
Application Transaction Counter (ATC)	ICC	'9F36'	2	b
Last Online Application Transaction Counter (LATC) Register	ICC	'9F13'	2	b
Lower Consecutive Offline Limit	ICC	'9F14'	1	b
Upper Consecutive Offline Limit	ICC	'9F23'	1	b
Non Domestic Control Factor (NDCF)	ICC		1	b
Application Currency Code	ICC	'9F42'	2	n 3
Transaction Currency Code	Terminal	'5F2A'	2	n 3
Issuer Country Code	ICC	'5F28'	2	n 3
Terminal Country Code	Terminal	'9F1A'	2	n 3

1.3.6.3.6 Offline Cumulative amount

This function determines if the current transaction, together with the cumulative value of the preceding consecutive offline transactions, exceeds a pre-determined cumulative value set by the issuer.

As a first step in applying this function the ICC verifies that the Transaction Currency Code (Tag'5F2A') supplied in the GENERATE AC command is the same as the Application Currency Code (Tag'9F42') stored in the ICC application. If these values are different, the ICC can try currency conversion or it can proceed to the next function in card risk management.

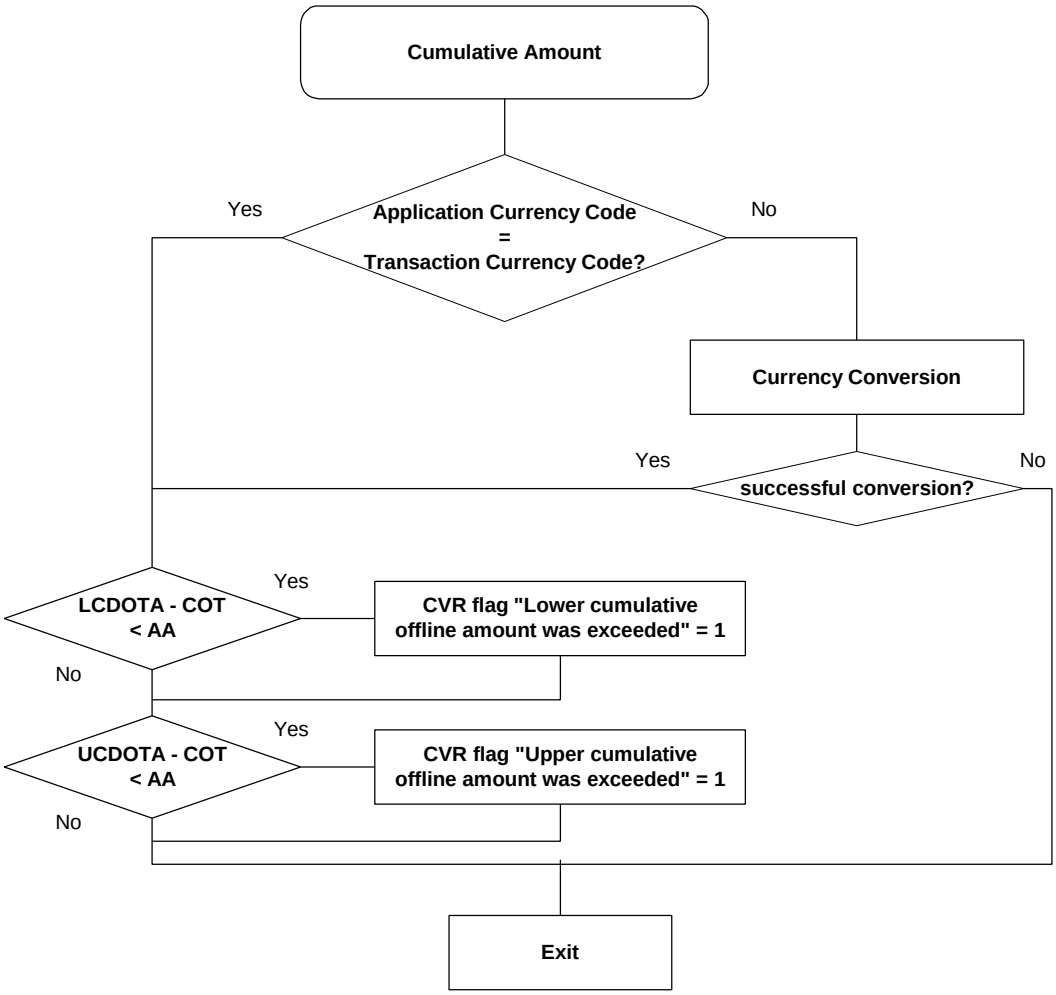
Lower and Upper bounds are used:

- The ICC determines if the value of Cumulative Offline Transactions plus the Amount Authorized (AA: tag '9F02') requested for the current transaction exceeds the Lower Cumulative Offline Transaction Amount. If it is, the ICC sets the "Lower Cumulative Offline Transaction Amount exceeded" bit in the CVR.

- The ICC determines if the value of Cumulative Offline Transactions plus the Amount Authorized (AA: tag '9F02') requested for the current transaction exceeds the Upper Cumulative Offline Transaction Amount. If it is, the ICC sets the "Upper Cumulative Offline Transaction Amount exceeded" bit in the CVR.

The "Offline Cumulative amount" function is illustrated in Figure 1.6.

FIGURE 1.6: CUMULATIVE AMOUNT CHECK PROCESSING



The ICC data objects used in the "Maximum Offline Cumulative Amount" function are defined in Table 1.13.

TABLE 1.13: DATA OBJECTS INVOLVED IN CUMULATIVE AMOUNT FUNCTION FOR A PAYMENT TRANSACTION

Name	Source	Tag	Length	Format
Lower Cumulative Domestic Offline Transaction Amount (LCDOTA)	ICC		6	n12
Upper Cumulative Domestic Offline Transaction Amount (UCDOTA)	ICC		6	n12
Cumulative Offline Transaction (COT)	ICC		6	n12
Application Currency Code	ICC	'9F42'	2	n 3
Transaction Currency Code	Terminal	'5F2A'	2	n 3
Reference Currency Conversion Table	ICC		Up to 20	b

1.3.6.3.6.1 Currency conversion

Currency conversion in the ICC is included at the issuers discretion. It is strongly recommended that issuers only use this function if they are confident that currency fluctuations will not expose them to unmanageable risks.

The ICC initiates currency conversion by listing the "Amount in Reference Currency" (tag '9F3A') and "Transaction Reference Currency Code" (tag '9F3C') in CDOL1.

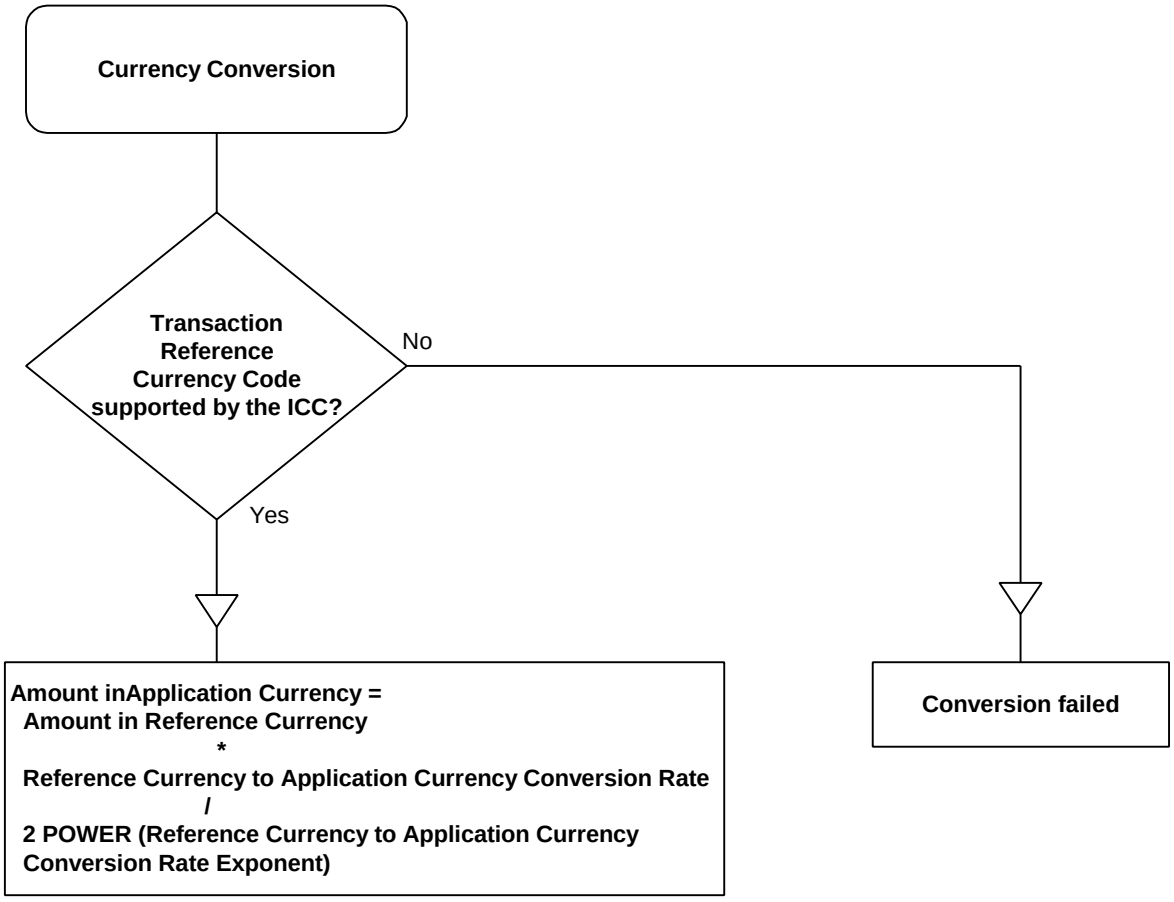
☞ **The terminals that support currency conversion, convert the "Amount Authorized" (AA: tag '9F02') into the "Amount in Reference Currency" (tag '9F3A'). It is done by using the conversion rate managed by the terminals (from the Transaction Currency Code (tag '5F2A') to the Transaction Reference Currency Code (tag '9F3C').**

Then the ICC converts the "Amount in Reference Currency" (tag '9F3A') into the "Amount in Application Currency" using a conversion rate found in the Reference Currency Conversion Table from the Transaction Reference Currency Code (tag '9F3C') to the "Application Currency Code" (tag '9F42').

Conversion at ICC level

The "Currency Conversion" function is illustrated in Figure 1.7.

FIGURE 1.7: CURRENCY CONVERSION AT ICC LEVEL



If the terminal's "Transaction Reference Currency Code" (tag '9F3C') equals one of the reference currencies supported by the ICC, then the ICC converts the "Amount in Reference Currency" (tag '9F3A') provided by the terminal in binary format into the "Amount in Application Currency code", which then is used by the ICC. Otherwise currency conversion has failed.

The "Reference Currency Conversion Table" is an ICC internal data object. It can contain the conversion rates for up to four reference currencies supported by the ICC. For each reference currency supported by the ICC, this table contains an entry as described in Table 1.14.

TABLE 1.14: REFERENCE CURRENCY CONVERSION TABLE

Field	Size in bytes	Format	Meaning
1	2	n3	Reference Currency Code (according ISO 4217)
2	2	b	Reference Currency to Application Currency Conversion rate
3	1	b	Reference Currency to Application Currency Conversion rate exponent

The ICC data objects involved in the currency conversion function are defined in Table 1.15.

TABLE 1.15: DATA OBJECTS INVOLVED IN CURRENCY CONVERSION AT ICC LEVEL

Name	Source	Tag	Length	Format
Reference Currency Conversion Table	ICC	--	var.	
Amount in Reference Currency	Terminal	'9F3A'	4	b
Transaction Reference Currency Code	Terminal	'9F3C'	2	n3

The Issuer can consider updating the conversion rates from time to time by means of script processing.

Example:

- *Terminal provides*:
  Amount in Reference Currency = 125
  Transaction Reference Currency Code = Dollar

- *The ICC table contains*
  Reference Currency = Dollar
  Conversion Rate = 5045
  Conversion Rate Exponent = $(/2^{10})$

The ICC computes the Amount in Application Currency
Amount in Application Currency = 125 5045 / 1024*

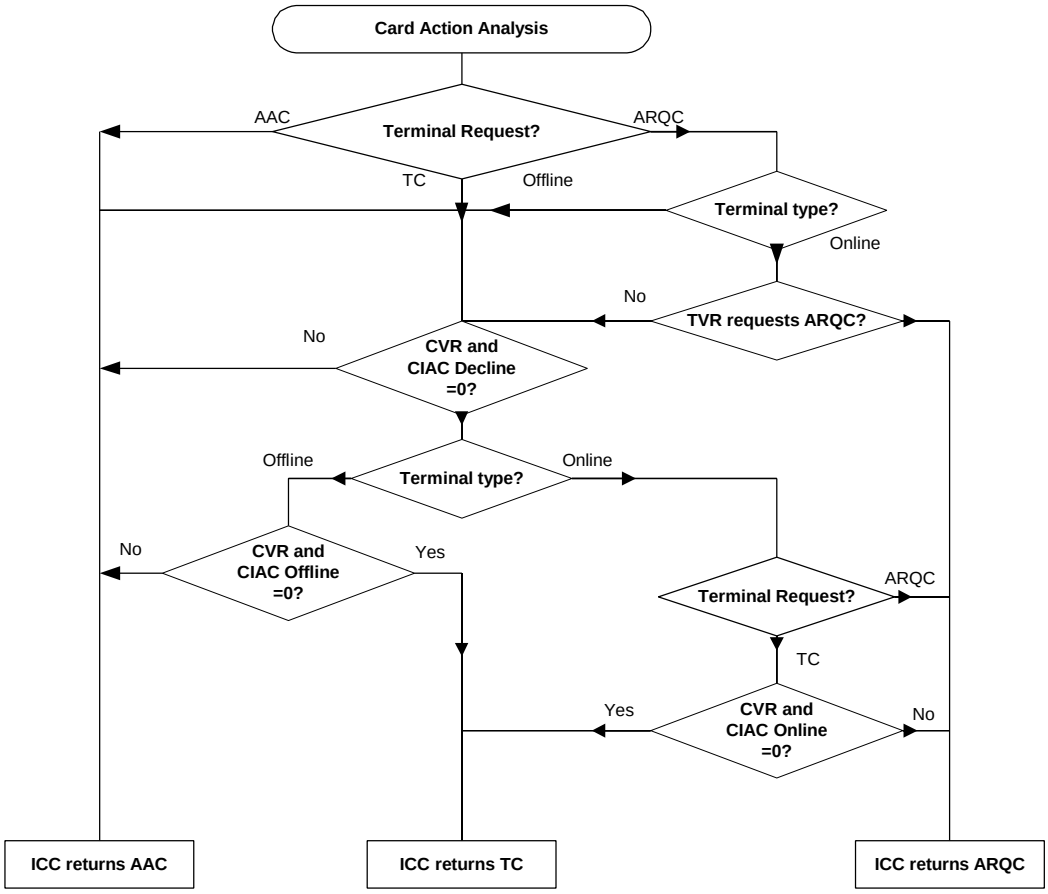
1.3.6.4 Card Action Analysis

This section describes the MasterCard/Europay recommended Card Action Analysis risk functions that are supported for international transactions. After having performed the different CRM functions, the ICC may decide to:

- Decline a transaction
- To complete a transaction online
- To complete a transaction offline

The 'Card Action Analysis' function is illustrated in Figure 1.8.

FIGURE 1.8: CARD ACTION ANALYSIS OF THE FIRST GENERATE AC



In this diagram, AND means "the logical function AND."

1.3.6.4.1 Terminal type?

The ICC checks the Terminal Type[2] (tag '9F35') using the EMV definition.

1.3.6.4.2 TVR asks for ARQC

As a result of the Terminal Risk Management (TRM) process performed before the first GENERATE AC command, the terminal may decide to complete a transaction online. The ICC contains a mandatory data element, the Card TVR Action Code to reflect the Issuer's selected action to be taken on this command. The Card TVR Action Code bit specifies a confirmation to the ICC for going online if the corresponding bit in the Terminal Verification Results (TVR) (tag '95') is set. The easiest way to implement the Card TVR Actions Code is to mirror the format of the TVR.

1.3.6.4.3 Card Issuer Action Codes

The easiest way to implement this flexible risk management is the CIACs having the same format as the CVRs: each CIAC has a bit correspondence to the CVR (when relevant).

When the terminal asks for an ARQC, and if a bit in the Card TVR Action Code and its corresponding bit in the TVR are both set, the ICC generates an ARQC without any more tests. If the bit in the Card TVR Action Code and its corresponding bit in the TVR are not set, the ICC continues the test with the CIAC data.

As a result of the CRM process performed during the first GENERATE AC command, the ICC may decide whether to decline the transaction, complete the transaction online or offline. For this purpose, the ICC contains three mandatory data elements, to reflect the Issuer's selected action to be taken, based upon the content of the CVR and the Terminal type. These three ICC internal data elements, called Card Issuer Action Codes (CIACs) are:

Card Issuer Action Code—Decline: it specifies the conditions that cause the decline of a transaction without attempting to go online according to the Issuer.

---

2 Terminal Type :
online only: 11, 21, 14, 24, 34
offline with online capability: 12, 22, 15 ,25 ,35
offline only: 13, 23, 16, 26, 36
Both 'online only' and 'offline with online capability' are considered as online in this specification

Card Issuer Action Code—Online: it specifies the conditions that cause a transaction to be completed online according to the Issuer. This data element should only be used when the transaction terminal has online capability.

Card Issuer Action Code—Offline: it specifies the bit-settings that cause a decline of a transaction in two situations:

- Transaction performed on an offline terminal
- Online transaction without Issuer Authentication present

1.3.6.4.3.1 CVR AND CIAC = 0

If a bit is set in the CIAC but no corresponding bit is set in the CVR, the ICC can continue with its card risk management (for CIAC-Decline) or accept the transaction offline (for CIAC-Online and CIAC-Offline).

1.3.6.4.3.2 CVR AND CIAC ¹ 0

The CIAC-bit specifies the action to be taken if the corresponding bit in the CVR is set. This can be a decline or an attempt to go online.

The easiest way to implement this flexible risk management is the CIACs having the same format as the CVRs: each CIAC has a bit correspondence to the CVR (when relevant).

The first step is to check the CVR with the CIAC-Decline. If, after CRM, a bit in the CIAC-Decline and its corresponding bit in the CVR are both set, then the transaction is declined and the ICC generates an AAC, If the bits do not match, the CVR will be verified against either the CIAC-Online, or the CIAC-Offline, depending on the terminal online/offline capability.

1.3.6.4.4.1 ICC used in an online capable terminal

The CVR is verified against the CIAC-Online. If, after CRM, a bit in the CIAC-Online and its corresponding bit in the CVR are both set, the transaction is completed online and the ICC generates an ARQC. If the bits do not match, the transaction is approved and the ICC generates a TC.

1.3.6.4.4.2 ICC used in an offline only terminal

The CVR is verified against the CIAC-Offline. If, after CRM, a bit in the CIAC-Offline and its corresponding bit in the CVR are both set, the transaction is declined and the ICC generates an AAC. If the bits do not match, the transaction is approved and the ICC generates a TC.

1.3.6.5 Update ICC Parameters

If the ICC approves the transaction offline (TC) after the CRM, it updates the cumulative offline transaction counters:

- The Cumulative Offline Transaction (if the function has been performed)
- The CVR bits
  - On 1st Generate AC indicate "TC returned in first Generate AC"
  - On 2nd Generate AC indicate "Second Generate AC not requested"

If the ICC requires an online transaction ARQC, it updates in the Application Flags the status of the transaction:

- Last online transaction not completed.

The ICC also updates the CVR bits:

- On 1st Generate AC indicate "ARQC returned in first Generate AC"
- On 2nd Generate AC indicate "Second Generate AC not requested"

If the ICC declines the transaction (AAC), an analysis of the TVR (tag '95') is performed to set several bits in the Application Flag:

- The "Offline Static data authentication failed on last transaction" is set if the TVR indicate that Offline Static Data Authentication has failed.
- The "Offline Dynamic data authentication failed on last transaction" is set if the TVR indicate that Offline Dynamic Data Authentication has failed.

The ICC also updates the CVR bits:

- On 1st Generate AC indicate "AAC returned in first Generate AC"
- On 2nd Generate AC indicate "Second Generate AC not requested"

1.3.6.5.1 Transaction completion/Online processing

The Application Cryptogram (tag '9F26') is computed, the CID (tag '9F27') is filled according to the cryptogram type and the CRM results and a response message is returned to the terminal.

Online processing is performed as described in the *EMV"96 ICC Specification for Payment Systems* and in the *EMV"96 ICC Terminal Specification for Payment Systems*.

1.3.7 Issuer Authentication

It is indicated in the ICC's Application Interchange Profile (AIP: tag '82') whether the terminal should issue the External Authenticate command or not. If the corresponding bit is set in the AIP, the terminal transmits an EXTERNAL AUTHENTICATE command to the ICC. If the bit is not set in the AIP, the Issuer Authentication Data may be requested by the ICC in the CDOL2.

1.3.7.1 Issuer Authentication Data definition

The Issuer Authentication Data (tag '91') consist of a cryptogram, generated by the Issuer or the Issuer agent, and the Issuer Authentication Response Code (IARC). The cryptogram can be verified by the ICC, as verification that the terminal went online.

1.3.7.2 Issuer Authentication Response Code definition

The Issuer Authentication Response Code is the response of the Issuer or Issuer agent, indicating whether the transaction is approved or declined. Since it is signed in the cryptogram, the ICC can be sure that it is effectively the answer of the Issuer.

1.3.7.3 ARQC generated?

When receiving the Issuer Authentication, the ICC always checks if it returned an ARQC in response to the first GENERATE AC command. If not, the ICC will abort the transaction and will respond with an error.

1.3.7.4 Issuer Authentication

The Issuer Authentication mechanism is described in Section 2. The ICC sets "Issuer Authentication performed" in Sequence flag.

1.3.7.5 Update Internal Indicator "Issuer Authentication Successful"

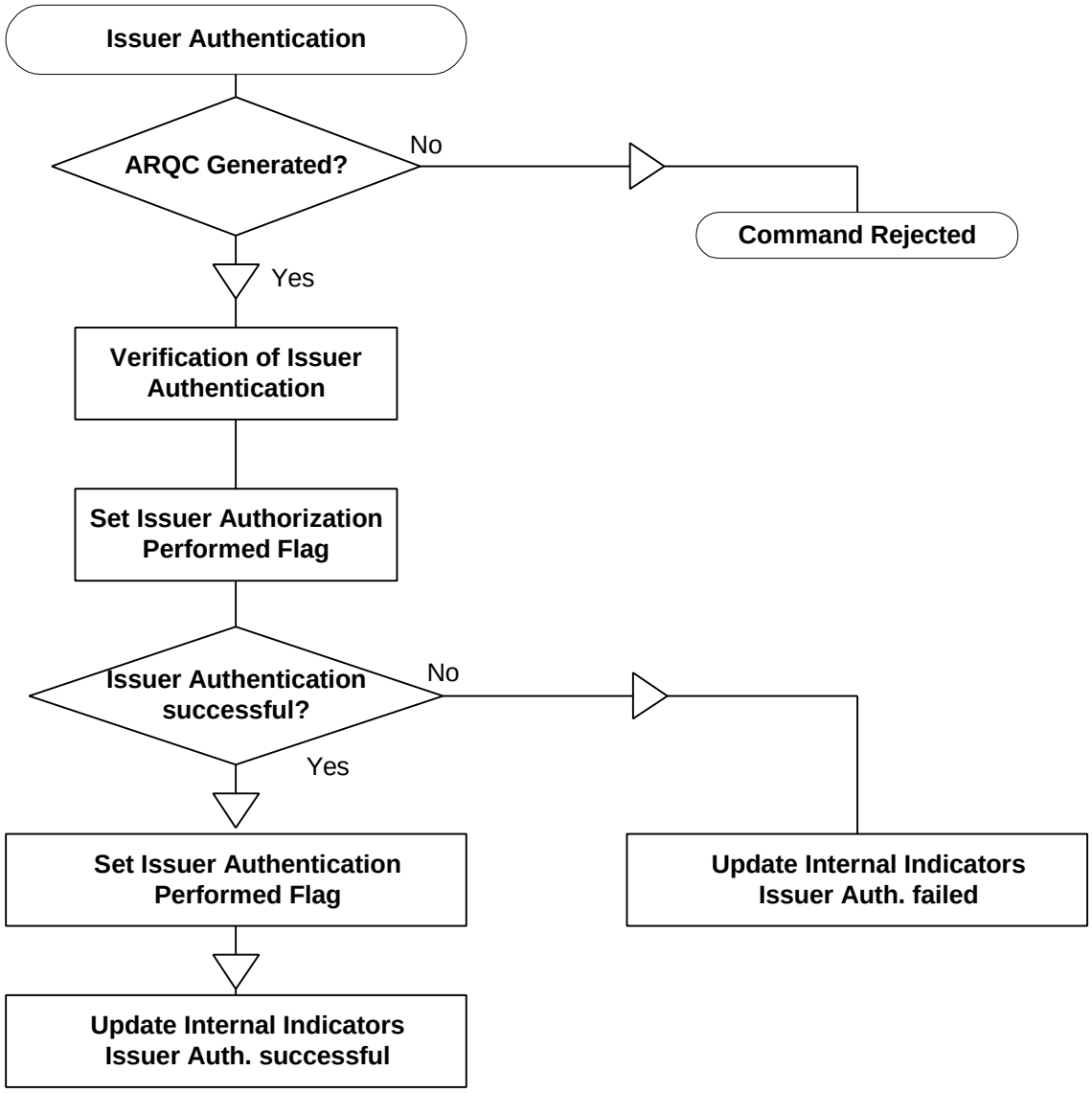
The ICC sets "Issuer Authentication successful" in the Sequence Flag because the Issuer Authentication is successful. The last two bytes of the Issuer Authentication Data indicate the Issuer Authentication Response Code. If these bytes are equal to '00', '01' or '08', the Issuer approves the transaction; else the transaction is declined.

1.3.7.6 Update Internal Indicator "Issuer Authentication Failed"

The ICC resets "Issuer Authentication successful" in Sequence Flag because the Issuer Authentication has failed.

Issuer Authentication is illustrated in Figure 1.9.

FIGURE 1.9: ISSUER AUTHENTICATION OVERALL DIAGRAM



1.3.8 Second GENERATE AC Command

Second GENERATE AC is performed as described in the *EMV' 96 ICC Specification for Payment Systems.*

When the ICC responds with an ARQC in response to the first GENERATE AC command, the terminal transmits the transaction online (if capable of doing so). After the terminal receives the response, it transmits a second GENERATE AC command to the ICC requesting a TC or an AAC

The second Generate AC command is similar to the first Generate AC command, except that the returned Application Cryptogram (AC) is limited to a TC or an AAC.The data field of the second GENERATE AC command is not TLV encoded and the format of the data to be included in the command data is specified in the CDOL2 (tag '8D').

As a result of the second GENERATE AC command, the ICC response consists of the following data objects:

TABLE 1.16: DATA ELEMENTS SENT TO THE TERMINAL

Name	Source	Tag	Length	Format
CID (Cryptogram Information Data)	ICC	'9F27'	1	b
ATC (Application Transaction Counter)	ICC	'9F36'	2	b
Application Cryptogram	ICC	'9F26'	8	b
Issuer Application Data	ICC	'9F10'	8-32[1]	b

1.3.8.1 Cryptogram Information data

For the purpose of this document, online advice can be required (bit b4 = 1 in CID) in case the PIN is blocked (reason: PIN Try Limit exceeded) but not in case the Issuer Authentication has failed.[3] The "Issuer Authentication Failed" bit of the CID is reset when the ICC performs a successful Issuer Authentication. If Issuer Authentication is not performed or failed, the "Issuer Authentication Failed" bit of the CID is set. In case both errors occur (PIN Try Limit exceeded and Issuer Authentication Failed), only PIN Try Limit exceeded is reported in CID.

[3] Advice messages will not be supported across MasterCard networks but may be supported under other circumstances.

1.3.8.2 Card Risk Management Data CDOL2

The CDOL2 is the EMV data required for the second GENERATE AC command. CDOL2 is defined in Section 3.

The Issuer Authentication Data object (tag '91') is present if the Issuer Authentication is performed during the second Generate AC; if not, this data object is absent in CDOL2.

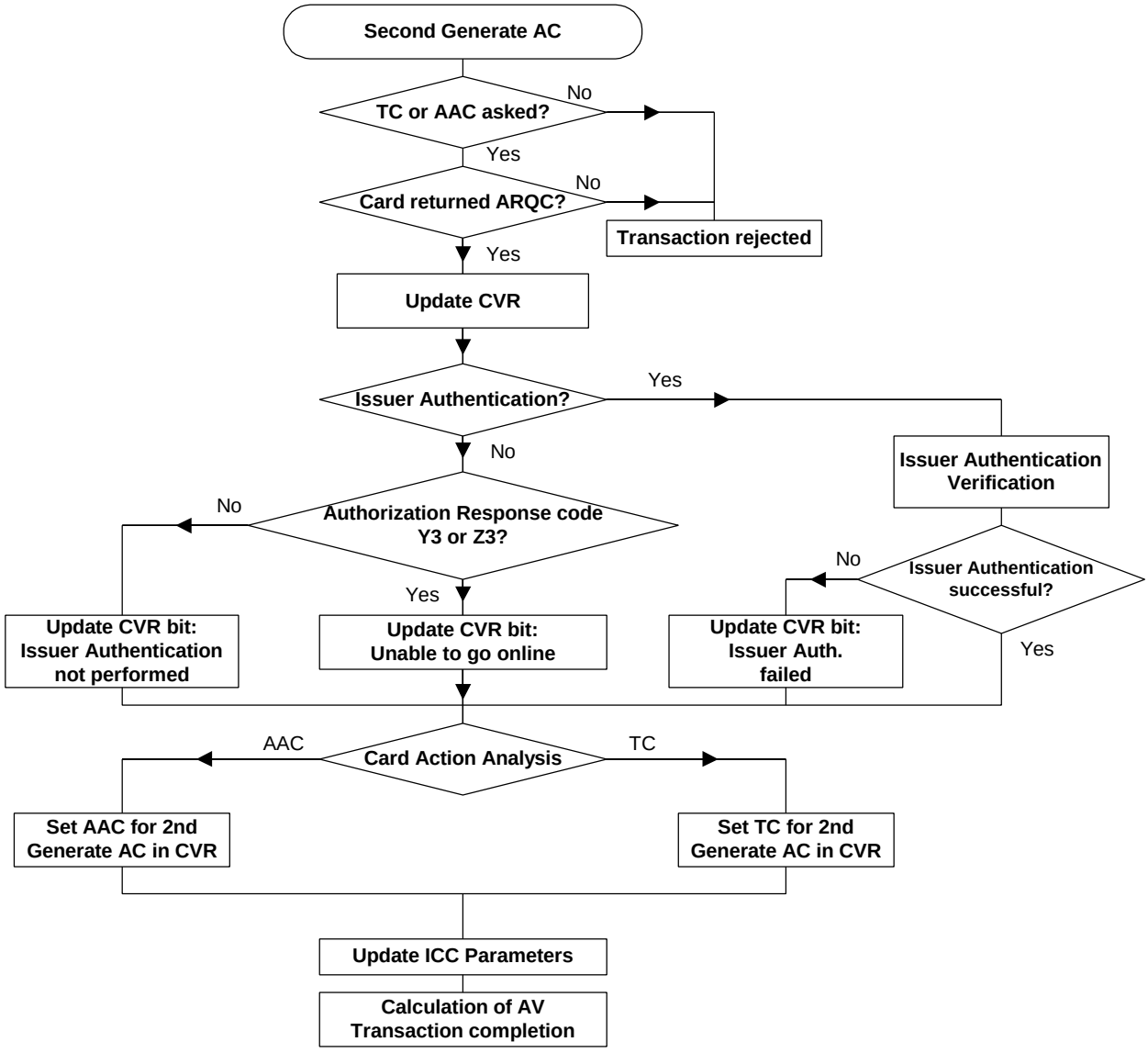
The Authorization Response Code (tag '8A') data object is always present in CDOL2. The Terminal Verification Results data object (tag '95') is always present in the CDOL2 because the contents have been updated between first and second Generate AC. For the other data objects in CDOL2 as referenced in Section 3, the ICC could use the data elements previously received in the CDOL1; therefore, they might not be requested in the CDOL2.

☞ **It is recommended to list in the CDOL2 only those data elements (TVR, Authorization Response Code...) that have changed between the first and the second GENERATE AC. It prevents the terminal from changing data (Amount, Transaction Currency Code...) that should not have been modified between the first and second GENERATE AC.**

1.3.8.3 Internal Process (Issuer Authentication Data performed during External Authenticate)

Figure 1.10 describes the functions performed by the ICC as a result of the second GENERATE AC command.

FIGURE 1.10: SECOND GENERATE AC OVERALL DIAGRAM



1.3.8.3.1 TC or AAC asked?

The second GENERATE AC must be for an AAC or a TC; otherwise the command is rejected.

1.3.8.3.2 Card returned ARQC?

The ICC checks if it returned an ARQC in response to the first GENERATE AC command. If not, the ICC will abort the transaction and respond with an error.

1.3.8.3.3 Update CVR

The CVR bits are updated accordingly:

- If the internal flag "Script Failed" is set, update CVR bit "Issuer script processing failed on last or current transaction."
- The CVR bit "PIN Try Limit exceeded" is updated according to the value of the PIN Try Counter at the beginning of the 2nd Generate AC processing. If the PIN Try Counter equals 0, the CVR bit is set; if not the CVR bit is reset.
- The CVR bit "Offline PIN Verification was performed" is set if a VERIFY command is received during the transaction.
- The CVR bit "Offline PIN Verification failed" is updated according to the last VERIFY command result. It is set if the PIN verification failed; it is reset if the VERIFY command was successful.

☞ **The last two bullets represent an exceptional case; normally a VERIFY command is not received between a 1st and 2nd Generate AC.**

1.3.8.3.4 Issuer Authentication

The meaning of this test depends if Issuer authentication is performed during or before the second Generate AC. If the Issuer Authentication is performed before the second Generate AC (by an External Authentication), the test means: Issuer Authentication already performed? (an External Authentication has been performed). If the Issuer Authentication is performed during the second Generate AC, the test means: Issuer Authentication data present? (Issuer Authentication data not totally filled with hexadecimal zeroes bytes).

1.3.8.3.5 Issuer Authentication verification

The meaning of this test depends if Issuer authentication is performed during or before the second Generate AC. If the Issuer Authentication is performed before the second Generate AC (by an External Authentication), this box is empty.

When Issuer Authentication data (tag '91') is present in the 2nd GENERATE AC command data (non- hexadecimal zeroes bytes), the Issuer Authentication is performed according to Issuer Authentication mechanism described in section 2.

1.3.8.3.6 Issuer Authentication successful?

If the Issuer Authentication is unsuccessful (performed during the 2nd Generate AC or previously in an External Authentication) the ICC sets the CVR bit "Issuer Authentication Failed."

1.3.8.3.7 Authorization Response Code "Y3" or "Z3"?

The Authorization Response Code (tag '8A') can be filled by the Issuer (or the Issuer agent) or by the Terminal. If the terminal was incapable of going online, the following options exist:

TABLE 1.17: AUTHORIZATION RESPONSE CODE - UNABLE TO GO ONLINE

Code	Meaning
'Y3'	Unable to go online – Offline Approved
'Z3'	Unable to go online – Offline Declined

If the Authorization Response Code is filled with the values 'Y3' or 'Z3', the ICC sets the CVR bit "Unable to go online" to indicate that the terminal failed going online.

If the Authorization Response Code is different from 'Y3' and 'Z3', the ICC sets the CVR bit "Issuer Authentication not performed after online authorization" to indicate that the terminal has completed the online authorization but Issuer Authentication Data was not provided.

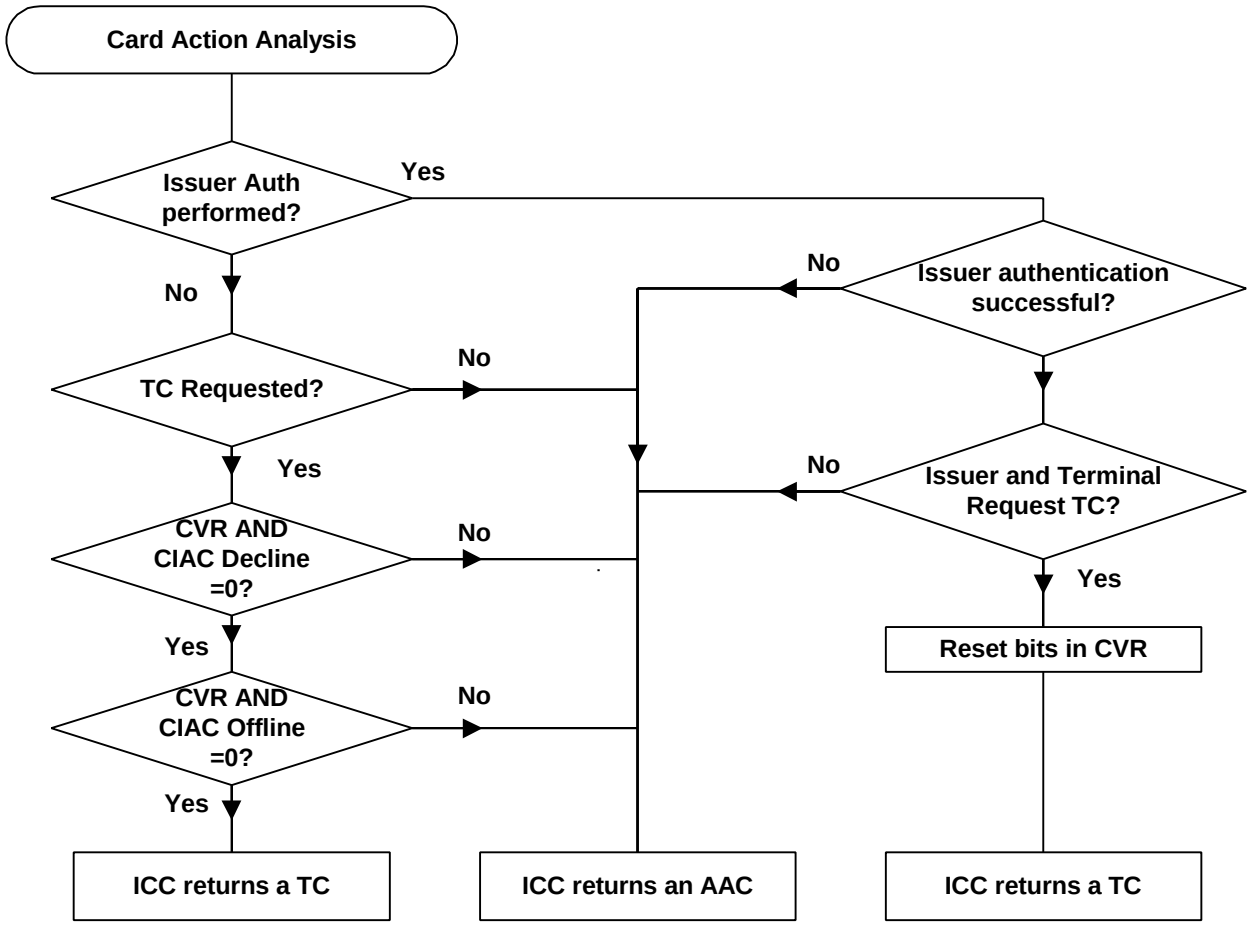
1.3.8.3.8 Card Action Analysis

This section describes the MasterCard/Europay recommended Card Action Analysis risk functions performed during the second Generate AC. The ICC may decide to:

- Decline a transaction
- To complete a transaction offline.

The 'Card Action Analysis' function is illustrated in Figure 1.11.

FIGURE 1.11: CARD ACTION ANALYSIS OF THE SECOND GENERATE AC



1.3.8.3.8.1 TC Requested?

The TC is requested by the Reference Control Parameter P1 with the code of 'TC'.

If the terminal requests an AAC, the ICC returns an AAC.

1.3.8.4.8.2 CVR AND CIAC Decline =0?

If the terminal asks for a TC (without providing online Issuer Authentication Data), the ICC uses the updated CVR and compares it with the CIAC Decline. If a bit in the CIAC Decline and its corresponding bit in the CVR are both set, the transaction is declined and the ICC generates an AAC. If the bits do not match, the test with CIAC Offline will be performed.

1.3.8.3.8.3 CVR AND CIAC Offline =0?

If the terminal asks for a TC (without providing online Issuer Authentication Data), the ICC uses the updated CVR and compares it with CIAC Offline. If a bit in CIAC offline and its corresponding bit in the CVR are both set, the transaction is declined and the ICC generates an AAC. If the bits do not match, the transaction is approved and the ICC generates a TC.

1.3.8.3.8.4 Issuer and Terminal Request a TC?

The last two bytes of the Issuer Authentication Data (tag '91') indicate the response code returned by the Issuer or its agent. If these bytes are equal to '00', '01' or '08', the transaction is approved, otherwise the transaction is declined. The Terminal requests a TC or an AAC with the value of the Reference Control Parameter (P1 parameter) of the second GENERATE AC command. The ICC returns a TC if both the Issuer and the Terminal require a TC; otherwise, the ICC returns an AAC. The ICC disregards the Authorization Response Code (tag '8A').

1.3.8.3.8.5 Reset bits in CVR

- The CVR "Last online transaction not completed" bit is reset
- The CVR "Issuer Authentication failure on last online transaction" bit is reset
- The CVR "Offline Static data authentication failed on last transaction" bit is reset
- The CVR "Offline Dynamic data authentication failed on last transaction" bit is reset
- The CVR "New Card" is reset
- The CVR "Issuer script processing failed on last or current transaction" bit is reset if no critical script has been performed during the current transaction or all critical scripts have been performed successfully during the current transaction
- If no critical script has been performed during the current transaction the Script Counter bits are reset

1.3.8.3.9 Set TC for 2nd Generate AC

Several CVR bits are updated:

- The CVR bits on the 2nd Generate AC indicate "TC is returned in second Generate AC"

1.3.8.3.10 Set AAC for 2nd Generate AC

Several CVR bits are updated:

- The CVR bits on the 2nd Generate AC indicate "AAC is returned in second Generate AC"

1.3.8.3.11 Update ICC parameters

The ICC updates internal indicators, Application Flag, Script Counter and LATC accordingly:

- Reset the Application Flag "Last online transaction not completed"
- If the Issuer Authentication has failed, the Application Flag "Issuer Authentication failure on last online transaction" is set
- If the Issuer Authentication is successful, several flags are reset in Application Flag:
  - "Issuer Authentication failure on last online transaction"
  - "Offline Static data authentication failed on last transaction"
  - "Offline Dynamic data authentication failed on last transaction"

- If the Issuer Authentication is successful and no critical script processing (no '71' received) has been performed during the current transaction:
  - The Script Counter is reset
  - The Script Status Flag is reset

- If the Issuer Authentication is successful and a TC is returned, the following data objects are updated:
  - The LATC is updated with the ATC value
  - The Offline Cumulative Amount is reset

- If the Issuer Authentication is not performed, a TC is returned for an Authorization Response Code different from 'Y3' and the resetting of parameters is allowed by the RIP, the following data objects are updated:
  - The LATC is updated with the ATC value
  - The Offline Cumulative Amount is reset

☞ **The Reset Internal Parameters (RIP) is an internal flag stored in the ICC, and filled by the Issuer which allows or not the reset of the card Risk Management parameters even if no Issuer Authentication is performed.**

1.3.8.3.12 Transaction completion

The Application Cryptogram is computed according to Section 2. The CID (tag '9F27') bits are set according to the cryptogram type and the CRM results and a response message is returned to the terminal.

1.4 STANDARD POST-ISSUANCE FUNCTIONS

Issuer Script processing is performed as described in the *EMV' 96 ICC Specification for Payment Systems.*

1.4.1 Script Processing Overview

Script Processing provides the Issuer with the capability to send Script Commands to the ICC, following an online connection and within a Payment Transaction. Either the critical Script Processing (tag '71') performed between the first and the second Generate AC or the non critical Script Processing (tag '72') performed after the second Generate AC can be supported.

The following functions may be performed using Issuer Script processing:

- Card Blocking
- Application Blocking
- Application Unblocking
- Updating Card Data
- PIN Change/Unblock

Issuer Script processing may be performed at any time after the first GENERATE AC command. Issuers may transmit one Issuer Script containing multiple script commands.

☞ **It is the terminal's responsibility to parse out the script commands and deliver them to the ICC individually.**

For the script processing, two functions are managed in the ICC:

- A counter which counts the script processing commands successfully performed
- A status of script processing

Secure messaging is used for all script commands sent. A complete description of the secure messaging mechanism is provided in Section 2.

If ICC processing of an Issuer Script command fails, the ICC performs:

- Terminate processing of the script processing; all following script commands in the current transaction will be rejected.

ICC Script processing fails if one of the following conditions occurred:

- Secure messaging failed
- Secure messaging not present
- Processing of a command failed

1.4.1.1 Script Processing Counter

The ICC manages a Script Counter; this Script Counter is incremented after each script processing command is successfully completed. It is reset during the second Generate AC after a successful Issuer Authentication if no critical script has been performed during the current transaction.

1.4.1.2 Script Processing Status

The ICC manages two kinds of Script Processing Status. A Script processing status stored in Sequence flag to indicate the script status during the current transaction and a script status in Application Flag to keep the script processing status for the next transaction.

In Sequence Flag, two flags manage the script status:

- "Script Performed": a script command has been received during the current transaction
- "Script Failed": a script command has failed during the current transaction

In Application Flag the script status indicates one of two conditions:

- "Script processing pending": a command of script processing has been received by the ICC; every script processing command updates script processing status to the value "script processing pending" at the beginning of the command.
- "Script processing not pending": this status indicates that the script processing is not pending ; the behavior of script processing status depends on whether the END OF SCRIPT command is supported or not.

*The END OF SCRIPT command is supported

The script processing is concluded by a specific command (END OF SCRIPT), indicating that all previous commands have been delivered; The ICC updates the Script Processing Status to "Script processing not pending" at end of the command when the command is successfully performed.

*The END OF SCRIPT is not supported:

The ICC updates the Script Processing Status to "Script processing not pending" at end of each script processing command when the command is successfully performed.

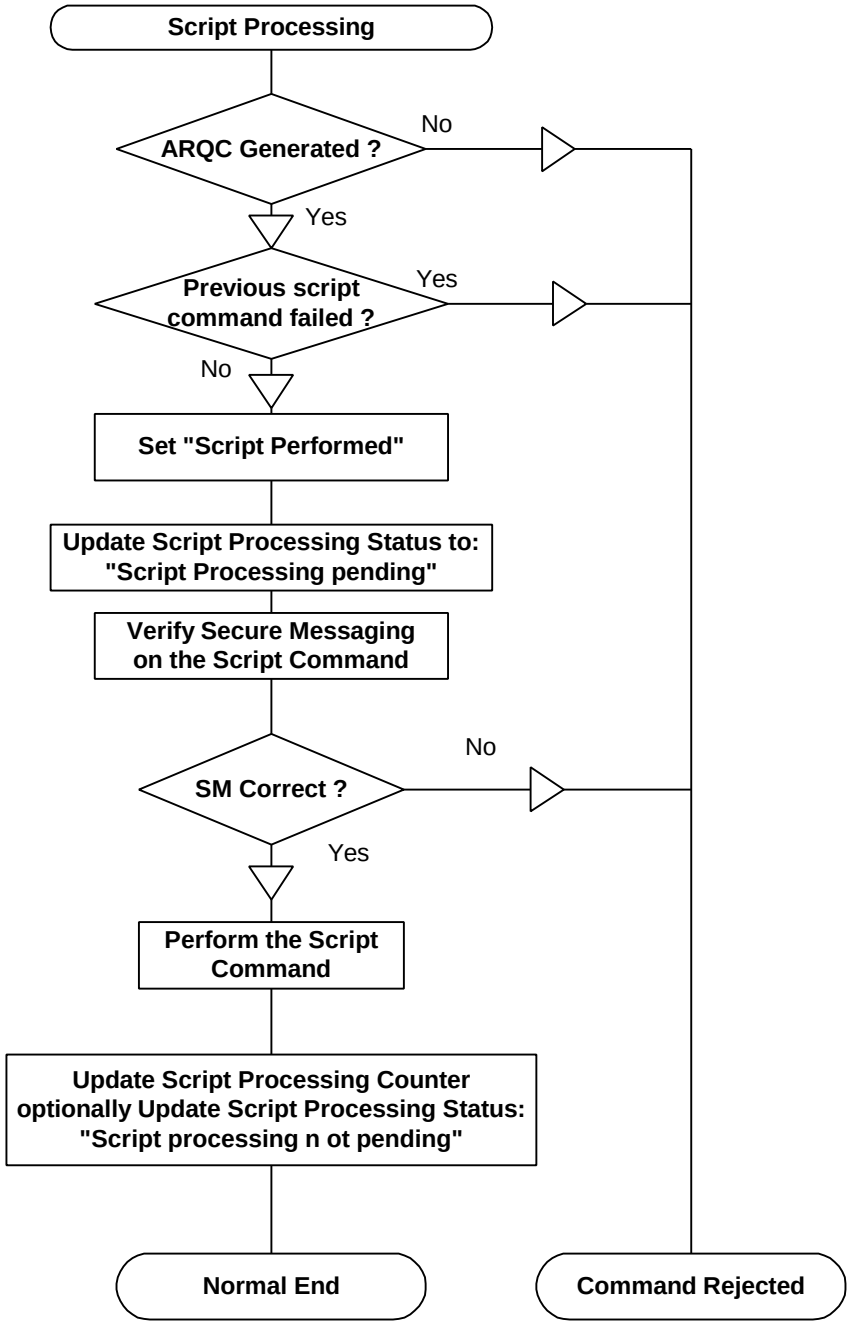
The support of END OF SCRIPT command could be set at personalization step.

If a script processing command has failed (with an error code according to EMV specifications), all next script processing commands during the current transaction are rejected. The Script Processing Status is reported to the Issuer during the current and next transactions (see first and Second Generate AC section).

1.4.1.3 Script Processing Overall Diagram

For each script command received by the ICC, the internal ICC processing is as indicated in Figure 1.12.

FIGURE1.12: SCRIPT PROCESSING OVERALL DIAGRAM



1.4.1.3.1 ARQC generated?

The ICC rejects the command if an ARQC was not generated.

1.4.1.3.2 Previous script command failed?

The ICC rejects the new command if the previous script processing command has failed during the current transaction ("Script failed" flag set in Sequence flag, see Table 1.4).

1.4.1.3.3 Set "script performed"

The ICC set "Script Performed" flag in Sequence flag.

1.4.1.3.4 Update script processing status to "script processing pending"

The ICC updates its internal script processing status to "pending" to indicate that script processing has started (in Application Flag see Table 1.5).

1.4.1.3.5 Verify secure messaging

The ICC verifies the Secure Messaging Cryptogram of that command using the Script session key (a complete description of the script session key generation and the verification of secure messaging is provided in Section 2).

1.4.1.3.6 SM correct ?

The ICC compares the result of its calculation with the Secure Messaging Cryptogram.

1.4.1.3.7 Perform the script command

The ICC performs the command.

1.4.1.3.8 Update script processing counter (optionally update script processing status)

The ICC increments its Script Processing Counter when the command is successfully performed. Optionally, if the END OF SCRIPT command is not supported in script processing Script Processing Status is updated to "Script processing not pending".

1.4.1.3.9 Command rejected

The ICC updates the "Script Failed" flag in Sequence Flag if the script command failed.

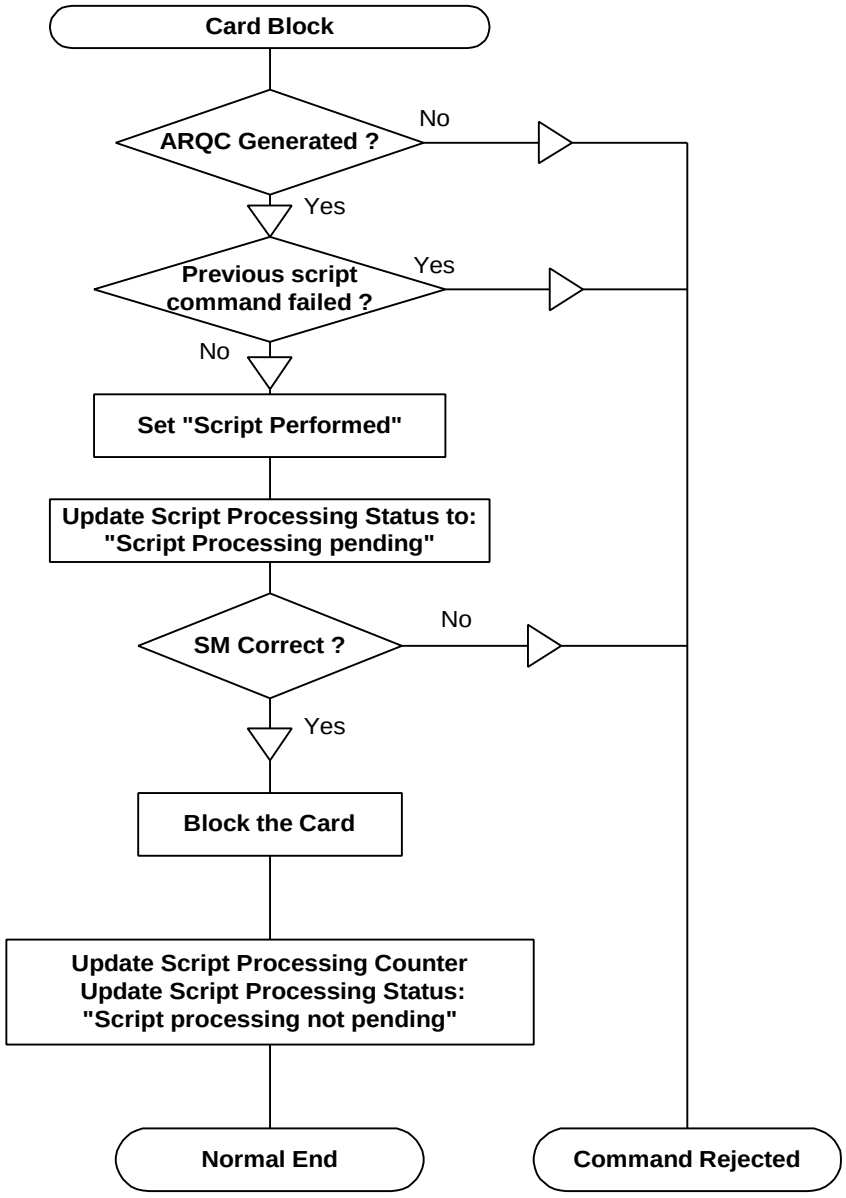
1.4.2 Card Blocking

Card blocking may be performed if the issuer determines that any future use of the ICC is to be prevented. Card blocking is usually performed only if the card has been reported as lost or stolen, since none of the applications in the ICC can be used after card blocking. The Issuer may block the card by using the EMV CARD BLOCK command in the Issuer script.

If during the processing of a transaction, the card is blocked, the ICC and terminal must continue to process the transaction through to completion. A blocked card can not be unblocked using an Issuer Script Command or any other command; therefore, the ICC is disabled. Therefore, the blocked ICC responds to a SELECT command or any other command, with status bytes indicating "Function not supported" (SW1 SW2 = '6A81') and performs no further actions.

☞ **Some Issuer proprietary commands may still be supported to retrieve the Card Life Cycle Data.**

FIGURE 1.13: SCRIPT PROCESSING TO BLOCK THE CARD



1.4.3 Application Blocking

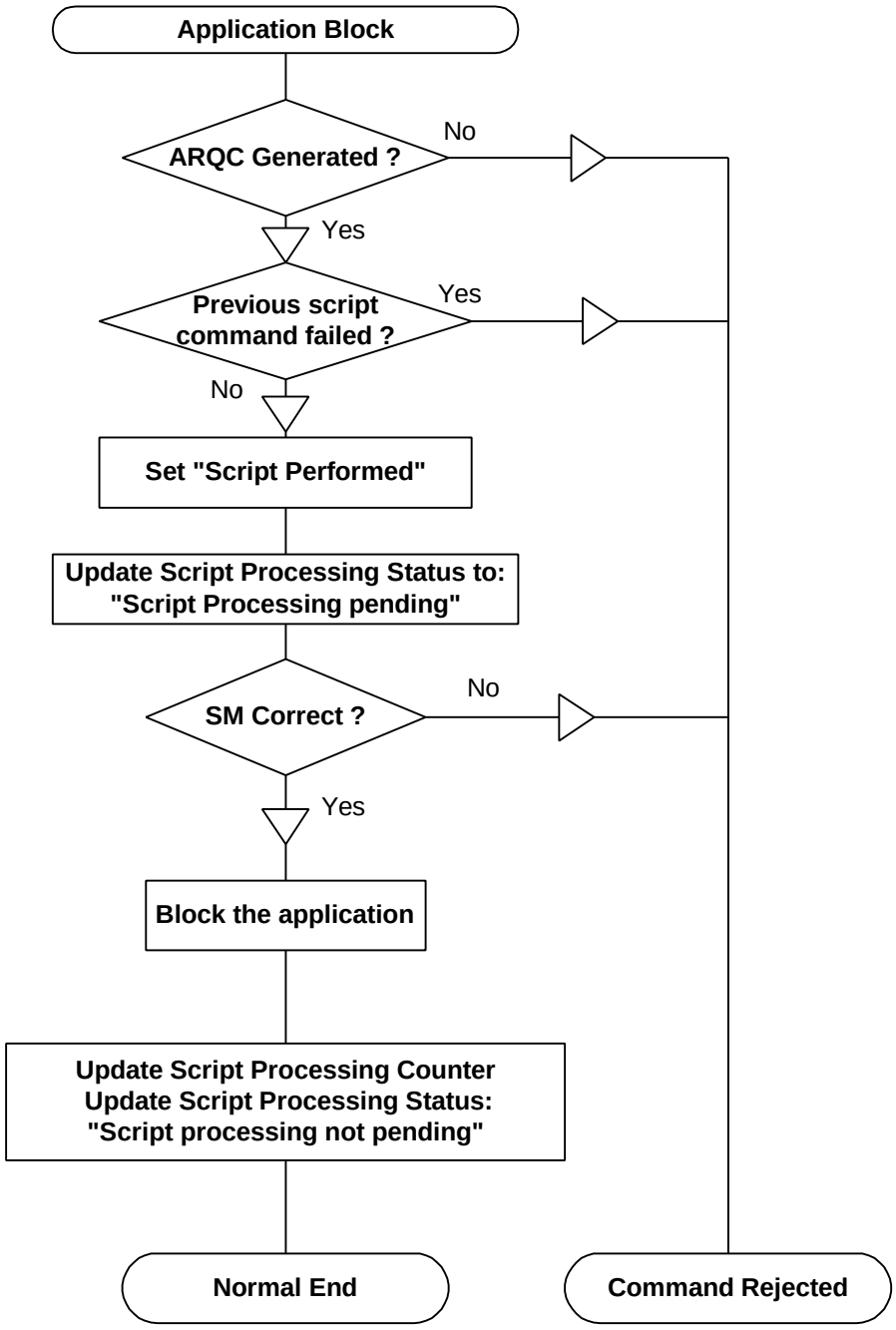
Application blocking may be performed if the issuer determines that the application in use is to be inactivated. The blocked application may subsequently be re-activated by the issuer with another command.

If during the processing of a transaction, the application is blocked, the ICC and the terminal continue to process the transaction through to completion. During any subsequent application selection, the ICC does not allow the inactivated application(s) to perform a financial transaction.

It is possible for a dedicated terminal to select an application that was previously inactivated to unblock the application. If this occurs, the ICC is required to return an AAC in response to a GENERATE AC command.

If Issuer Script processing is used to block an application, the APPLICATION BLOCK command is used.

FIGURE 1.14: SCRIPT PROCESSING TO BLOCK THE SELECTED APPLICATION



1.4.4 Application Unblocking

Application unblocking is performed only at dedicated terminals controlled by the issuer. Issuer Script processing may be used to perform this function.

If Issuer Script processing is used to unblock an application, the APPLICATION UNBLOCK command is used.

If the application is blocked a dedicated terminal for unblocking applications will use the AAC obtained from the ICC as random number for the Secure Messaging (see Section). Issuer Authentication by means of secure messaging is sufficient; (therefore the EXTERNAL AUTHENTICATE command is not needed as an additional Issuer Authentication).

☞ **The Application Unblock may be processed before the second Generate AC, or the second Generate AC may be omitted.**

1.4.4.1.1 Verify Secure Messaging

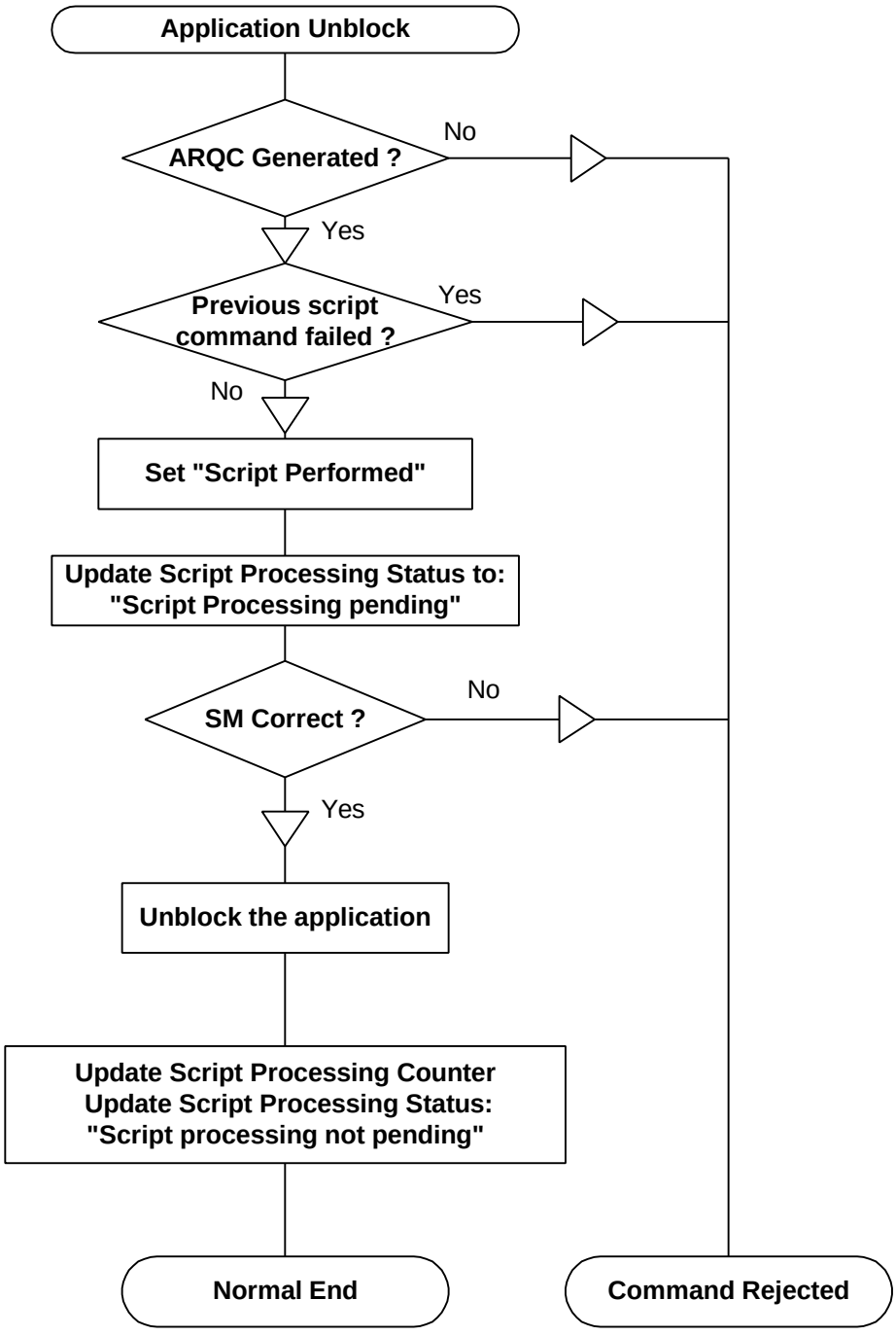
Secure Messaging is used to send the command to the ICC, meaning that the command is protected for Integrity. A Script Session Key (see Section 2) is derived based on the AAC (obtained in the response to the 1st Generate AC).

Using the Script Session Key, the ICC verifies the Secure Messaging Cryptogram of the APPLICATION UNBLOCK command.

1.4.4.1.2 Unblock the selected ADF

The ICC re-activates the currently selected blocked application.

FIGURE 1.15: SCRIPT PROCESSING TO UNBLOCK THE APPLICATION



1.4.5 Updating Card Data

The Update Card Data function provides the Issuer with the ability to update ICC data after issuance of the ICC.

Updating ICC data is normally limited to updating the card risk parameters. This function is performed using Secure Messaging for Integrity. Either the PUT DATA command or the UPDATE RECORD command is used.

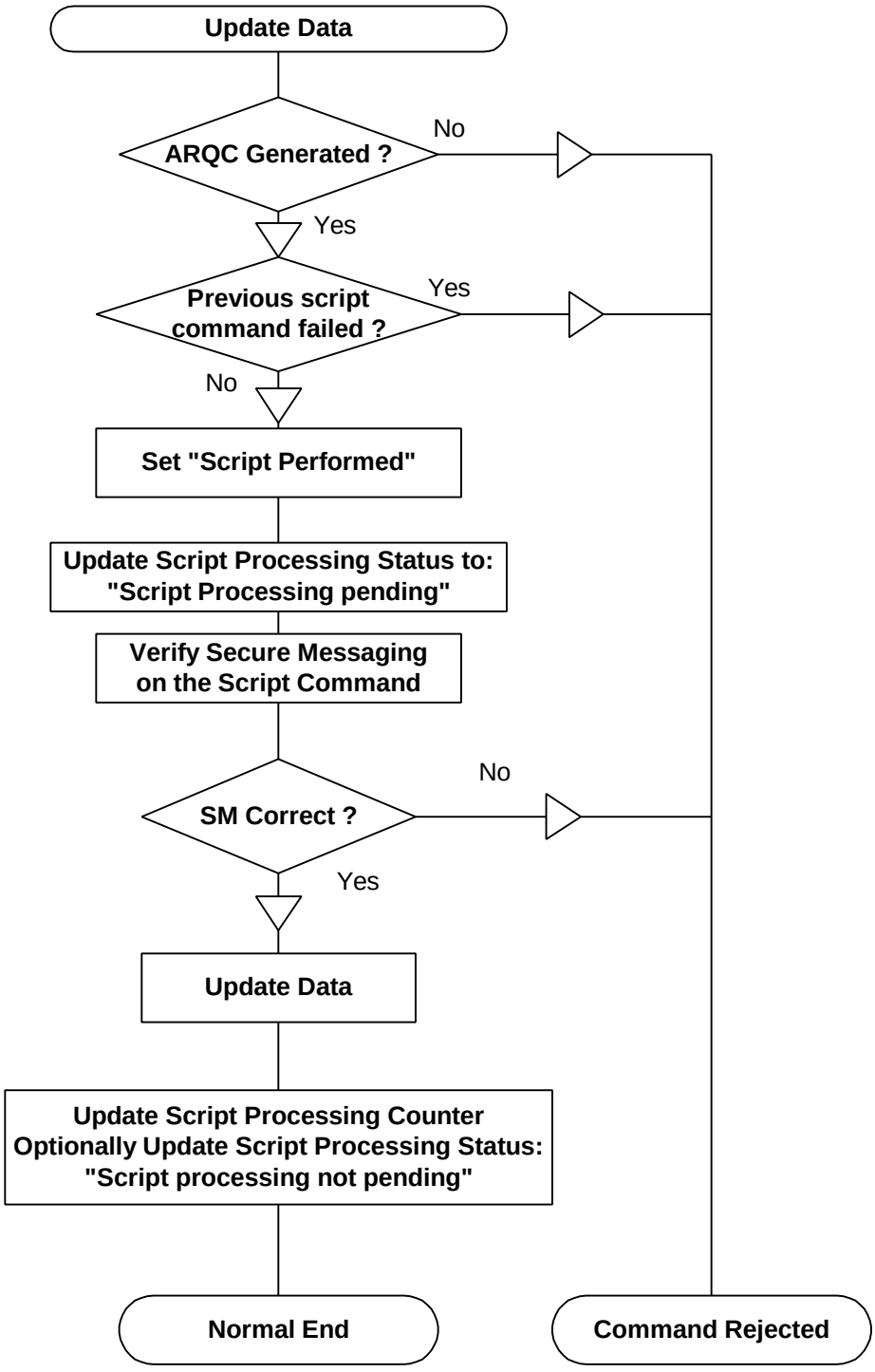
The following data elements may be updated using Issuer Script processing:

- Upper Consecutive Offline Limit (tag '9F23')
- Lower Consecutive Offline Limit (tag '9F14')
- Maximum Domestic Offline Transaction Amount
- Upper Cumulative Domestic Offline Transaction Amount
- Lower Cumulative Domestic Offline Transaction Amount
- Non-Domestic Control Factor
- Card Issuer Action Codes
- Card TVR Action Code
- Reference Currency Conversion Table

1.4.5.1 Update the data

If the access conditions are satisfied, the ICC replaces the data, using the data provided in the update data command.

FIGURE 1.16: SCRIPT PROCESSING TO UPDATE DATA



1.4.6 PIN Change/Unblock

Issuer Script Processing is used to perform this function. If issuer Script Processing is used to change or unblock a PIN, the PIN CHANGE/UNBLOCK command is used.

☞ **A standard terminal can use the ARQC returned by the first GENERATE AC as random number for the Secure Messaging; a dedicated terminal can use both the AAC or the ARQC by the first GENERATE AC as random number.**

This function is performed using Secure Messaging for Confidentiality and Integrity when the Change option is used. Secure Messaging for Integrity is used when the Unblock option is used. If Secure Messaging is successful, the PIN CHANGE/UNBLOCK command is performed.

☞ **The PIN changing/unblocking may be processed before the second GENERATE AC, or the second GENERATE AC may be omitted (2nd Generate AC can be omitted if AAC was generated in 1st Generate AC).**

The key management and cryptographic security architecture to be used is provided in Section 2.

☞ **Whenever the PIN change is evoked, then Issuers must ensure that no PVV is included on the magnetic stripe data and in the Track 2 equivalent data field in the chip. Only the Application Reference PIN in the chip can be updated by this command.**

1.4.6.1 Unblock or Change

The same command can contain either a PIN change and unblock request or only a PIN unblock request without any PIN change. The ICC checks, which option is to be performed.

1.4.6.2 Decipher PIN

The new PIN value is sent in enciphered format conforming to Secure Messaging for Confidentiality. The ICC using the Script Session Key deciphers the new PIN value.

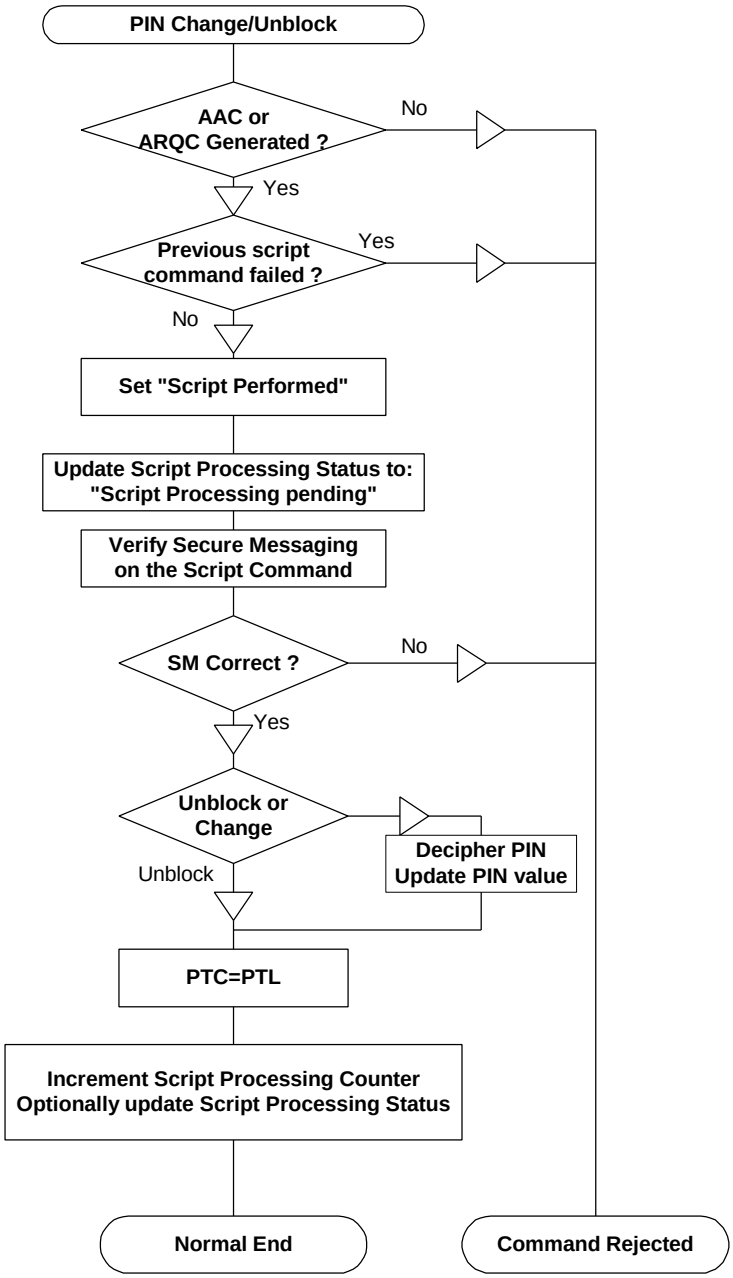
1.4.6.3 Update PIN value

The new PIN value is overwritten, replacing the old PIN value (PIN length may vary from 4 to 12 numeric digits).

1.4.6.4 Update PTC to PTL

The ICC copies the value of the PIN Try Limit into the PIN Try Counter.

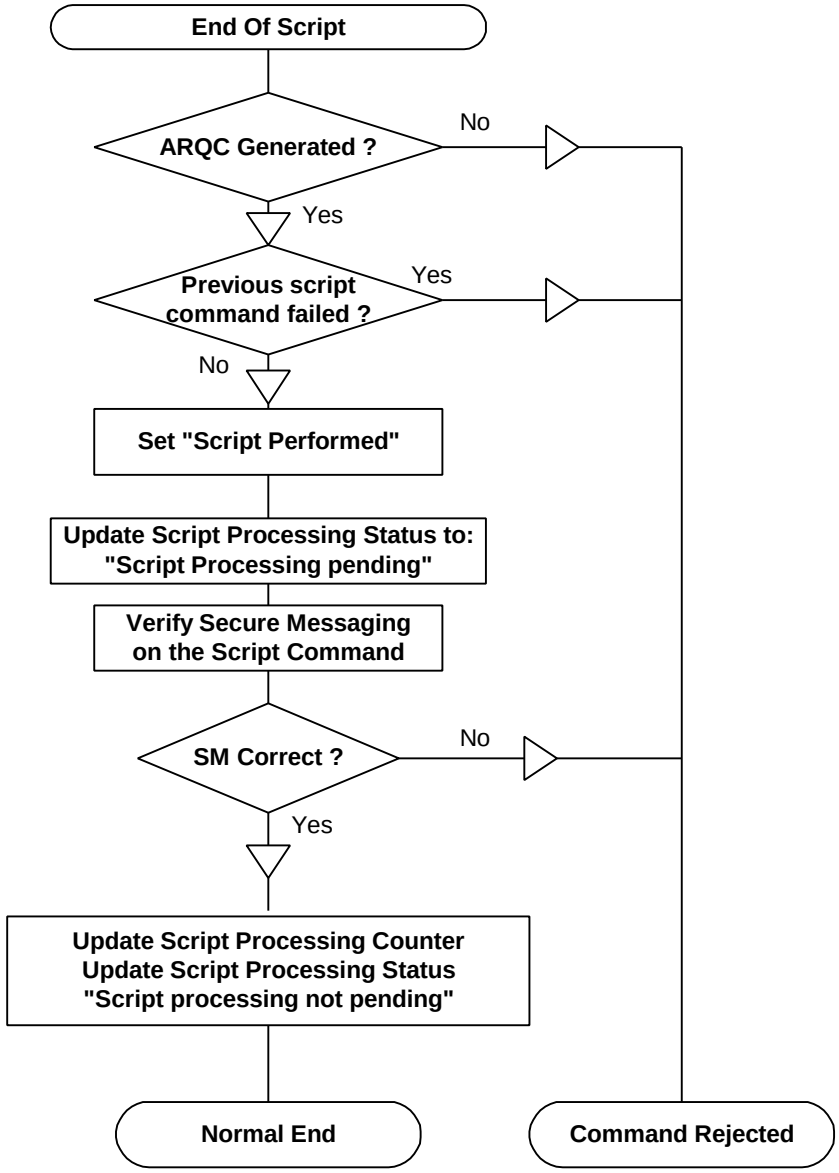
FIGURE 1.17: SCRIPT PROCESSING TO PIN CHANGE/UNBLOCK DURING A PAYMENT TRANSACTION



1.4.7 End Of Script

The END OF SCRIPT command terminates a script. This function is performed using Secure Messaging for Integrity. It updates the Script Processing Status to "Script processing not pending."

FIGURE 1.18: END OF SCRIPT PROCESSING



SECTION 2 SECURITY SPECIFICATION OF EMV '96, VERSION 3.1.1 ICC SPECIFICATION FOR PAYMENT SYSTEMS TRANSACTIONS

2.0 OVERVIEW

This section contains a detailed specification of the ICC application security:

- Static Data Authentication
- Dynamic Data Authentication
- PIN Encipherment
- Application Cryptogram generation
- Issuer Authentication
- Secure Messaging

Then the following concepts are described:

- Issuer derivation of the unique DES3ICC Master Keys for Application Cryptogram Generation and Secure Messaging
- Issuer derivation of the Data Authentication Codes
- ICC and Issuer derivation of the ICC Dynamic Numbers

The specification for the security mechanisms and cryptographic algorithms used, are specified in Appendix B.

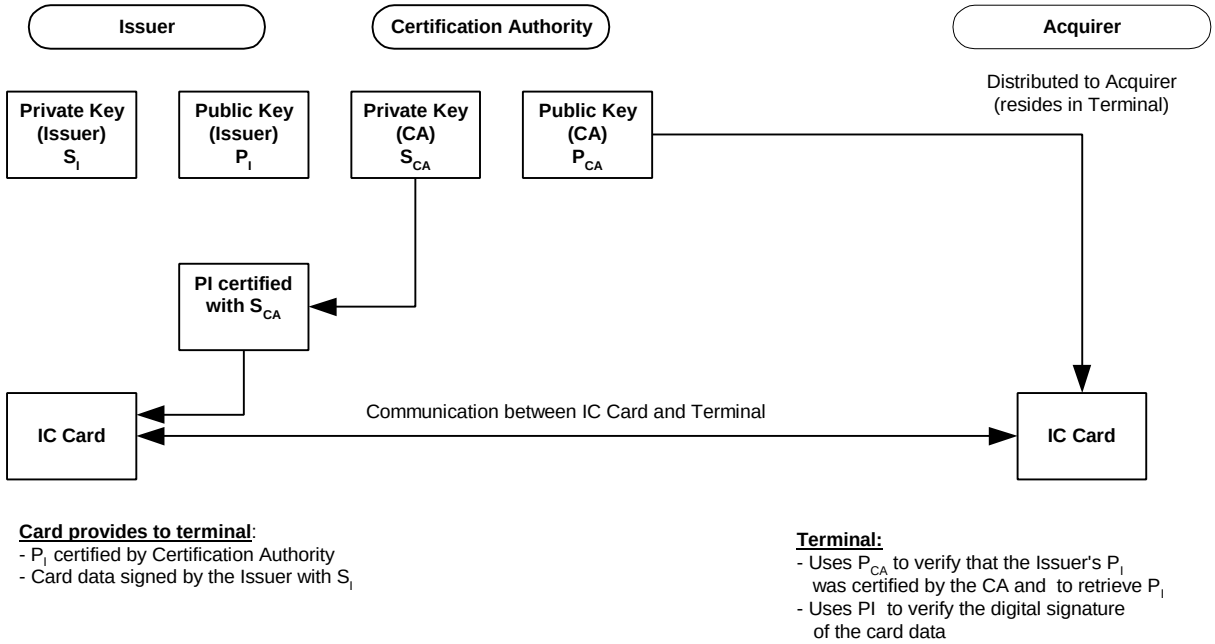
This document does not cover the security aspects of ICC production, initialization and personalization, and the overall key management.

2.1 STATIC DATA AUTHENTICATION

2.1.1 Keys and Certificates

To support Static Data Authentication, an ICC contains the Signed Static Application Data, which is signed with the Issuer Private Key. The Issuer Public Key is stored on the ICC as an Issuer Public Key Certificate.

FIGURE 2.1: TWO LAYER PUBLIC KEY CERTIFICATION SCHEME



A two layer public key certification scheme as depicted in Figure 2.1 is used. Each Issuer Public Key is signed, with the Certification Authority (CA) Private Key, to obtain the Issuer Public Key Certificate. To perform Static Data Authentication, the terminal first verifies the Issuer Public Key Certificate with the CA Public Key to retrieve and authenticate the Issuer Public Key. Then the Issuer Public Key is used to verify the Signed Static Application Data stored in the ICC.

The signature scheme specified in Appendix B is applied to the data specified in Table 2.1 using the Certification Authority Private Key 'S_{CA}' to obtain the Issuer Public Key Certificate. The public key pair of the CA has a public key modulus of N_{CA} bytes.

The signature scheme specified in Appendix B is applied to the data specified in Table 2.2 using the Issuer Private Key 'S_I' to obtain the Signed Static Application Data. The public key pair of the Issuer has an Issuer Public Key Modulus of N_I bytes. If $N_I > (N_{CA} - 36)$, the Issuer Public Key Modulus is split into two parts:

- One part consisting of the $N_{CA} - 36$ most significant bytes of the modulus (the Leftmost Digits of the Issuer Public Key).
- A second part consisting of the remaining $N_I - (N_{CA} - 36)$ least significant bytes of the modulus (the Issuer Public Key Remainder).

All the information necessary for Static Data Authentication is specified in Table 2.5 and stored in the ICC. With the exception of the RID, which is obtained from the AID, this data is retrieved with the READ RECORD command. If any of this data is missing, Static Data Authentication fails.

For length specifications of CA and Issuer Public Key Modules and the values of CA and Issuer Public Key Exponents, see respectively Table C.1 and Table C.2 in appendix C.

TABLE 2.1: ISSUER PUBLIC KEY DATA TO BE SIGNED BY THE CERTIFICATION AUTHORITY

Field Name	Length	Description	Format
Certificate Format	1	Hex. value '02'	b
Issuer Identification Number	4	Leftmost 3-8 digits from the Primary Account Number (PAN) (padded to the right with hex. 'F's)	cn 8
Certificate Expiration Date	2	MMYY after which this certificate is invalid	n 4
Certificate Serial Number	3	Binary number unique to this certificate assigned by the certification authority	b
Hash Algorithm Indicator	1	Identifies the hash algorithm used to produce the Hash Result in the digital signature scheme	b
Issuer Public Key Algorithm Indicator	1	Identifies the digital signature algorithm to be used with the Issuer Public Key	b
Issuer Public Key Length	1	Identifies the length of the Issuer Public Key Modulus in bytes	b
Issuer Public Key Exponent Length	1	Identifies the length of the Issuer Public Key Exponent in bytes	b
Issuer Public Key or Leftmost Digits of the Issuer Public Key	$N_{CA} - 36$	The $N_{CA} - 36$ most significant bytes of the Issuer Public Key	b
Issuer Public Key Remainder	0 or $N_I - N_{CA} + 36$	This field is only present if $N_I > N_{CA} - 36$ and consists of the $N_I - N_{CA} + 36$ least significant bytes of the Issuer Public Key	b
Issuer Public Key Exponent	1 or 3	Issuer Public Key Exponent	b

TABLE 2.2: STATIC APPLICATION DATA TO BE SIGNED BY THE ISSUER

Field Name	Length	Description	Format
Signed Data Format	1	Hex. value '03'	b
Hash Algorithm Indicator	1	Identifies the hash algorithm used to produce the Hash Result in the digital signature scheme	b
Data Authentication Code (DAC)	2	Issuer-assigned code	b
Pad Pattern	$N_I - 26$	Pad pattern consisting of $N_I - 26$ bytes of value 'BB'	b
Static Data to be Authenticated	var.	Static data to be authenticated as specified in Table 3 and Table 4	-

The Data Authentication Code (DAC) in Table 2.2 is an Issuer assigned value. A method for generating a DAC is described in Section 2.

According to EMV'96, the input to the Static Data Authentication process is a concatenation of the records defined by the Application File Locator (AFL), followed by a concatenation of the value fields of the elements specified in the SDA Tag List (which has tag '9F4A').

The input to the Static Data Authentication process consists of the concatenation of the tags, lengths and values of the data objects specified in Table 2.3 and the value of the data object specified in Table 2.4.

TABLE 2.3: STATIC DATA TO BE AUTHENTICATED SPECIFIED BY THE AFL

Value	Format	Length	Tag
Application Effective Date	n 6	3	'5F25'
Application Expiration Date	n 6	3	'5F24'
Application Usage Control	b	2	'9F07'
Application Primary Account Number (PAN)	Cn var. up to 19	var. up to 10	'5A'
Application PAN Sequence number	n 2	1	'5F34'
Issuer Action Code— Default	b	5	'9F0D'
Issuer Action Code—Denial	b	5	'9F0E'
Issuer Action Code—Online	b	5	'9F0F'

Only Application Effective Date is optional in Table 2.3.

If AIP or AID is included in the Static Data to be authenticated, they are indicated in the SDA Tag List.

TABLE 2.4: STATIC DATA, IF TO BE AUTHENTICATED, INDICATED IN THE SDA TAG LIST

Value	Format	Length	Tag
Application Identifier	b	5-16	'84'
Application Interchange Profile (AIP)	b	2	'82'

Table 2.5 describes the data elements needed by the terminal for SDA.

TABLE 2.5: DATA OBJECTS REQUIRED FOR STATIC DATA AUTHENTICATION

Tag	Length	Value	Format
—	5	Registered Application Provider Identifier (RID) (obtained from the AID)	b
'8F'	1	Certification Authority Public Key Index	b
'90'	N_{CA}	Issuer Public Key Certificate	b
'92'	$N_I - N_{CA} + 36$	Issuer Public Key Remainder, if present	b
'9F32	1 or 3	Issuer Public Key Exponent	b
'93'	N_I	Signed Static Application Data	b
—	var.	Static data to be authenticated as specified by Table 2.3 and Table 2.4.	—

2.1.2 Retrieval of the Certification Authority Public Key

The terminal reads the CA Public Key Index. Using the CA Public Key Index and the RID, the terminal identifies and retrieves the terminal-stored CA Public Key Modulus and Exponent and the associated key-related information, and the corresponding algorithm to be used.

If the terminal does not have the key stored associated with this CA Public Key Index and RID, Static Data Authentication fails.

2.1.3 Retrieval of the Issuer Public Key

1. If the Issuer Public Key Certificate has a length different from the length of the CA Public Key Modulus obtained in the previous section, Static Data Authentication fails.

2. To obtain the recovered data specified in Table 2.6, apply the recovery function specified in Appendix B to the Issuer Public Key Certificate using the CA Public Key in conjunction with the corresponding algorithm. If the Recovered Data Trailer is not equal to 'BC', Static Data Authentication fails.

TABLE 2.6: FORMAT OF THE DATA RECOVERED FROM THE ISSUER PUBLIC KEY CERTIFICATE

Field Name	Length	Description	Format
Recovered Data Header	1	Hex. Value '6A'	b
Certificate Format	1	Hex. Value '02'	b
Issuer Identification Number	4	Leftmost 3-8 digits from the PAN (padded to the right with hex. 'F's)	cn 8
Certificate Expiration Date	2	MMYY after which this certificate is invalid	n 4
Certificate Serial Number	3	Binary number unique to this certificate assigned by the certification authority	b
Hash Algorithm Indicator	1	Identifies the hash algorithm used to produce the Hash Result in the digital signature scheme	b
Issuer Public Key Algorithm Indicator	1	Identifies the digital signature algorithm to be used with the Issuer Public Key	b
Issuer Public Key Length	1	Identifies the length of the Issuer Public Key Modulus in bytes	b
Issuer Public Key Exponent Length	1	Identifies the length of the Issuer Public Key Exponent in bytes	b
Issuer Public Key or Leftmost Digits of the Issuer Public Key	$N_{CA}-36$	The $N_{CA} - 36$ most significant bytes of the Issuer Public Key	b
Hash Result	20	Hash of the Issuer Public Key and its related information	b
Recovered Data Trailer	1	Hex. Value 'BC'	b

3. Check the Recovered Data Header. If it is not '6A', Static Data Authentication fails.

4. Check the Certificate Format. If it is not '02', Static Data Authentication fails.

5. Concatenate, from left to right, the second data element through the tenth data element in Table 2.6 ('Certificate Format' through 'Issuer Public Key or Leftmost Digits of the Issuer Public Key'), followed by the Issuer Public Key Remainder and finally the Issuer Public Key Exponent.

6. Apply the indicated hash algorithm (derived from the Hash Algorithm Indicator) to the result of the concatenation of the previous step to produce the hash result.

7. Compare the calculated hash result from the previous step with the recovered hash result. If they are not the same, Static Data Authentication fails.

8. Verify that the Issuer Identification Number matches the leftmost 3-8 PAN digits (allowing for the possible padding of the Issuer Identification Number with hexadecimal 'F's). If not a match, Static Data Authentication fails.

9. Verify that the last day of the month specified in the Certificate Expiration Date is equal to or later than today's date. If the Certificate Expiration Date is earlier than today's date, the certificate has expired, in which case Static Data Authentication fails.

10. If the Issuer Public Key Algorithm Indicator is not recognized, Static Data Authentication fails.

11. If all the checks above are correct, concatenate the Leftmost Digits of the Issuer Public Key and the Issuer Public Key Remainder to obtain the Issuer Public Key Modulus, and continue with the next steps for the verification of the Signed Static Application Data.

2.1.4 Verification of the Signed Static Application Data

1. If the Signed Static Application Data has a length different from the length of the Issuer Public Key Modulus, Static Data Authentication fails.

2. To obtain the Recovered Data specified in Table 2.7, apply the recovery function specified in Appendix B to the Signed Static Application Data using the Issuer Public Key in conjunction with the corresponding algorithm. If the Recovered Data Trailer is not equal to 'BC', Static Data Authentication fails.

TABLE 2.7: FORMAT OF THE DATA RECOVERED FROM THE SIGNED STATIC APPLICATION DATA

Field Name	Length	Description	Format
Recovered Data Header	1	Hex. Value '6A'	b
Signed Data Format	1	Hex. Value '03'	b
Hash Algorithm Indicator	1	Identifies the hash algorithm used to produce the Hash Result in the digital signature scheme	b
Data Authentication Code	2	Issuer-assigned code	b
Pad Pattern	$N_I - 26$	Pad pattern consisting of $N_I - 26$ bytes of value 'BB'	b
Hash Result	20	Hash of the Static Application Data to be authenticated	b
Recovered Data Trailer	1	Hex. Value 'BC'	b

3. Check the Recovered Data Header. If it is not '6A', Static Data Authentication fails.

4. Check the Signed Data Format. If it is not '03', Static Data Authentication fails.

5. Concatenate, from left to right, the second data element through the fifth data element in Table 2.7 ('Signed Data Format' through 'Pad Pattern'), followed by the Static Data to be Authenticated as specified in Table 2.7.

6. Apply the indicated hash algorithm (derived from the Hash Algorithm Indicator) to the result of the concatenation of the previous step to produce the hash result.

7. Compare the calculated hash result from the previous step with the recovered hash result. If they are not the same, Static Data Authentication fails.

8. If all of the above steps were executed successfully, Static Data Authentication passes.

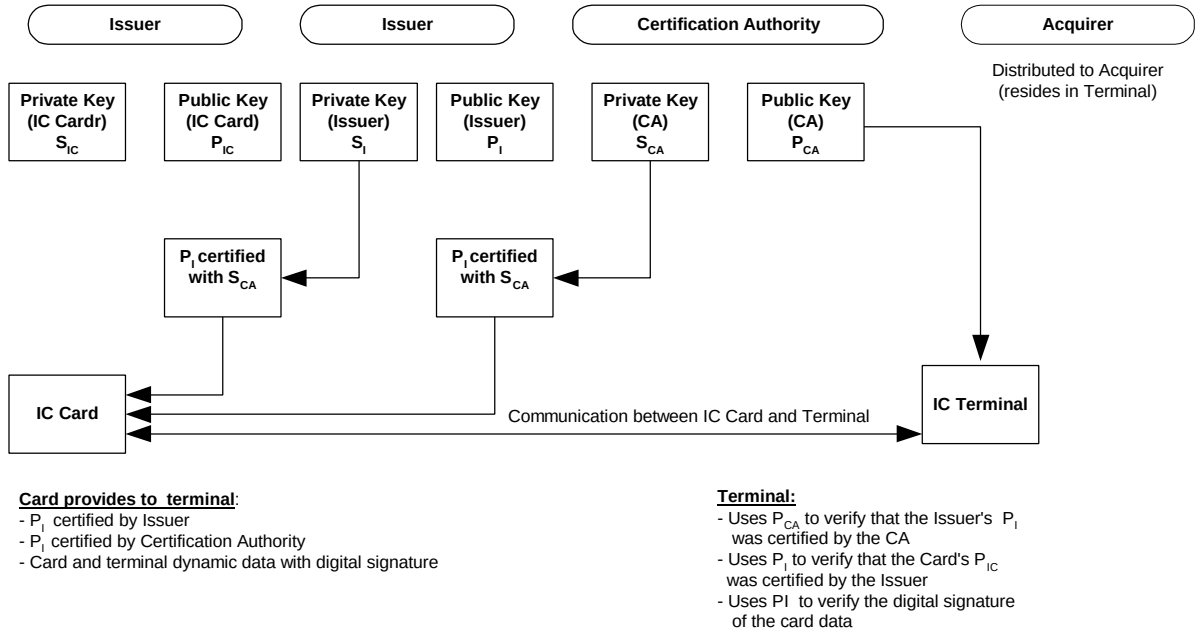
9. The terminal stores the retrieved Data Authentication Code for further processing as described in the *EMV'96 ICC Application Specification for Payment Systems*.

2.2 DYNAMIC DATA AUTHENTICATION

2.2.1 Keys and Certificates

To support Dynamic Data Authentication, an ICC possesses its own unique ICC Public Key pair consisting of a private signature key and the corresponding public verification key. The ICC Public Key is stored on the ICC in the form of an ICC Public Key Certificate.

FIGURE 2.2: THREE LAYER PUBLIC KEY CERTIFICATION SCHEME



A three layer public key certification scheme as depicted in Figure 2.2 is used. Each ICC Public Key is certified by its Issuer, and the Certification Authority (CA) certifies the Issuer Public Key. This implies that, for the verification of an ICC signature, the terminal first needs to verify two certificates to retrieve and authenticate the ICC Public Key, which is then used to verify the ICC dynamic signature.

The signature scheme specified in Appendix B is applied to the data in Table 2.8 and to the data in Table 2.9 using the CA Private Key S_{CA} and the Issuer Private Key S_I to obtain the Issuer Public Key Certificate and ICC Public Key Certificate, respectively.

The CA public key pair has a CA Public Key Modulus of N_{CA} bytes.

The Issuer public key pair has an Issuer Public Key Modulus of N_I bytes. If $N_I > (N_{CA} - 36)$, the Issuer Public Key Modulus is divided into two parts:

- One part consisting of the $N_{CA} - 36$ most significant bytes of the modulus (the Leftmost Digits of the Issuer Public Key)
- A second part consisting of the remaining $N_I - (N_{CA} - 36)$ least significant bytes of the modulus (the Issuer Public Key Remainder)

The ICC public key pair has an ICC Public Key Modulus of N_{IC} bytes. If $N_{IC} > (N_I - 42)$, the ICC Public Key Modulus is divided into two parts:

- One part consisting of the $N_I - 42$ most significant bytes of the modulus (the Leftmost Digits of the ICC Public Key)
- A second part consisting of the remaining $N_{IC} - (N_I - 42)$ least significant bytes of the modulus (the ICC Public Key Remainder)

To execute Dynamic Data Authentication, the terminal first retrieves and authenticates the ICC Public Key (this process is called ICC Public Key authentication). All the information necessary for ICC Public Key authentication is specified in Table 2.10 and stored in the ICC. With the exception of the RID, which can be obtained from the AID, this information is retrieved with the READ RECORD command. If any of this data is missing, Dynamic Data Authentication fails.

For specification of the lengths of CA, Issuer and ICC Public Key Modules and the values of CA, Issuer and ICC Public Key Exponents, see respectively C.1 and C.2 in Appendix C.

TABLE 2.8: ISSUER PUBLIC KEY DATA TO BE SIGNED BY THE CERTIFICATION AUTHORITY

Field Name	Length	Description	Format
Certificate Format	1	Hex. value '02'	b
Issuer Identification Number	4	Leftmost 3-8 digits from the PAN (padded to the right with hex. 'F's)	cn 8
Certificate Expiration Date	2	MMYY after which this certificate is invalid	n 4
Certificate Serial Number	3	Binary number unique to this certificate assigned by the certification authority	b
Hash Algorithm Indicator	1	Identifies the hash algorithm used to produce the Hash Result in the digital signature scheme	b
Issuer Public Key Algorithm Indicator	1	Identifies the digital signature algorithm to be used with the Issuer Public Key	b
Issuer Public Key Length	1	Identifies the length of the Issuer Public Key Modulus in bytes	b
Issuer Public Key Exponent Length	1	Identifies the length of the Issuer Public Key Exponent in bytes	b
Issuer Public Key or Leftmost Digits of the Issuer Public Key	$N_{CA} - 36$	The $N_{CA} - 36$ most significant bytes of the Issuer Public Key	b
Issuer Public Key Remainder	0 or $N_I - N_{CA} + 36$	This field is only present if $N_I > N_{CA} - 36$ and consists of the $N_I - N_{CA} + 36$ least significant bytes of the Issuer Public Key	b
Issuer Public Key Exponent	1 or 3	Issuer Public Key Exponent	b

TABLE 2.9: ICC PUBLIC KEY DATA TO BE SIGNED BY THE ISSUER

Field Name	Length	Description	Format
Certificate Format	1	Hex. value '04'	b
Application PAN	10	PAN (padded to the right with hex. 'F's)	cn 20
Certificate Expiration Date	2	MMYY after which this certificate is invalid	n 4
Certificate Serial Number	3	Binary number unique to this certificate assigned by the Issuer	b
Hash Algorithm Indicator	1	Identifies the hash algorithm used to produce the Hash Result in the digital signature scheme	b
ICC Public Key Algorithm Indicator	1	Identifies the digital signature algorithm to be used with the ICC Public Key	b
ICC Public Key Length	1	Identifies the length of the ICC Public Key Modulus in bytes	b
ICC Public Key Exponent Length	1	Identifies the length of the ICC Public Key Exponent in bytes	b
ICC Public Key or Leftmost Digits of the ICC Public Key	$N_I - 42$	The $N_I - 42$ most significant bytes of the ICC Public Key	b
ICC Public Key Remainder	0 or $N_{IC} - N_I + 42$	The $N_I - N_{CA} + 42$ least significant bytes of the ICC Public Key	b
ICC Public Key Exponent	1 or 3	ICC Public Key Exponent	b
Static Data to be Authenticated	var.	Static data to be authenticated as specified in Table 2.3 and Table 2.4	-

TABLE 2.10: DATA OBJECTS REQUIRED FOR PUBLIC KEY AUTHENTICATION FOR DYNAMIC AUTHENTICATION

Tag	Length	Value	Format
—	5	Registered Application Provider Identifier (RID)	b
'8F'	1	Certification Authority Public Key Index	b
'90'	N_{CA}	Issuer Public Key Certificate	b
'92'	$N_I - N_{CA} + 36$	Issuer Public Key Remainder, if present	b
'9F32'	1 or 3	Issuer Public Key Exponent	b
'9F46'	N_I	ICC Public Key Certificate	b
'9F48'	$N_{IC} - N_I + 42$	ICC Public Key Remainder, if present	b
'9F47'	1 or 3	ICC Public Key Exponent	b
—	var.	Static data to be authenticated as specified in Table 2.3 and Table 4.	-

2.2.2 Retrieval of the Certification Authority Public Key

The terminal reads the CA Public Key Index. Using this index and the RID, the terminal identifies and retrieves the terminal-stored CA Public Key Modulus and Exponent and the associated key-related information, and the corresponding algorithm to be used. If the terminal does not have the key stored associated with this index and RID, Dynamic Data Authentication fails.

2.2.3 Retrieval of the Issuer Public Key

1. If the Issuer Public Key Certificate has a length different from the length of the CA Public Key Modulus obtained in the previous section, Dynamic Data Authentication fails.

2. To obtain the recovered data specified in Table 2.11, apply the recovery function specified in Appendix B on the Issuer Public Key Certificate using the CA Public Key in conjunction with the corresponding algorithm. If the Recovered Data Trailer is not equal to 'BC', Dynamic Data Authentication fails.

TABLE 2.11: FORMAT OF THE DATA RECOVERED FROM THE ISSUER PUBLIC KEY CERTIFICATE

Field Name	Length	Description	Format
Recovered Data Header	1	Hex. value '6A'	b
Certificate Format	1	Hex. value '02'	b
Issuer Identification Number	4	Leftmost 3-8 digits from the PAN (padded to the right with hex. 'F's)	cn 8
Certificate Expiration Date	2	MMYY after which this certificate is invalid	n 4
Certificate Serial Number	3	Binary number unique to this certificate assigned by the certification authority	b
Hash Algorithm Indicator	1	Identifies the hash algorithm used to produce the Hash Result in the digital signature scheme	b
Issuer Public Key Algorithm Indicator	1	Identifies the digital signature algorithm to be used with the Issuer Public Key	b
Issuer Public Key Length	1	Identifies the length of the Issuer Public Key Modulus in bytes	b
Issuer Public Key Exponent Length	1	Identifies the length of the Issuer Public Key Exponent in bytes	b
Issuer Public Key or Leftmost Digits of the Issuer Public Key	$N_{CA} - 36$	The $N_{CA} - 36$ most significant bytes of the Issuer Public Key	b
Hash Result	20	Hash of the Issuer Public Key and its related information	b
Recovered Data Trailer	1	Hex. value 'BC'	b

3. Check the Recovered Data Header. If it is not '6A', Dynamic Data Authentication fails.

4. Check the Certificate Format. If it is not '02', Dynamic Data Authentication fails.

5. Concatenate, from left to right, the second through the tenth data elements in Table 2.11 ('Certificate Format' through 'Issuer Public Key or Leftmost Digits of the Issuer Public Key'), followed by the Issuer Public Key Remainder and finally the Issuer Public Key Exponent.

6. Apply the indicated hash algorithm (derived from the Hash Algorithm Indicator) to the result of the concatenation of the previous step to produce the hash result.

7. Compare the calculated hash result from the previous step with the recovered hash result. If they are not the same, Dynamic Data Authentication fails.

8. Verify that the Issuer Identification Number matches the leftmost 3-8 PAN digits (allowing for the possible padding of the Issuer Identification Number with hexadecimal 'F's). If not, Dynamic Data Authentication fails.

9. Verify that the last day of the month specified in the Certificate Expiration Date is equal to or later than today's date. If the Certificate Expiration Date is earlier than today's date, the certificate has expired, in which case Dynamic Data Authentication fails.

10. If the Issuer Public Key Algorithm Indicator is not recognized, Dynamic Data Authentication fails.

11. If all the checks above are correct, concatenate the Leftmost Digits of the Issuer Public Key and the Issuer Public Key Remainder to obtain the Issuer Public Key Modulus, and continue with the next steps for the retrieval of the ICC Public Key.

2.2.4 Retrieval of the ICC Public Key

1. If the ICC Public Key Certificate has a length different from the length of the Issuer Public Key Modulus obtained in the previous section, Dynamic Data Authentication fails.

2. To obtain the recovered data specified in Table 2.12, apply the recovery function specified in Appendix B on the ICC Public Key Certificate using the Issuer Public Key in conjunction with the corresponding algorithm. If the Recovered Data Trailer is not equal to 'BC', Dynamic Data Authentication fails.

TABLE 2.12: FORMAT OF THE DATA RECOVERED FROM THE ICC PUBLIC KEY CERTIFICATE

Field Name	Length	Description	Format
Recovered Data Header	1	Hex. value '6A'	b
Certificate Format	1	Hex. value '04'	b
Application PAN	10	PAN (padded to the right with hex. 'F's)	cn 20
Certificate Expiration Date	2	MMYY after which this certificate is invalid	n 4
Certificate Serial Number	3	Binary number unique to this certificate assigned by the issuer	b
Hash Algorithm Indicator	1	Identifies the hash algorithm used to produce the Hash Result in the digital signature scheme	b
ICC Public Key Algorithm Indicator	1	Identifies the digital signature algorithm to be used with the ICC Public Key	b
ICC Public Key Length	1	Identifies the length of the ICC Public Key Modulus in bytes	b
ICC Public Key Exponent Length	1	Identifies the length of the ICC Public Key Exponent in bytes	b
ICC Public Key or Leftmost Digits of the ICC Public Key	$N_I - 42$	The $N_I - 42$ most significant bytes of the ICC Public Key	b
Hash Result	20	Hash of the ICC Public Key and its related information	b
Recovered Data Trailer	1	Hex. value 'BC'	b

3. Check the Recovered Data Header. If it is not '6A', Dynamic Data Authentication fails.

4. Check the Certificate Format. If it is not '04', Dynamic Data Authentication fails.

5. Concatenate from left to right the second data element through the tenth data element in Table 2.12 ('Certificate Format' through 'ICC Public Key or Leftmost Digits of the ICC Public Key'), followed by the ICC Public Key Remainder, the ICC Public Key Exponent and finally the static data to be authenticated specified in Table 2.3 and Table 2.4.

6. Apply the indicated hash algorithm (derived from the Hash Algorithm Indicator) to the result of the concatenation of the previous step to produce the hash result.

7. Compare the calculated hash result from the previous step with the recovered hash result. If they are not the same, Dynamic Data Authentication fails.

8. Check if the recovered PAN is equal to the Application PAN, read from the ICC. If not, Dynamic Data Authentication fails.

9. Verify that the last day of the month specified in the Certificate Expiration Date is equal to or later than today's date. If not, Dynamic Data Authentication fails.

10. If the ICC Public Key Algorithm Indicator is not recognized, Dynamic Data Authentication fails.

11. If all the checks above are correct, concatenate the Leftmost Digits of the ICC Public Key and the ICC Public Key Remainder to obtain the ICC Public Key Modulus, and continue with the actual Dynamic Data Authentication described in the two sections below.

2.2.5 Dynamic Signature Generation

1. After successfully retrieving the ICC Public Key as described above, the terminal issues an INTERNAL AUTHENTICATE command including the concatenation of the data elements specified by the Dynamic Data Object List (DDOL) according to the rules specified in Part II of the *EMV'96: IC Card Specification for Payment Systems.*

The ICC may contain the DDOL, but there is a default DDOL in the terminal, specified by the Payment System, in case the DDOL is not present in the ICC.

The DDOL always contains a 4 byte Unpredictable Number (UN) generated by the terminal.

If any of the following cases occur, Dynamic Data Authentication fails:

- Both the ICC and the terminal do not contain a DDOL
- The DDOL in the ICC does not include the Unpredictable Number
- The ICC does not contain a DDOL and the default DDOL in the terminal does not include the Unpredictable Number. Table 2.13 specifies the length and format of the data elements which are in the DDOL.

TABLE 2.13: DYNAMIC DATA OBJECT LIST

Value	Tag	Presence	Length	Format
Unpredictable number	'9F37'	M		b
Terminal Identification	'9F1C'	O	8	an 8
Terminal Country Code(CRF075)	'9F1A'	O	2	n 3
Transaction Date	'9A'	O	3	n 6

2. The ICC generates a digital signature as described in Appendix B on the data specified in Table 2.14 using the ICC Private Key 'S_{IC}' in conjunction with the corresponding algorithm. The result is called the Signed Dynamic Application Data.

TABLE 2.14: DYNAMIC APPLICATION DATA TO BE SIGNED

Field Name	Length	Description	Format
Signed Data Format	1	Hex. value '05'	b
Hash Algorithm Indicator	1	Identifies the hash algorithm used to produce the Hash Result	b
ICC Dynamic Data Length	1	Identifies the length L_{DD} of the ICC dynamic data in bytes	b
ICC Dynamic Data	L_{DD}	Dynamic data generated by and/or stored in the ICC	—
Pad Pattern	$N_{IC} - L_{DD} - 25$	($N_{IC} - L_{DD} - 25$) padding bytes of value 'BB'	b
Terminal Dynamic Data	var.	Concatenation of the data elements specified by the DDOL	—

The ICC Dynamic Data field consists of the concatenation of the length of the ICC Dynamic Number and an 8 byte Dynamic Number as specified in Table 2.15 ($L_{DD} = 9$). Section 2 describes how the ICC Dynamic Number is computed during Dynamic Data Authentication.

TABLE 2.15: ICC DYNAMIC NUMBER

Field Name	Length	Description	Format
ICC Dynamic Number	8	Time-variant number generated by the ICC to be captured by the terminal	b

TABLE 2.16: ICC DYNAMIC DATA

Name	Length	Description		Format
ICC Dynamic Data	9 (=L_{DD})	08	ICC Dynamic Number	b

In addition to those data objects specified in Table 2.10, the data objects required for Dynamic Data Authentication are specified in Table 2.17.

TABLE 2.17: ADDITIONAL DATA OBJECTS REQUIRED FOR DYNAMIC SIGNATURE GENERATION AND VERIFICATION

Tag	Length	Value	Format
'9F4B'	N_{IC}	Signed Dynamic Application Data	b
'9F49'	var.	DDOL	—

2.2.6 Dynamic Signature Verification

1. If the Signed Dynamic Application Data has a length different from the length of the ICC Public Key Modulus, Dynamic Data Authentication fails.

2. To obtain the recovered data specified in Table 2.18, apply the recovery function specified in Appendix B to the Signed Dynamic Application Data using the ICC Public Key in conjunction with the corresponding algorithm. If the Recovered Data Trailer is not equal to 'BC', Dynamic Data Authentication fails.

TABLE 2.18: FORMAT OF THE DATA RECOVERED FROM THE SIGNED DYNAMIC APPLICATION DATA

Field Name	Length	Description	Format
Recovered Data Header	1	Hex. value '6A'	b
Signed Data Format	1	Hex. value '05'	b
Hash Algorithm Indicator	1	Identifies the hash algorithm used to produce the Hash Result in the digital signature scheme	b
ICC Dynamic Data Length	1	Identifies the length of the ICC dynamic data in bytes	b
ICC Dynamic Data	L_{DD}	Dynamic data generated by and/or stored in the ICC	—
Pad Pattern	$N_{IC} - L_{DD} - 25$	$(N_{IC} - L_{DD} - 25)$ padding bytes of value 'BB'	b
Hash Result	20	Hash of the Dynamic Application Data and its related information	b
Recovered Data Trailer	1	Hex. value 'BC'	b

3. Check the Recovered Data Header. If it is not '6A', Dynamic Data Authentication fails.

4. Check the Signed Data Format. If it is not '05', Dynamic Data Authentication fails.

5. Concatenate, from left to right, the second data element through the sixth data element in Table 2.18 ('Signed Data Format' through 'Pad Pattern'), followed by the data elements specified by the DDOL.

6. Apply the indicated hash algorithm (derived from the Hash Algorithm Indicator) to the result of the concatenation of the previous step to produce the hash result.

7. Compare the calculated hash result from the previous step with the recovered hash result. If they are not the same, Dynamic Data Authentication fails.

8. If all the above steps were executed successfully, Dynamic Data Authentication passes. The terminal stores the retrieved ICC Dynamic Number for further processing as described in the *EMV'96 ICC Specification for Payment Systems.*

2.3 PIN ENCIPHERMENT

If supported, Personal Identification Number (PIN) encipherment for offline PIN verification is performed by the terminal using an asymmetric based encipherment mechanism to ensure the secure transfer of a PIN from a secure tamper-evident PIN pad to the ICC. The ICC owns a public key pair associated with PIN encipherment. The public key is used by the PIN pad to encipher the PIN, and the private key is used by the ICC to verify the enciphered PIN.

2.3.1 Keys and Certificates

There are two options for key usage:

1. The ICC owns a specific ICC PIN Encipherment Private and Public Key. The ICC PIN Encipherment Public Key is stored on the ICC in a public key certificate in exactly the same way as the ICC Public Key for dynamic data authentication as specified in Section 2.2.

   The ICC PIN Encipherment public key pair has an ICC PIN Encipherment Public Key Modulus of N_{PE} bytes, where $N_{PE} < N_I$, N_I is the length of the Issuer Public Key Modulus (see Section 2.1). If $N_{PE} > (N_I - 42)$, the ICC PIN Encipherment Public Key Modulus is divided into two parts:

   - One part consisting of the $N_I - 42$ most significant bytes of the modulus (the Leftmost Digits of the ICC PIN Encipherment Public Key)
   - Second part consisting of the remaining $N_{PE} - (N_I - 42)$ least significant bytes of the modulus (the ICC PIN Encipherment Public Key Remainder)

   The ICC PIN Encipherment Public Key Exponent is equal to 3 or $2^{16}+1$. The ICC PIN Encipherment Public Key Certificate is obtained by applying the digital signature scheme specified in Appendix B to the data in Table 2.19 using the Issuer Private Key (see Section 2.1).

TABLE 2.19: ICC PIN ENCIPHERMENT PUBLIC KEY DATA TO BE SIGNED BY THE ISSUER

Field Name	Length	Description	Format
Certificate Format	1	Hex. Value '04'	b
Application PAN	10	PAN (padded to the right with hex. 'F's)	cn 20
Certificate Expiration Date	2	MMYY after which this certificate is invalid	n 4
Certificate Serial Number	3	Binary number unique to this certificate assigned by the issuer	b
Hash Algorithm Indicator	1	Identifies the hash algorithm used to produce the Hash Result in the digital signature scheme	b
ICC PIN Encipherment Public Key Algorithm Indicator	1	Identifies the digital signature algorithm to be used with the ICC PIN Encipherment Public Key	b
ICC PIN Encipherment Public Key Length	1	Identifies the length of the ICC PIN Encipherment Public Key Modulus in bytes	b
ICC PIN Encipherment Public Key or Leftmost Digits of the ICC PIN Encipherment Public Key	$N_I - 42$	If $N_{PE} \leq N_I - 42$, this field consists of the full ICC PIN Encipherment Public Key padded to the right with $N_I - 42 - N_{PE}$ bytes of value 'BB' If $N_{PE} > N_I - 42$, this field consists of the $N_I - 42$ most significant bytes of the ICC PIN Encipherment Public Key [4]	b
ICC PIN Encipherment Public Key Remainder	0 or $N_{IC} - N_I + 42$	This field is only present if $N_{PE} > N_I - 42$ and consists of the $N_I - N_{PE} + 42$ least significant bytes of the ICC PIN Encipherment Public Key	b
ICC PIN Encipherment Public Key Exponent	1 or 3	ICC PIN Encipherment Public Key Exponent equal to 3 or $2^{16}+1$	b

4 As can be seen in Appendix B, $N_I - 22$ bytes of the data signed are retrieved from the signature. Since the first through the eighth data elements in Table 2.9 total 20 bytes, there are $N_I - 22 - 20 = N_I - 42$ bytes left for the data to be stored in the signature.

2. The ICC does not own a specific ICC PIN Encipherment public key pair, but owns an ICC public key pair for Dynamic Data Authentication as specified in Section 2.2. This key pair is used for PIN encipherment, if and only if the ICC Public Key Exponent is equal to 3 or $2^{16}+1$. The ICC Public Key is stored on the ICC in a public key certificate as specified in Section 2.2.

The first step of PIN Encipherment is the retrieval of the public key to be used by the terminal for the encipherment of the PIN. This process takes place as:

1. If the terminal has obtained all the data objects specified in Table 2.20 from the ICC, then the terminal retrieves the ICC PIN Encipherment Public Key in exactly the same way as it retrieves the ICC Public Key for Dynamic Data Authentication (see Section 2.4).

2. If the terminal has not obtained all the data objects specified in Table 2.20, but has obtained all the data objects specified in Table 2.10, and the ICC Public Key Exponent is equal to 3 or $2^{16}+1$, then the terminal retrieves the ICC Public Key as described in Section 2.4.

3. If the conditions under points 1 and 2 above are not satisfied, then PIN Encipherment fails.

TABLE 2.20: DATA OBJECTS REQUIRED FOR THE RETRIEVAL OF THE ICC PIN ENCIPHERMENT PUBLIC KEY

Tag	Length	Value	Format
-	5	Registered Application Provider Identifier (RID)	b
'8F'	1	Certification Authority Public Key Index	b
'90'	N_{CA}	Issuer Public Key Certificate	b
'92'	$N_I - N_{CA} + 36$	Issuer Public Key Remainder, if present	b
'9F32'	1 or 3	Issuer Public Key Exponent	b
'9F2D'	N_I	ICC PIN Encipherment Public Key Certificate	b
'9F2E'	$N_{PE} - N_I + 42$	ICC PIN Encipherment Public Key Remainder, if present	b
'9F2F'	1 or 3	ICC PIN Encipherment Public Key Exponent	b

2.3.2 PIN Encipherment and Verification

The exchange and verification of an enciphered PIN between terminal and ICC takes place as:

1. The PIN is entered in plain text format on the PIN pad and a PIN block is constructed as defined in the *EMV'96 ICC Specification for Payment Systems.*

2. The terminal issues a GET CHALLENGE command to the ICC to obtain an 8-byte unpredictable number from the ICC.

3. The terminal generates a Random Pad Pattern consisting of N – 17 bytes, where N is the length in bytes of the public key to be used for PIN encipherment retrieved as specified in Section 2.3 (hence $N = N_{PE}$ or $N = N_{IC}$).

4. Using the PIN Encipherment Public Key or the ICC Public Key retrieved as specified in Section 2.3, the terminal applies the RSA Recovery Function specified in Appendix B to the data specified in Table 2.21 to obtain the Enciphered PIN Data.

TABLE 2.21: DATA TO BE ENCIPHERED FOR PIN ENCIPHERMENT

Field Name	Length	Description	Format
Data Header	1	Hex. value '7F'	b
PIN Block	8	PIN in PIN Block	b
ICC Unpredictable Number	8	Unpredictable number obtained from the ICC with the GET CHALLENGE command	b
Random Pad Pattern	N – 17	Random Pad Pattern generated by the terminal	b

5. The terminal issues a VERIFY command including the Enciphered PIN Data obtained in the previous step.

6. With the ICC Private Key, the ICC applies the RSA Signing Function algorithm specified in Appendix B to the Enciphered PIN Data to recover the plain text data specified in Table 2.21.

7. The ICC verifies if the Data Header recovered is equal to '7F'. If the Data Header recovered is not equal to '7F', PIN verification fails.

8. The ICC verifies whether the ICC Unpredictable Number recovered is equal to the ICC Unpredictable Number generated by the ICC with the GET CHALLENGE command. If the ICC Unpredictable Number recovered is not equal to the ICC Unpredictable Number generated by the ICC (with the GET CHALLENGE command), PIN verification fails.

9. The ICC verifies whether the PIN included in the recovered PIN Block corresponds with the PIN stored in the ICC. If the PIN included in the recovered PIN Block does not correspond with the PIN stored in the ICC, PIN verification fails.

10. If all the above steps were executed successfully, enciphered PIN verification passes.

☞ **For this mechanism to be secure, the steps 1 through 4 are executed in the secure environment of the tamper-evident PIN pad.**

2.4 APPLICATION CRYPTOGRAMS

This section describes how to generate a TC, AAC or ARQC.

2.4.1 Initial Selection of Data

A data object to be input to the TC, AAC or ARQC algorithm is either:

- Referenced in the ICC's CDOLs and transmitted in plain text from the terminal to the ICC in the GENERATE AC command

or

- Accessed internally by the ICC.

☞ **Only certain data elements (for example, data elements and data objects retrievable by the GET PROCESSING OPTIONS command) can be accessed internally by the ICC.**

This specification supports a limited set of methods to generate an application cryptogram. Each method is identified by a Cryptogram Version Number. The Cryptogram Version Number indicates the data objects used to generate the application cryptogram and the method of data input for each data object. (See Appendix D for other data elements which could be included in the CDOL.) Table 2.22 lists the 11 objects recommended by MasterCard for input to application cryptograms and the source of the each data object.

☞ **The algorithm described is the recommended implementation. It is referenced as cryptogram version number x. Issuers that wish to use different algorithms (for example those issuers that do not require session keys in the cryptogram) can be accommodated by establishing a different cryptogram version number for their algorithm. It is strongly recommended that issuers contact MasterCard before committing to an unspecified cryptogram version number.**

TABLE 2.22: TC, AAC, AND ARQC DATA ELEMENTS

Data Element	Supplied by Terminal	Input by ICC
Amount, Authorized	X	
Amount, Other	X	
Terminal Country Code	X	
Terminal Verification Results	X	
Transaction Currency Code	X	
Transaction Date	X	
Transaction Type	X	
Unpredictable Number	X	
Application Interchange Profile		x
ATC		x
Card Verification Results		x

If a data object is included as input to the algorithm, the data object is used in the order specified in Table 2.22.

It the Issuer wants to include other data elements than the ones recommended by MCI/EPI, they should be taken from the data elements listed in Appendix D, all of which have an EMV Tag.

2.4.2 TC, AAC, and ARQC Algorithm

1. In the first step, the terminal issues the first/second GENERATE AC , using the data specified in CDOL1/CDOL2.

2. The second step of the AC generation consists of deriving a 16-byte Session Key SK_{AC} from the ICC Master Key Cryptogram Key MK_{AC} using (see Table 2.23).

   - The 2-byte Application Transaction Counter (ATC) of the ICC
   - A 4-byte terminal Unpredictable Number (UN)

   and the Session Key Derivation (SKD) function described in 2.7.

☞ **This UN is not necessarily the same UN that was possibly used previously for Dynamic Data Authentication (see Section 2) or the enciphered PIN (see Section 2).**

☞ **SK_{AC}: = SKD(MK_{AC})[(ATC || '00' || '00' ||UN)].**

TABLE 2.23: DATA ELEMENTS INVOLVED IN THE APPLICATION CRYPTOGRAM SESSION KEY DERIVATION

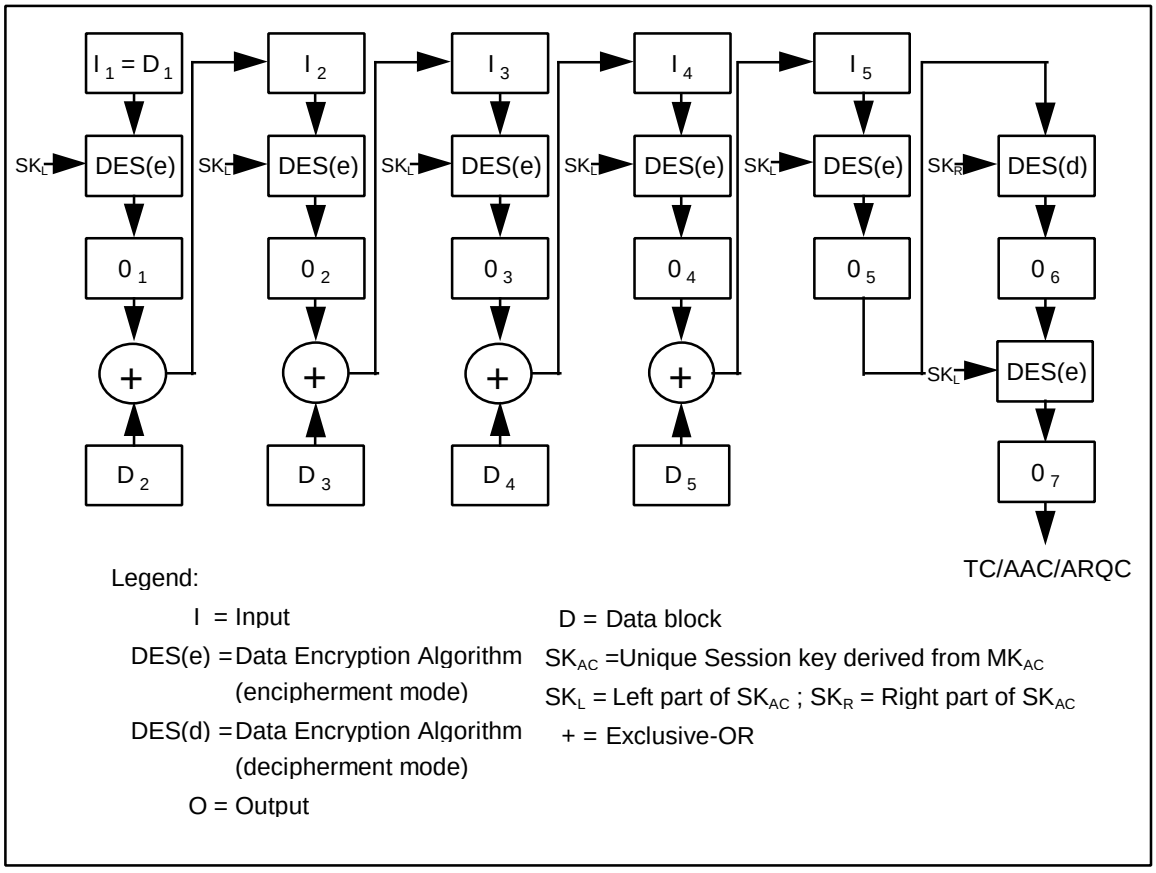
Value	Source	Tag	Length	Format
Application Transaction Counter (ATC)	ICC	'9F36'	2	b
Unpredictable Number (UN)	Terminal	'9F37'	4	b

3. Based upon card risk management, the ICC determines whether to return a TC, an AAC or an ARQC in the response message. The data to be used as input for the cryptogram algorithm is hard coded in the ICC. The ICC concatenates the data specified in Table 2.22 in the order specified to create a block of data.

4. The ICC formats this block of data into 8-byte data blocks, labelled D_1, D_2, D_3, D_4, D_5. The ICC pads this block according to method 2 of ISO/IEC 9797, hence add a mandatory byte with hexadecimal '80' to the right and then add the smallest number of hexadecimal '00' bytes to the right such that the length of the resulting message $\underline{M}$:= (M || '80' || '00' || '00' || . . . || '00') is a multiple of 8 bytes.

5. The ICC performs the algorithm shown in Figure 3 to generate the TC, AAC or ARQC using the Session keys SK_{AC}. The TC, AAC or ARQC is the 8 byte final result.

FIGURE 2.3: AC ALGORITHM FOR A DOUBLE-LENGTH SESSION KEY



In the above figure, the number of cycles can be expanded beyond five

2.5 ISSUER AUTHENTICATION

The validation process for the ARPC is described:

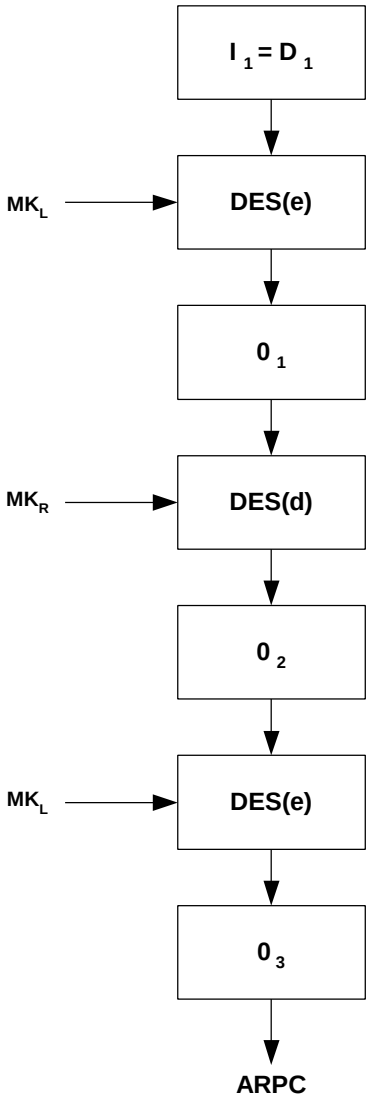
1. The ICC shall perform an exclusive-OR operation:
   ARQC ⊕ ARPC Response Code

   The ARPC Response Code (ARC) used in this operation is transmitted to the ICC in the Issuer Authentication Data. Prior to performing the exclusive-OR operation, the ICC left-justifies the ARPC Response Code in an 8-byte field and zero fills (hexadecimal zero) the remaining 6 bytes.

2. The results of the exclusive-OR operation is used as input to the 8-byte data block D_1.

3. The ICC performs the authentication algorithm as shown in Figure 2.4 to generate the ARPC using the ICC Master Key Cryptogram Key MK_{AC}. The ICC shall generate the ARPC by applying the DES3 algorithm to the result of the exclusive-OR operation in step 1. The ARPC is the 8 byte final result.

FIGURE 2.4: ARPC ALGORITHM FOR A DOUBLE-LENGTH DES KEY



Legend:

I = Input
DES(e) = Data Encryption Algorithm (encipherment mode)
DES(d) = Data Encryption Algorithm (decipherment mode)
O = Output
D = Data block
MK_L = Left part of MK_{AC}
MK_R = Right part of MK_{AC}

☞ **There is no need to recalculate the ARQC to verify the ARPC. The ARQC transmitted in the request message is used as input to the exclusive-OR operation.**

2.6 SECURE MESSAGING

Although Secure Messaging as described here may be used with a command other than the Issuer Script Commands described in Section 1, this section describes the use of Secure Messaging for Issuer Script Commands.

The principle objectives of secure messaging are to ensure:

- Message Integrity
- Data Confidentiality
- Issuer authentication

Data confidentiality is achieved using encipherment of the plain text command data. Message integrity and issuer authentication are achieved using a MAC.

2.6.1 Secure Messaging for Integrity

The Secure Messaging format defined in this specification is based upon ISO 7816-4. Secure Messaging is indicated for any *EMV'96* command that may require it when the second nibble of the CLA byte is equal to hexadecimal '4'.

Data encipherment is done in accordance with the *EMV'96* specification with the provision that the length field in front of the enciphered data is omitted.

The MAC is the last data element in the command data field and has a length of 8 bytes.

The length of the MAC is known by the originator of the command and by the currently selected application in the ICC.

The MAC Session Key used during Secure Message processing is generated using the session key generation process described in Section 2.

The MAC is generated using DES3 encipherment:

1. An initial vector is set equal to 8 bytes of hexadecimal zeroes.

2. The following data is concatenated in the order specified to create a block of data:
   - CLA, INS, P1, P2, Lc
   - Last ATC (for Issuer Script processing, this is the ATC transmitted in the request message)
   - The RAND used for session key derivation

☞ **For Issuer Script processing, this is usually the ARQC transmitted in the request message; when the application has been blocked prior to transmission of the request, the Application Cryptogram is an AAC.**

   - Plain text data contained in the command data field (if present)

3. This block of data is formatted into 8-byte data blocks, labelled D_1, D_2, D_3, D_4, etc. The last data block may be 1-8 bytes in length.

4. If the last data block is 8 bytes in length, an additional 8-byte data block is concatenated to the right of the last data block: hexadecimal '80 00 00 00 00 00 00 00'. Proceed to step 5.

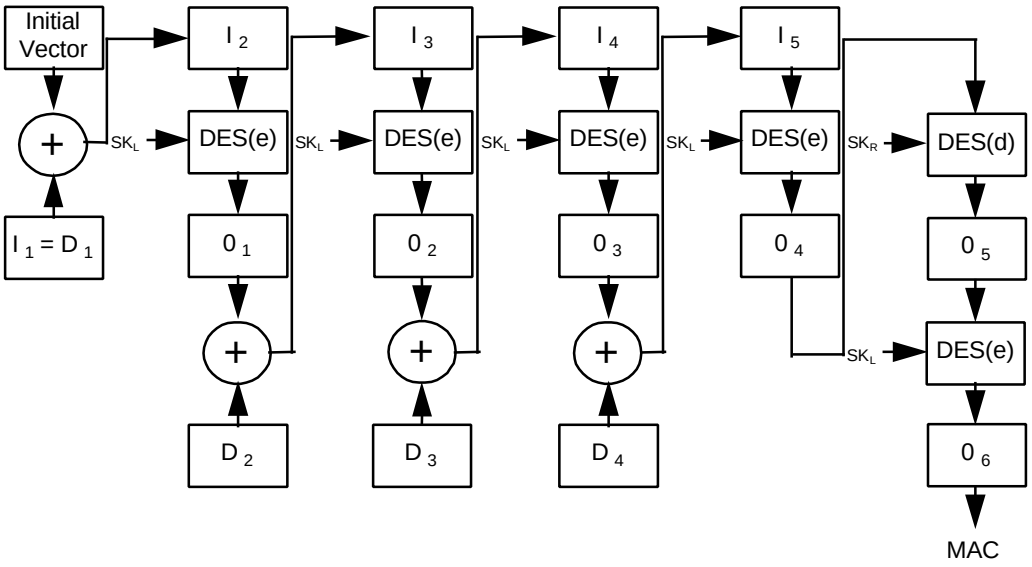
   If the last data block is less than 8 bytes in length, it is padded to the right with a 1-byte hexadecimal '80'. If the last data block is now 8 bytes in length, proceed to step 5. If the last data block is still less than 8 bytes in length, it is right filled with 1-byte hexadecimal zeroes until it is 8 bytes in length.

5. The MAC is generated using the MAC Session Keys SK_{SMI} as shown in Figure 2.5.

☞ **Depending on the length of the concatenated block of data created in step 2, there may be less than four 8-byte data blocks to input to the algorithm**

6. The resultant value is the 8-byte MAC.

FIGURE 2.5: MAC ALGORITHM FOR A DOUBLE-LENGTH SESSION KEY



Legend:

I = Input

D = Data block

DES(e) = Data Encryption Algorithm (encipherment mode)

DES(d) = Data Encryption Algorithm (decipherment mode)

O = Output

SK_{SMI} = MAC Session Key (Integrity)

SK_L = Left part of SK_{SMI}

SK_R = Right part of SK_{SMI}

\+ = Exclusive - OR

2.6.2 Secure Messaging for Confidentiality

The encryption mechanism used for Secure Messaging for Confidentiality is the DES3 algorithm in CBC mode.

Encryption of a message M of arbitrary length with a double-length Session Key $SK_{SMC} = (SK_L \| SK_R)$ takes place in the following steps (see Section 2).

1. If the length of the message M is not a multiple of 8 bytes, pad the message M according to ISO 7816-4, hence add a mandatory byte with hexadecimal '80' to the right of M, and then add the smallest number of hexadecimal '00' bytes to the right such that the length of resulting message $\underline{M}$:= (M || '80' || '00' || '00' || . . . || '00') is a multiple of 8 bytes.

2. Divide $\underline{M}$ into 8-byte blocks $X_1, X_2, \ldots, X_k$ and encrypt the blocks into the 8-byte blocks $Y_1, Y_2, \ldots, Y_k$ with the T-DES algorithm in CBC mode. Compute for i = 1, 2, . . . , k-1:

$$Y_i := DES3\ (SK_{SMC})[X_i \oplus Y_{i-1}],$$

where $Y_0 := 0$.

☞ **$Y := (Y_1 \| Y_2 \| \ldots \| Y_k) = ENC(SK_{SMC})[M]$**

Decryption of Y with a double-length Session Key $SK_{SMC} = (SK_L \| SK_R)$ takes place in the following steps:

1. Compute for i = 1, 2, . . . , k:

$$X_i := DES3\ (SK_{SMC})[Y_i\ ] \oplus Y_{i-1},$$

where $Y_0 := 0$.

2. To obtain the original message M, concatenate the blocks $X_1, X_2, \ldots, X_k$ and, if padding took place, remove the trailing ('80' || '00' || '00' || . . . || '00') byte-string from the last block X_k.

2.6.3 Combined Integrity and Confidentiality

The MAC is generated using all elements of the script command, including the command header. The integrity of the command, including the data component contained in the command data field (if present), is ensured by Secure Messaging.

When a specific command requires both Secure Messaging for Integrity and Confidentiality (for example for the PIN CHANGE/UNBLOCK command), the following applies:

1. The plain text command data is enciphered as specified in Section 2 to obtain the enciphered command data.

2. The MAC is then computed over the enciphered command data.

3. The command data field consists of the enciphered command data, concatenated to the right with the MAC.

This mechanism enables the usage of the same ICC Master Key for both Secure Messaging for Integrity and Confidentiality.

2.7 ICC KEY DERIVATION

2.7.1 ICC Master Key Derivation

The Debit/Credit application on the ICC uses the following unique 16-byte DES3 keys

- MK_{AC} — ICC Master Key for Application Cryptogram Generation (see Section 2).
- MK_{SMI} — ICC Master Key for Secure Messaging for Integrity (see Section 2).
- MK_{SMC} — ICC Master Key for Secure Messaging for Confidentiality (see Section 2)
- MK_{IDN} — ICC Master Key for ICC Dynamic Number generation (see Section 2).

This section illustrates the method of key derivation used to generate the keys stored in the ICC during personalization.

The ICC Master Keys (MK) are derived from the corresponding Issuer Master Keys (IMK) (see Table 2.24).

TABLE 2.24: ISSUER MASTER KEYS USED TO DERIVE ICC MASTER KEYS

Unique ICC Master Key	Derived from Issuer Master Key
MK_{AC}	IMK_{AC}
MK_{SMI}	IMK_{SMI}
MK_{SMC}	IMK_{SMC}
MK_{IDN}	IMK_{IDN}

Figure 2.6 shows the method for the derivation of ICC unique Master keys.

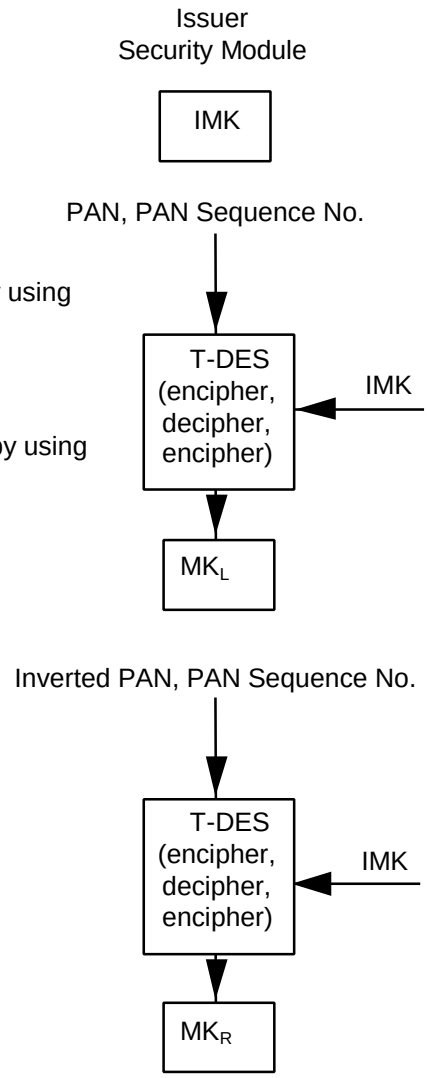
FIGURE 2.6: KEY DERIVATION METHODOLOGY

Issuer generates its double-length Master Key (IMK).

For each application in ICC, issuer personalises ICC with an ICC Master MK.

The left part of Master Key MK is derived by using Application PAN and Application PAN Sequence Number as input and performing T-DES encipherment.

The right part of Master Key MK is derived by using the inverted Application PAN and Application PAN Sequence Number as input and performing T-DES encipherment.



To derive the left part of ICC Master Key MK, the Application PAN and Application PAN Sequence Number are concatenated together in a 16-hexadecimal field. If the combined length of the concatenation of the Application PAN and the Application PAN Sequence Number is not equal to 16 digits, the following formatting rules apply:

- If the combined length of 'Application PAN' plus the 'Application PAN Sequence Number' is less than 16 digits, right-justify the data in a 16-hexadecimal field and pad on the left with '0's;
- If the combined length of 'Application PAN' followed by the 'Application PAN Sequence Number' exceeds 16 digits, use only the rightmost 16 digits.

To derive the right part of the ICC Master Key, the Application PAN and Application PAN Sequence Number are first concatenated together into a 16-hexadecimal field using the formatting rules described above and then inverted. Inversion is performed at the bit level, each bit with value '1' is set to '0' and each bit with value '0' is set to '1'.

2.7.2 ICC Session Key Derivation

The session key derivation function uses a 16-byte Master Key MK and an 8-byte random number $R = (R_0 \| R_1 \| R_2 \| R_3 \| R_4 \| R_5 \| R_6 \| R_7)$, and produces a 16-byte Session Key $SK = (SK_L \| SK_R)$:

$$SK_L := DES3\ (MK)[(R_0 \| R_1 \| \text{'F0'} \| R_3 \| R_4 \| R_5 \| R_6 \| R_7)]$$

$$SK_R := DES3\ (MK)[(R_0 \| R_1 \| \text{'0F'} \| R_3 \| R_4 \| R_5 \| R_6 \| R_7)].$$

The left part of the double-length Session Key is obtained by applying the DES3 algorithm with the double-length ICC Master Key to the 8-byte data-block obtained by replacing the third most significant byte R_2 of R by the byte 'F0'.

The right part of the double-length Session Key is obtained by applying the DES3 algorithm with the double-length ICCMaster Key to the 8-byte data-block obtained by replacing the third most significant byte R_2 of R by the byte '0F'.

☞ **SK: = SKD(MK)[R]**

2.8 RANDOM NUMBER FOR SESSION KEY DERIVATION

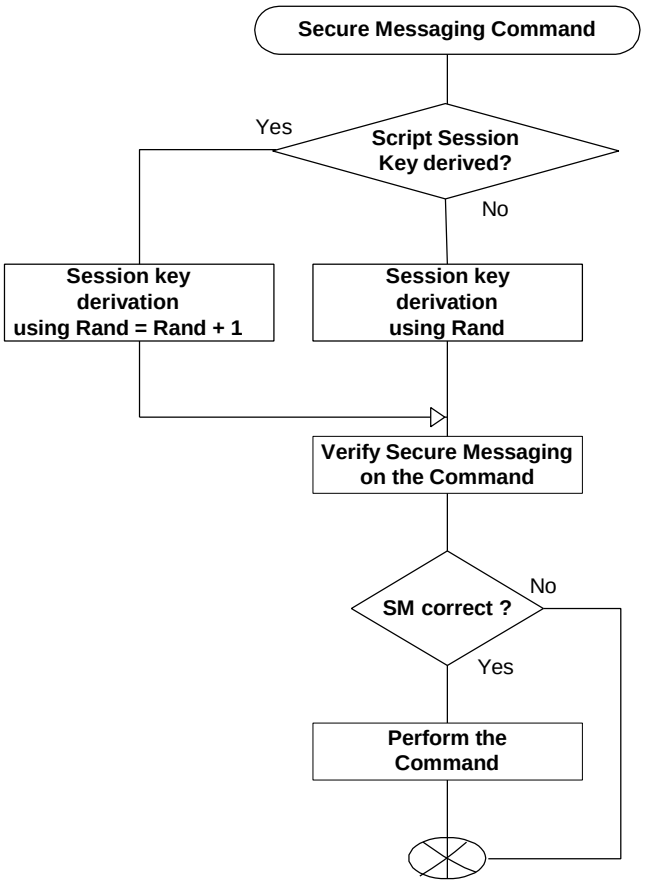
In Secure Messaging for Integrity and Confidentiality, the Session Key used is derived from an ICC Master Key using a random number RAND generated by the ICC. For script processing, the RAND can either be an ARQC or an AAC generated by the ICC in response to a GENERATE AC command.

If more than one subsequent command is sent using Secure Messaging, a separate Session Key is derived for the Secure Messaging of each subsequent command:

- The RAND used for the Session Key derivation for the first command is equal to the ARQC or the AAC

- The RAND for the Session Key derivation of a subsequent command is obtained by incrementing the RAND used for the previous command by 1 (see Figure 2.7).

FIGURE 2.7: DIAGRAM OF A SECURE MESSAGING COMMAND



2.9 DATA AUTHENTICATION CODE GENERATION

The Data Authentication Code (DAC) is retrieved by the terminal during Static Data Authentication. The Data Authentication Code is sent to the Issuer as proof that the terminal executed SDA successfully.

This section describes how the Issuer computes the Data Authentication Code (DAC) based on ICC specific data for personalization and verification.

A 16-byte DES3 Issuer Master Key IMK_{DAC} is used by the Issuer for the purpose of generating a DAC.

The 2-byte DAC is equal to the 2 leftmost bytes of Z, which is given by

$$Z := DES3\ (IMK_{DAC})[Y]$$

where the 8-byte data block Y is obtained:

- If the combined length of 'Application PAN' plus the 'Application PAN Sequence Number' is less than 16 digits, right-justify the data in a 16-hexadecimal field and pad on the left with hexadecimal '0's

- If the combined length of 'Application PAN' followed by the 'Application PAN Sequence Number' exceeds 16 digits, use only the rightmost 16 digits.

☞ **For security reasons IMK_{DAC} must be distinct from the Issuer Master Keys specified in Table 2.24.**

2.10 ICC DYNAMIC NUMBER GENERATION

The ICC Dynamic Number (IDN) is computed by the ICC and retrieved by the terminal during Dynamic Data Authentication. The 2 leftmost bytes of the ICC Dynamic Number are sent to the Issuer as proof that the terminal executed DDA successfully.

This section describes how the ICC generates and the Issuer verifies the ICC Dynamic Number.

A 16-byte DES3 ICC Master Key MK_{IDN} is stored in the ICC for the purpose of generating the ICC Dynamic Number.

☞ **This ICC Master Key is unique for each ICC. The key management at Issuer level of these keys is described in Section 2.**

The ICC Dynamic Number is computed using the ICC Application Transaction Counter (ATC) and the Unpredictable Number (UN) requested in the DDOL:

$$IDN := DES3(MK_{IDN})[(ATC \,||\, '00' \,||\, '00' \,||\, '00' \,||\, '00' \,||\, '00' \,||\, '00' \,||)].$$

Table of Contents

SECTION 3 DATA SPECIFICATION OF EMV '96 ICC SPECIFICATION FOR PAYMENT SYSTEMS TRANSACTIONS

3.1 DATA ELEMENTS AND FILES

3.1.1 Management of data elements by ICC

Table 3.1 gives an overview of the management of the ICC data elements, whether they are always present or not, whether they can be updated or not and how the terminal accesses them.

TABLE 3.1: MANAGEMENT OF ICC DATA ELEMENTS

Name	Tag	Presence	Updated by	Access	Condition
Application Currency Code	'9F42'	C	—	READ RECORD —	If the value for either Amount "X" or Amount "Y" contained in the CVM List is nonzero, the Application Currency Code is mandatory Present if the card supports "Maximum Domestic Offline Transaction Amount", "Offline Cumulative Amount" or "Currency conversion"
Application Discretionary Data	'9F05'	O	—	—	
Application Effective Date	'5F25'	O	—	READ RECORD	
Application Expiration Date	'5F24'	M	—	READ RECORD	
Application File Locator	'94'	M	—	GET PROCESSING OPTIONS	
Application Control	'9F60'	O	Script	—	
Application Flag	'9F5E'	M	Internally	—	
Application Identifier (AID)	'4F'	C	—	READ RECORD	If the PSE is present, the AID is mandatory in the ADF entry format
Application Interchange Profile	'82'	M	—	GET PROCESSING OPTIONS	

Name	Tag	Presence	Updated by	Access	Condition
Application Label	'50'	C	—	READ RECORD	The Application Label is mandatory in the ADF entry of the PSE. It is optional in the ADF.
Application Preferred Name	'9F12'	O	—	READ RECORD	
Application Primary Account Number (PAN)	'5A'	M	—	READ RECORD	
Application Primary Account Number (PAN) Sequence Number	'5F34'	M	—	READ RECORD	
Application Priority Indicator	'87'	O	—	READ RECORD SELECT	
Application Template	'61'	C	—		The Application Template is mandatory if the PSE is present
Application Transaction Counter (ATC)	'9F36'	M	Internally	GENERATE AC GET DATA	The ATC is always returned in the response to the GENERATE AC The ATC can be retrieved by the GET DATA command if the card wants the terminal to do the velocity checking
Application Usage Control	'9F07'	M	—	READ RECORD	
Application Version Number	'9F08'	M	—	READ RECORD	
Card Issuer Action Code—Decline	—	M	Script	—	
Card Issuer Action Code—Offline	—	M	Script	—	
Card Issuer Action Code—Online	—	M	Script	—	

Name	Tag	Presence	Updated by	Access	Condition
Card Risk Management Data Object List 1 (CDOL1)	'8C'	M	—	READ RECORD	
Card Risk Management Data Object List 2 (CDOL2)	'8D'	M	—	READ RECORD	
Cardholder Name	'5F20'	M	—	READ RECORD	The Cardholder Name is mandatory if present on the magnetic stripe track 1 and in that case both shall be identical.
Cardholder Name Extended	'9F0B'	O	—	READ RECORD	
Cardholder Verification Method (CVM) List	'8E'	M	—	READ RECORD	
Card TVR Action Code	—	M	Script	—	
Certification Authority Public Key Index	'8F'	C	—	READ RECORD	Mandatory for cards that support offline CAM
Cryptogram Version Number	—	M	—	GENERATE AC	
Data Authentication Code	'9F45'	C	—	READ RECORD	Mandatory for cards that support offline CAM (included in the Signed Static Application Data)
DDF Name	'9D'	C	—	READ RECORD	If the PSE is present, the DDF Name is mandatory if an DDF entry is present
Dedicated File (DF) Name	'84'	M	—	SELECT	
Dynamic Data Authentication Data Object (DDOL)	'9F49'	O	—	READ RECORD	

Name	Tag	Presence	Updated by	Access	Condition
File Control Information (FCI) Issuer Discretionary Data	'BF0C'	O	—	SELECT	
File Control Information (FCI) Proprietary Template	'A5'	M	—	SELECT	
File Control Information (FCI) Template	'6F'	M	—	SELECT	
Integrated Circuit Card (ICC) PIN Encipherment Public Key Certificate	'9F2D'	C	—	READ RECORD	Mandatory if a separate key pair is used for offline PIN encryption. Absent when the same key pair is used as for DDA
Integrated Circuit Card (ICC) PIN Encipherment Public Key Exponent	'9F2F'	C	—	READ RECORD	Mandatory if a separate key pair is used for offline PIN encryption. Absent if the same key pair is used as for DDA
Integrated Circuit Card (ICC) PIN Encipherment Private Key		C	—	—	Present if offline enciphered PIN verification is supported and a dedicated key pair is used for that purpose (ICC private key for DDA is not used for that purpose)
Integrated Circuit Card (ICC) PIN Encipherment Public Key Remainder	'9F2E'	C	—	READ RECORD	Present if a separate key pair is used for offline PIN encryption and $N_{PE} - (N_I - 42) > 0$. Absent if the same key pair is used as for DDA or if $N_{PE} - (N_I - 42) = 0$
Integrated Circuit Card (ICC) Public Key Certificate	'9F46'	C	—	READ RECORD	Mandatory if offline dynamic CAM is supported by the card
Integrated Circuit Card (ICC) Public Key Exponent	'9F47'	C	—	READ RECORD	Mandatory if offline dynamic CAM is to be supported by the card
Integrated Circuit Card (ICC) Private Key		C	—	—	Mandatory if offline dynamic CAM is to be supported by the card

Name	Tag	Presence	Updated by	Access	Condition
Integrated Circuit Card (ICC) Public Key Remainder	'9F48'	C	—	READ RECORD	Present if offline dynamic CAM is to be supported by the card and $N_{IC} - (N_I - 42) > 0$. Absent otherwise
Issuer Action Code—Default	'9F0D'	M	—	READ RECORD	
Issuer Action Code—Denial	'9F0E'	M	—	READ RECORD	
Issuer Action Code—Online	'9F0F'	M	—	READ RECORD	
Issuer Code Table Index	'9F11'	O	—	READ RECORD	
Issuer Country Code	'5F28'	M	—	READ RECORD	
Issuer Public Key Certificate	'90'	C	—	READ RECORD	Mandatory if the card is to support offline CAM
Issuer Public Key Exponent	'9F32'	C	—	READ RECORD	Mandatory if the card is to support offline CAM
Issuer Public Key Remainder	'92'	C	—	READ RECORD	Mandatory if the card is to support offline CAM and $N_I - (N_{CA} - 36) > 0$; Absent otherwise
Issuer Script Command Counter	—	M	Internally	GENERATE AC	Only the least three significant bits are indicated in the CVR
Key Derivation Index	—	M	—	GENERATE AC	
Language Preference	'5F2D'	O	—	READ RECORD	
Last Online Application Transaction Counter (LATC)	'9F13'	C	Internally	GET DATA	Mandatory if the card wants the terminal to do the velocity checking. Mandatory if the card does the velocity checking internally
Lower Consecutive Offline Limit	'9F14'	C	Script	GET DATA —	Mandatory if the card wants the terminal to do the velocity checking. Mandatory if the card does the velocity checking internally

Name	Tag	Presence	Updated by	Access	Condition
Lower Cumulative Domestic Offline Transaction Amount	—	C	Script	—	Mandatory if the card includes the "Maximum Domestic Offline Transaction Amount" function in its card risk management
Maximum Domestic Offline Transaction Amount	—	O	Script	—	Present if the card includes the "Maximum Domestic Offline Transaction Amount" function in its card risk management
MK_{AC}		M	—	—	
MK_{SMI}		C	—	—	Mandatory if the card supports script commands
MK_{SMC}		C	—	—	Mandatory if the card supports a script command that uses Secure Messaging for Confidentiality This key can be the same as MK_{SMI}
MK_{IDN}		C	—	—	Mandatory if offline dynamic CAM is to be supported by the card
Non Domestic Control Factor	—	C	Script	—	Mandatory if the card does the velocity checking internally and wants a different behavior for domestic and non-domestic transactions
Cumulative Offline Transaction	—	C	Internally	—	Present if the card includes the "Offline Cumulative Amount" function in its card risk management
PIN Try Counter	'9F17'	C	Internally	GET DATA VERIFY	Mandatory if the cards supports offline PIN verification and the GET DATA command. Present if the card supports offline PIN verification
PIN Try Limit	—	C	—	—	Mandatory if the card supports offline PIN verification.
Processing Options Data Object List (PDOL)	'9F38'	O	—	SELECT	

Name	Tag	Presence	Updated by	Access	Condition
Reference Currency Conversion Table	—	C	Script	—	Present if the card supports currency conversion
Reference Personal Identification Number (PIN)	—	C	Script	—	Mandatory if the card supports offline PIN verification.
Service Code	'5F30'	O	Script	READ RECORD	
Short File Identifier (SFI)	'88'	C	—	SELECT	Mandatory if PSE is present on the card
Signed Dynamic Application Data	'9F4B'	C	—	READ RECORD	Mandatory if offline dynamic CAM is supported by the card
Signed Static Application Data	'93'	C	—	READ RECORD	Mandatory if offline static CAM is supported by the card
Static Data Authentication Tag List	'9F4A'	C	—	READ RECORD	Present if AIP or AID are include in the Static Data to be Authenticated.
Track 1 Discretionary Data	'9F1F'	C	—	READ RECORD	The Track 1 Discretionary data must be present in the ICC if present on the magnetic stripe and they must be identical.
Track 2 Equivalent Data	'57'	C	Script	READ RECORD	The data included in the Track 2 Equivalent Data must be identical to the data on the Track 2 of the magnetic stripe except for the Issuer proprietary Discretionary Data.(CRF003.6)
Transaction Certificate Data Object List (TDOL)	'97'	O	—	READ RECORD	
Upper Consecutive Offline Limit	'9F23'	C	Script	GET DATA	Mandatory if the card wants the terminal to do the velocity checking. Mandatory if the card does the velocity checking internally
Upper Cumulative Domestic Offline Transaction Amount	—	C	Script	—	Mandatory if the card includes the "Maximum Domestic Offline Transaction Amount" function in its card risk management

3.1.2 EMV data elements

Table 3.2 gives a dictionary of EMV data elements to support financial transactions. This dictionary contains all relevant data elements from Section 2 of the *EMV '96, Version 3.1.1, ICC Specification for Payment Systems* and in the *EMV'96 ICC Terminal Specification for Payment Systems.*

TABLE 3.2 EMV DATA ELEMENTS

Name	Description	Source	Format	Tag	Length	Values
Amount, Authorized (Numeric)	Authorized amount of the transaction.	Terminal	n 12	'9F02'	6	
Amount, Authorized (Binary)	Authorized amount of the transaction (excluding adjustments)	Terminal	b	'81'	4	Should not be used for CDOL1 or CDOL2
Amount, Other (Numeric)	Secondary amount associated with the transaction representing a cashback amount.	Terminal	n 12	'9F03'	6	
Amount, Other (Binary)	A secondary amount associated with the transaction representing a cashback amount	Terminal	b	'9F04'	4	Should not be used for CDOL1 or CDOL2
Amount, Reference Currency	Authorized amount expressed in the reference currency	Terminal	b	'9F3A'	4	
Application Authentication Cryptogram (AAC)	Application cryptogram computed by the card for a declined transaction.	ICC	b	'9F26'	8	see 2.4
Application Currency Code	Indicates the currency in which the account is managed according to ISO 4217	ICC	n 3	'9F42	2	
Application Discretionary Data	Issuer-specified data relating to the card application.	ICC	b	'9F05'	1-32	The use of this data object is reserved for future use by MCI/EPI and issuers are not allowed to use it for private or domestic use

Name	Description	Source	Format	Tag	Length	Values
Application Effective Date	Date from which the card application may be used.	ICC	n 6 YYMMDD	'5F25'	3	
Application Elementary File (AEF) Data Template	Indicates the record template of a record containing data objects.	ICC	var.	'70'	var.	
Application Expiration Date	Date after which the card application expires.	ICC	n 6 YYMMDD	'5F24'	3	
Application File Locator	Indicates the location (SFI, range of records) of the AEFs related to a given application.	ICC	var.	'94'	var. up to 252	Byte 1: bits 8-4 = SFI bits 3-1 = 000 Byte 2: First record number to be read for that SFI (never equal to zero) Byte 3: Last record number to be read for that SFI (shall be greater than or equal to byte 2) Byte 4: Number of consecutive records signed in Signed Application Data, starting with record number in byte 2 (may be equal to zero) Byte 1-4 Can be repeated for other files or subsequences of records within a file
Application Interchange Profile	Indicates the capabilities of the card to support specific functions in the application.	ICC	b 16	'82'	2	Byte 1 bit 8: 1 = Initiate bit 7: 1 = Offline static Data Authentication is supported (this bit is only zero for ATM-only cards)

Name	Description	Source	Format	Tag	Length	Values
						bit 6: 1 = Off line Dynamic Data Authentication is supported bit 5: 1 = Cardholder Verification is supported bit 4: 1 = Terminal Risk Management is to be performed bit 3: 1 = Issuer authentication is supported using EXTERNAL AUTHENTICATE 0 = Issuer authentication is supported using second GENERATE AC bit 2: 1 = RESERVED bit 1: 1 = RESERVED Byte 2 = RESERVED
Application Label	Mnemonic associated with AID according to ISO/IEC 7816-5. Used in application selection.	ICC	an 1-16	'50'	1-16	See *Minimum Card Requirements* for list of all Application Labels.
Application Preferred Name	Preferred mnemonic associated with the AID	ICC	an 1-16	'9F12'	1-16	See *Minimum Card Requirements* for list of all Application Preferred Names.
Application Primary Account Number (PAN)	Valid cardholder account number.	ICC	var. up to cn 19	'5A'	var. up to 10	
Application Primary Account Number (PAN) Sequence Number	Identifies and differentiates card applications with the same PAN.	ICC	n 2	'5F34'	1	
Application Priority Indicator	Indicates the priority of a given application or group of applications in a directory.	ICC	b 8	'87'	1	bit 8: 1= Application shall not be selected without confirmation of cardholder 0 =

Name	Description	Source	Format	Tag	Length	Values
						Application may be selected without confirmation of cardholder bits 7-5: RESERVED bits 4-1:0000 = No priority assigned xxxx = Order in which the application is to be listed or selected, ranging from 1 to 15, with 1 being the highest priority
Application Template	Template containing one or more data objects relevant to an application directory entry according to ISO/IEC 7816-5.	ICC	b	'61'	var. up to 252	Only present if the card has a PSE
Application Transaction Counter (ATC)	Counter maintained by the application in the card.	ICC	b	'9F36'	2	Initial value is zero. It is incremented by 1 each time the GET PROCESSING OPTIONS command is executed.
Application Usage Control	Indicates issuer-specified restrictions on the geographic usage and services allowed for the card application.	ICC	b	'9F07'	2	Byte 1: bit 8: 1 = Valid for domestic cash transactions bit 7: 1 = Valid for international cash transactions bit 6: 1 = Valid for domestic goods bit 5: 1 = Valid for international goods bit 4: 1 = Valid for domestic services bit 3: 1 =

Name	Description	Source	Format	Tag	Length	Values
						Valid for international services bit 2: 1 = Valid at ATMs bit 1: 1 = Valid at terminals other than ATMs Byte 2: bit 8: 1 = Domestic cashback allowed bit 7: 1 = International cashback allowed bits 6-1 = RESERVED See *Minimum Card Requirements* for the different settings of each product.
Application Version Number	Version number assigned by the payment system for the application.	ICC	b	'9F08'	2	For this version of the specifications, the version number should be '0002'
Authorization Request Cryptogram (ARQC)	Value computed by the card application for online card authentication.	ICC	b	'9F28'	8	see 2.4
Authorization Response Code	Indicates the disposition of the transaction.	Issuer or terminal	an 2	'8A'	2	Codes generated by the issuer are as indicated in ISO 8583:1987. The following codes are generated by the terminal for the following exception conditions: Y1= Offline approved Z1 = Offline declined Y3 = Unable to go

						online (offline approved) Z3 =Unable to go online (offline declined) If the terminal did not receive an answer from the Issuer, the data element will be (hex) zero filled. Other values are considered as Issuer response, possibly modified by intermediate networks.
Card Risk Management Data Object List 1 (CDOL1)	List of data objects (tag and length) to be passed to the card application with the first GENERATE AC command.	ICC	an	'8C'	var.	
Card Risk Management Data Object List 2 (CDOL2)	List of data elements (tag and length) to be passed to the card application with the second GENERATE AC command.	ICC	an	'8D'	var.	
Cardholder Name	Indicates cardholder name according to ISO 7813.	ICC	ans 2-26	'5F20'	2-26	
Cardholder Name Extended	Indicates the whole cardholder name when greater than 26 characters using the same coding convention as in ISO 7813	ICC	ans 27-45	'9F0B'	27-45	
Cardholder Verification Method (CVM) List	Identifies a prioritized list of methods of verification of the cardholder supported by the card application.	ICC	b	'8E'	var. up to 252	Bytes 1-4 Amount ('X') Bytes 5-8: Amount ('Y')

Name	Description	Source	Format	Tag	Length	Values
						Byte 9 (CVM Method Codes): bit 8: 0 = Only value bit 7: 0 = Fail cardholder verification if this CVM is unsuccessful 1 = Apply succeeding CVM field if this CVM is unsuccessful bits 6-1: 000000 = Fail CVM processing 000001 =Plaintext PIN verification performed by the ICC 000010 = Enciphered PIN verified online 000011 = Plaintext PIN verification performed by ICC and signature (paper) 000100 = Enciphered PIN verification performed by ICC 000101 = Enciphered PIN verification performed by ICC and signature (paper)000110-011101 = RESERVED 011110 = Signature 011111 = No CVM required 100000-101111 = RESERVED by the individual Payment Systems 110000-111110 = RESERVED by the Issuer 111111 = RESERVED

Name	Description	Source	Format	Tag	Length	Values
						Byte 10 (CVM Conditions): 00 = Always 01 = If cash or cashback (includes quasi-cash) 02 = If not cash or cashback (includes quasi-cash) 03 = If terminal supports the CVM 04 = If terminal does not support the CVM 05 = If terminal support not operative 06 = If transaction is in Application Currency Code and is under X value 07 = If transaction is in Application Currency Code and is over X value 08 = If transaction is in Application Currency Code and is under Y value 09 = If transaction is in Application Currency Code and is over Y value An additional 2 bytes is added following byte 10 for each additional CVM and corresponding CVM Condition. For the CVM lists of the different products, refer to *Minimum Card Requirements*

Name	Description	Source	Format	Tag	Length	Values
Certification Authority Public Key Index	Identifies the certification authority's public key in conjunction with the RID for use in static data authentication.	ICC	b 8	'8F'	1	Values assigned by Card Schemes.
Command Template	Identifies the data field of a command message.	Terminal	b	'83'	var.	Used in Get Processing Options response
Cryptogram Information Data	Indicates the type of cryptogram and the actions to be performed by the terminal	ICC	b	'9F27'	1	bit 8-7: 00 = AAC 01 = TC 10 = ARQC 11 = Not used bit 6-5: RESERVED bit 4: 1 = Advice required 0 = No advice required bit 3-1: (Reason/advice/ referral code) 000 = No information given 001 = Service not allowed 010 = PIN Try Limit exceeded 011 = Issuer authentication failed xxx = All other values are RESERVED
Data Authentication Code	Issuer-assigned value contained in Signed Application Data used to 'prove' that static data authentication was performed.	ICC	b	'9F45'	2	see 2.9
Dedicated File (DF) Name	Identifies the name of the DF as described in ISO/IEC 7816-4.	ICC	b	'84'	5-16	

Name	Description	Source	Format	Tag	Length	Values
Directory Definition File (DDF) Name	Identifies the name of a DF associated with a directory	ICC	b	'9D'	5-16	
Dynamic Data Authentication Data Object (DDOL)	List of data objects (tag and length) to be passed to the ICC in the INTERNAL AUTHENTICATE command	ICC	b	'9F49'	up to 252	
File Control Information (FCI) Issuer Discretionary Data	Issuer discretionary part of the FCI	ICC	var.	'BF0C'	var. up to 222	
File Control Information (FCI) Proprietary Template	Identifies the data object proprietary to this specification in the FCI template according to ISO/IEC 7816-4	ICC	var.	'A5'	var.	
File Control Information (FCI) Template	Identifies the FCI template according to ISO/IEC 7816-4	ICC	var.	'6F'	var. up to 252	
Integrated Circuit Card (ICC) Dynamic Number	Time variant number generated by the card to be retrieved by the terminal during dynamic data authentication	ICC	b	'9F4C'	8	
Integrated Circuit Card (ICC) PIN Encipherment Public Key Certificate	ICC PIN Encipherment Public Key certified by the issuer	ICC	b	'9F2D'	N_I	
Integrated Circuit Card (ICC) PIN Encipherment Public Key Exponent	ICC PIN Encipherment Public Key Exponent used for PIN Encipherment	ICC	b	'9F2F'	1 or 3	
Integrated Circuit Card (ICC) PIN Encipherment Public Key Remainder	Remaining digits of the ICC PIN Encipherment Public Key Modulus	ICC	b	'9F2E'	$N_{PE} - N_I + 42$	

Name	Description	Source	Format	Tag	Length	Values
Integrated Circuit Card (ICC) Public Key Certificate	ICC Public Key certified by the issuer	ICC	b	'9F46'	N_I	
Integrated Circuit Card (ICC) Public Key Exponent	ICC Public Key Exponent used for the verification of the Signed Dynamic Application Data	ICC	b	'9F47'	1 or 3	
Integrated Circuit Card (ICC) Public Key Remainder	Remaining digits of the ICC Public Key Modulus	ICC	b	'9F48'	$N_{IC} - N_I + 42$	
Issuer Action Code—Default	Specifies the issuer's conditions that cause a transaction to be declined if it might have been approved online, but the terminal is unable to process the transaction online.	ICC	b	'9F0D'	5	Bit assignments are identical to those for Terminal Verification Results. The settings for the different products are described in *Minimum Card Requirements*
Issuer Action Code—Denial	Specifies the issuer's conditions that cause the decline of a transaction without attempting to go online.	ICC	b	'9F0E'	5	Bit assignments are identical to those for Terminal Verification Results. The settings for the different products are described in *Minimum Card Requirements*
Issuer Action Code—Online	Specifies the issuer's conditions that cause a transaction to be transmitted online.	ICC	b	'9F0F'	5	Bit assignments are identical to those for Terminal Verification Results. The settings for the different products are described in *Minimum Card Requirements*

Name	Description	Source	Format	Tag	Length	Values
Issuer Application Data	Contains proprietary application data for transmission to the issuer in an online transaction	ICC	b	'9F10'	var. up to 32	For this version of the specifications, the Issuer Application Data is 8 bytes: • Key Derivation Index (1 byte) • Cryptogram Version Number (1 byte) • CVR: information about the ICC decisions taken during the CRM processing. (4 bytes) • DAC or 2 leftmost bytes of ICC Dynamic Number: proving that the terminal correctly performed Static or Dynamic Data Authentication
Issuer Authentication Data	Issuer data transmitted to card for online issuer authentication.	Issuer	b	'91'	8-16	For this version of the specifications, the Issuer Authentication Data is 10 bytes: • ARPC Cryptogram(first 8 bytes) • Issuer Authentication Response Code (IARC)' Code (last 2 bytes)

Name	Description	Source	Format	Tag	Length	Values
Issuer Code Table Index	Indicates the code table according to ISO 8859 for displaying the Application Preferred Name	ICC	n 2	'9F11'	1	Values are: 01 = Part 1 of ISO 8859 02 = Part 2 of ISO 8859 03 = Part 3 of ISO 8859 04 = Part 4 of ISO 8859 05 = Part 5 of ISO 8859 06 = Part 6 of ISO 8859 07 = Part 7 of ISO 8859 08 = Part 8 of ISO 8859 09 = Part 9 of ISO 8859 10 = Part 10 of ISO 8859
Issuer Country Code	Indicates the country of the issuer, represented according to ISO 3166.	ICC	n 3	'5F28'	2	
Issuer Public Key Certificate	Issuer public key certified by a certification authority	ICC	b	'90'	N_{CA}	
Issuer Public Key Exponent	Issuer public key exponent used for the verification of the Signed Static Application Data and the ICC Public Key Certificate	ICC	b	'9F32'	1 or 3	
Issuer Public Key Remainder	Remaining digits of the Issuer Public Key Modulus	ICC	b	'92'	$N_I - N_{CA} + 36$	
Issuer Script Command	Contains a command for transmission to the card.	Issuer	b	'86'	var. up to 261	
Issuer Script Identifier	Identification of the Issuer Script	Issuer	b	'9F18'	4	

Name	Description	Source	Format	Tag	Length	Values
Issuer Script Template 1	Contains proprietary issuer data for transmission to the ICC before the second GENERATE AC command	Issuer	b	'71'	var.	
Issuer Script Template 2	Contains proprietary issuer data for transmission to the ICC after the second GENERATE AC command	Issuer	b	'72'	var.	
Language Preference	1-4 languages stored in order of preference, each represented by 2 alphabetical characters according to ISO 639	ICC	an 2	'5F2D'	2-8	
Last Online Application Transaction Counter (LATC)	ATC value of the last transaction that went online successfully.	ICC	b	'9F13'	2	Initial value is zero. Updated to contain the current value of the ATC when a transaction has been transmitted online and issuer authentication is successful.
Lower Consecutive Offline Limit	Issuer-specified data element indicating a preference for maximum number of consecutive offline transactions allowed for the application in a terminal with online capability.	ICC	b	'9F14'	1	
Merchant Category Code		Terminal	n 4	'9F15'	2	
Offline Enciphered Transaction Personal Identification Number	Data entered by the cardholder for his authentication, and enciphered by terminal	Terminal	b	'9F4D'	N_{IC} or N_{PE}	
PIN Try Counter	Number of tries remaining	ICC	b	'9F17'	1	Initial value is PIN Try Limit

Name	Description	Source	Format	Tag	Length	Values
Point of Service (POS) Entry Mode	Indicates source of cardholder account data at the terminal according to ISO 8583:1987	Terminal	n 2	'9F39'	1	
Processing Options Data Object List (PDOL)	Contains a list of terminal resident data objects (tags and lengths) needed by the ICC in processing the GET PROCESSING OPTIONS command	ICC	b	'9F38'	var.	For this version of specifications, the PDOL is not used.
Response Message Template Format 1	Contains the data objects (without tags and lengths) returned by the ICC in response to a command.	ICC	var.	80	-	
Response Message Template Format 2	Contains the data objects (with tags and lengths) returned by the ICC in response to a command	ICC	var.	77	-	
Service Code	Service code as defined on magnetic stripe tracks 1 and 2 according to ISO/IEC 7813.	ICC	n 3	'5F30'	2	
Short File Identifier (SFI)	Identifies the SFI to be used in the commands related to a given AEF	ICC	b	'88'	1	Values are: 1-10: Governed by joint payment systems 11-20: Payment system specific 21-30: Issuer specific
Signed Dynamic Application Data	Digital Signature on critical application parameters for dynamic data authentication	ICC	b	'9F4B'	N_{IC}	see 2.2.1
Signed Static Application Data	Digital Signature on critical application parameters for static data authentication	ICC	b	'93'	N_I	see 2.1.1

Name	Description	Source	Format	Tag	Length	Values
Static Data Authentication Tag List	List of tags of primitive data objects defined in this specification whose value fields are to be included in the Signed Static or Dynamic Application Data	ICC	b	'9F4A'	var.	For this specification, the SDA Tag list is limited to the AIP and the AID.
Terminal Capabilities	Indicates the card data input, CVM, and security capabilities of the terminal	Terminal	b	'9F33'	3	Byte 1 (Card Data Input Capability) bit 8: 1 = Manual Key Entry bit 7: 1 = Magnetic Stripe bit 6: 1 = IC with Contacts bit 5: 1 = RESERVED Byte 2 (CVM Capability) bit 8: 1 = Plaintext PIN for ICC verification bit 7: 1 = Enciphered PIN for online verification bit 6: 1 = Signature (Paper) bit 5: 1 = Enciphered PIN for offline verification bit 4: 1 = RESERVED Byte 3 (Security Capability) bit 8: 1 = Static Data Authentication bit 7: 1 = Dynamic Data Authentication bit 6: 1 = Card Capture bit 5: 1 = RESERVED

Name	Description	Source	Format	Tag	Length	Values
Terminal Country Code	Indicates the country of the terminal represented according to ISO 3166.	Terminal	n 3	'9F1A'	2	
Terminal Type	Indicates the environment of the terminal, its communication capability, and its operational control	Terminal	n 2	'9F35'	1	Values are: Financial Institution 11 = Attended Online only 12 = Attended Offline with online capability 13 = Attended Offline only 14 = Unattended Online only 15 = Unattended Offline with online capability 16 = Unattended Offline only Merchant 21 = Attended Online only 22 = Attended Offline with online capability 23 = Attended Offline only 24 = Unattended Online only 25 = Unattended Offline with online capability 26 = Unattended Offline only Cardholder 34 = Unattended Online only 35 = Unattended Offline with online capability 36 = Unattended Offline only
Terminal Verification Results	Status of the different functions as seen from the terminal	Terminal	b	'95'	5	Byte 1 bit 8: 1 = Offline Data authentication was not performed

Name	Description	Source	Format	Tag	Length	Values
						bit 7: 1 = Offline Static data authentication failed bit 6: 1 = ICC data missing bit 5: 1 = Card appears on terminal exception list bit 4: 1 = Offline Dynamic data authentication failed bit 3-1 = RESERVED <u>Byte 2</u> bit 8: 1 = ICC and terminal have different application versions bit 7: 1 = Expired application bit 6: 1 = Application not yet active bit 5: 1 = Requested service not allowed for card product bit 4: 1 = New card bit 3-1 = RESERVED <u>Byte 3</u> <u>b</u>it 8: 1 = Cardholder verification was not successful bit 7: 1 = Unrecognized CVM bit 6: 1 = PIN try limit exceeded bit 5: 1 = PIN entry required, PIN pad not present or not working bit 4: 1 = PIN entry required, PIN pad present, PIN not entered bit 3: 1 = Online PIN entered bit 2-1 = RESERVED

Name	Description	Source	Format	Tag	Length	Values
						Byte 4 bit 8: 1 = Transaction exceeds floor limit bit 7: 1 = Lower Consecutive Offline limit exceeded bit 6: 1 = Upper Consecutive Offline limit exceeded bit 5: 1 = Transaction selected randomly for online processing bit 4: 1 = Merchant forced the transaction online bit 3-1 = RESERVED Byte 5 bit 8: 1 = Default TDOL used bit 7: 1 = Issuer authentication was unsuccessful bit 6: 1 = Script processing failed before final GENERATE AC bit 5: 1 = Script processing failed after final GENERATE AC bit 4-1 = RESERVED
Track 1 Discretionary Data	Discretionary Data from track 1 of the magnetic stripe according to ISO/IEC 7813	ICC	ans	'9F1F'	var.	

Name	Description	Source	Format	Tag	Length	Values
Track 2 Equivalent Data	The Track 2 Equivalent Data contains the track 2 data elements as defined in ISO/IEC 7813 as follows: • PAN, Field Separator(hex 'D'), Expiration Date(YYMM), Service Code, Card Verification Code (CVC), and PIN Verification Value (PVV) (if present) • Issuer proprietary discretionary data (defined by individual payment systems) is optional; • Start sentinel, end sentinel, and longitudinal redundancy check (LRC) shall be excluded. The data included in the Track 2 Equivalent Data must be identical to the data on the Track 2 of the magnetic stripe except for the Issuer proprietary Discretionary Data. (CRF003.6	ICC	var. up to cn 37	'57'	var. up to 19	Pad with hex 'F' if needed to ensure whole bytes
Transaction Certificate (TC)	Application cryptogram computed by the card for an approved financial transaction.	ICC	b	'9F29'	8	
Transaction Certificate (TC) Hash Value	Results of a hash function	Terminal	b	'98'	8-20	
Transaction Certificate Data Object List (TDOL)	List of data objects (tag and length) to be used by the terminal in generating the TC Hash	ICC	b	'97'	var. up to 252	Not used in this specification

Name	Description	Source	Format	Tag	Length	Values
	Value					
Transaction Currency Code	Indicates the currency code of the transaction according to ISO 4217. The implied exponent is indicated by the minor unit of currency associated with the Transaction Currency Code in ISO 4217.	Terminal	n 3	'5F2A'	2	
Transaction Date	Local date that the transaction was authorized.	Terminal	n 6 YYMMDD	'9A'	3	
Transaction Personal Identification Number (PIN) Data	Data entered by the cardholder for the purpose of PIN verification	Terminal	cn	'99'	8	
Transaction Reference Currency Code	Code defining the common currency used by the terminal in case the Transaction Currency code is different from the Application Currency Code.	Terminal	n 3	'9F3C'	2	
Transaction Status Information	Indicates the functions performed in a transaction	Terminal	b	'9B'	2	Byte 1 bit 8: 1 = Offline Data authentication was performed bit 7: 1 = Cardholder verification was performed bit 6: 1 = Card risk management was performed bit 5: 1 = Issuer authentication was performed bit 4: 1 = Terminal risk management was performed bit 3: 1 = Script processing was performed bit 2-1 = RESERVED Byte 2

Name	Description	Source	Format	Tag	Length	Values
						bit 8-1 = RESERVED
Transaction Time	Local time that the transaction was authorized	Terminal	n 6 HHMMSS	'9F21'	3	
Transaction Type	Indicates the type of financial transaction, represented by the first two digits of ISO 8583:1987 Processing Code	Terminal	n 2	'9C'	1	
Unpredictable Number	Value to provide variability and uniqueness to the generation of the application cryptogram.	Terminal	b	'9F37'	4	
Upper Consecutive Offline Limit	Issuer-specified preference for the maximum number of consecutive offline transactions for this ICC application allowed in a terminal without online capability	ICC	b	'9F23'	1	

When the length defined for the data object is greater than the length of the actual data, the following rules apply:

- A data element in format n is right-justified and padded with leading hexadecimal zeroes
- A data element in format cn is left justified and padded with trailing 'F'
- A data element in format an is left-justified and padded with trailing hexadecimal zeroes
- A data element in format ans is left-justified and padded with trailing hexadecimal zeroes

3.1.3 MasterCard proprietary data elements

Table 3.3 gives the MasterCard proprietary data elements to support a MasterCard implementation.

TABLE 3.3: MASTERCARD PROPRIETARY DATA ELEMENTS

Name	Description	Source	Format	Length	Values
Application Control	Stores control of application (resetting of parameters, script control)	ICC	b	1	Could be used for indicating whether • the End Of Script is supported • resetting of internal parameters is allowed when Issuer Authentication Data is not present (RIP) • domestic requirements
Application Flag	Stores status of application in non-volatile memory	ICC	b		Complete with list of Volume I
Authorization Response Cryptogram (ARPC)	Value computed by the issuer host for online host authentication; first 8 bytes of the Issuer Authentication Data	Issuer	b	8	See 2.5
Card Issuer Action Code—Decline	EPI/MCI proprietary data element indicating the Issuer's conditions that cause the ICC to decline a transaction without attempting to go online.	ICC	b	3	Byte 1: bits 8-7: 00 bits 6-5: 00 bit 4: 0 bit 3: 0 bit 2: 1 = Offline PIN verification failed bit 1: 0 Byte 2: bit 8: 1 = Last online transaction not completed bit 7: 1 = PIN Try Limit exceeded bit 6: 1 = Exceeded velocity checking bit 5: 0 bit 4: 1 = Issuer authentication failure on last online transaction

Name	Description	Source	Format	Length	Values
					bit 3: 1 = Issuer authentication not performed after online authorization bit 2: 1 = Application blocked by card because PIN Try Limit exceeded bit 1: 1 = Static data authentication failed on last transaction and transaction declined offline Byte 3: bits 8-6: 000 bit 5: 1 = DDA failed on last offline transaction and transaction declined offline. bit 4: 1= Issuer Script processing failed on last transaction bit 3: 1 = Lower Consecutive Offline Limit or Lower Cumulative Offline Transaction Amount Exceeded bit 2: 1 = Upper Consecutive Offline Limit or Upper Cumulative Offline Transaction Amount Exceeded bit 1: 1 = Maximum Offline transaction amount exceeded
Card Issuer Action Code—Offline	EPI/MCI proprietary data element indicating the Issuer's conditions that cause the ICC to decline a transaction in an offline transaction terminal.	ICC	b	3	Byte 1: bits 8-7: 00 bits 6-5: 00 bit 4: 0 bit 3: 0 bit 2: 1 = Offline PIN verification failed bit 1: 1 = Unable to go online Byte 2: bit 8: 1 = Last online transaction not completed bit 7: 1 = PIN Try Limit exceeded bit 6: 1 = Exceeded velocity checking bit 5: 1 = New Card bit 4: 1 = Issuer authentication failure on last online transaction bit 3: 1 = Issuer authentication not performed after online authorization bit 2: 1 = Application blocked by card because PIN Try Limit exceeded bit 1: 1 = Static data authentication failed on last transaction and transaction declined offline

Name	Description	Source	Format	Length	Values
					Byte 3: bits 8-6: 000 bit 5: 1 = DDA failed on last offline transaction and transaction declined offline. bit 4: 1 = Issuer Script processing failed on last transaction bit 3: 1 = Lower Consecutive Offline Limit or Lower Cumulative Offline Transaction Amount Exceeded bit 2: 1 = Upper Consecutive Offline Limit or Upper Cumulative Offline Transaction Amount Exceeded bit 1: 1 = Maximum Offline transaction amount exceeded Byte 3: bit 3: 1 = Lower Consecutive Offline Limit or Lower Cumulative Offline Transaction Amount Exceeded bit 1: 1 = Maximum Offline transaction amount exceeded
Card Issuer Action Code—Online	EPI/MCI proprietary data element indicating the conditions that cause a transaction to be completed online according to the issuer and in an online capable transaction terminal.	ICC	b	3	Byte 1: bits 8-7: 00 bits 6-5: 00 bit 4: 0 bit 3: 0 bit 2:1 = Offline PIN verification failed bit 1: 0 Byte 2: bit 8: 1 = Last online transaction not completed bit 7: 1 = PIN Try Limit exceeded bit 6: 1 = Exceeded velocity checking bit 5: 1 = New Card bit 4: 1 = Issuer authentication failure on last online transaction bit 3: 0 bit 2: 1 = Application blocked by card because PIN Try Limit exceeded bit 1: 1 = Static data authentication failed on last transaction and transaction declined offline Byte 3:

Name	Description	Source	Format	Length	Values
					bits 8-6: 000 bit 5: 1 = DDA failed on last offline transaction and transaction declined offline. bit 4: 1 = Issuer cript processing failed on last transaction bit 3: 1 = Lower Consecutive Offline Limit or Lower Cumulative Offline Transaction Amount Exceeded bit 2: 1 = Upper Consecutive Offline Limit or Upper Cumulative Offline Transaction Amount Exceeded bit 1: 1 = Maximum Offline transaction amount exceeded
Card TVR Action Code	MCI/EPI proprietary data element indicating the Issuer's selected actions based upon the content of the Terminal Verification Results bits	ICC	b	5	Byte 1: bit 8: 1 = Data authentication was not performed bit 7: 1 = Static data authentication failed bit 6: 1 = ICC data missing bit 5: 1 = Card appears on terminal exception list bit 4: 1 = Dynamic data authentication failed bit 3-1 = RESERVED Byte 2: bit 8: 1 = ICC and terminal have different application versions bit 7: 1 = Expired application bit 6: 1 = Application not yet active bit 5: 1 = Requested service not allowed for card product bit 4: 1 = New card bit 3-1 = RESERVED Byte 3: bit 8: 1 = Cardholder verification was not successful bit 7: 1 = Unrecognized CVM bit 6: 1 = PIN try limit exceeded bit 5: 1 = PIN entry required, PIN pad not present or not working bit 4: 1 = PIN entry required, PIN pad present, PIN not entered bit 3: 1 = Online PIN entered bit 2-1 = RESERVED

Name	Description	Source	Format	Length	Values
					Byte 4: bit 8: 1 = Transaction exceeds floor limit bit 7: 1 = Lower Consecutive Offline limit exceeded bit 6: 1 = Upper Consecutive Offline limit exceeded bit 5: 1 = Transaction selected randomly for online processing bit 4: 1 = Merchant forced the transaction online bit 3-1 = RESERVED Byte 5: bit 8: 1 = Default TDOL used bit 7: 1 = Issuer authentication was unsuccessful bit 6: 1 = Script processing failed before final GENERATE AC bit 5: 1 = Script processing failed after final GENERATE AC bit 4-1 = RESERVED
Card Verification Results	Data element indicating the exception conditions that occurred during card risk management. Transmitted to the terminal in Issuer Application Data.	ICC	b	4	Byte 1: Length indicator ('03') Byte 2: bits 8-7: 00 = AAC returned in second GENERATE AC 01 = TC returned in second GENERATE AC 10 = Second GENERATE AC not requested 11 = RESERVED bits 6-5: 00 = AAC returned in first GENERATE AC 01 = TC returned in first GENERATE AC 10 = ARQC returned in first GENERATE 11 = RESERVED bit 4: 1 = Issuer authentication failed bit 3: 1 = Offline PIN verification performed bit 2: 1 = Offline PIN verification failed bit 1: 1 = Unable to go online

Name	Description	Source	Format	Length	Values
					Byte 3: bit 8: 1 = Last online transaction not completed bit 7: 1 = PIN Try Limit exceeded bit 6: 1 = Exceeded velocity checking bit 5: 1 = New Card bit 4: 1 = Issuer authentication failure on last online transaction bit 3: 1 = Issuer authentication not performed after online authorization bit 2: 1 = Application blocked by card because PIN Try Limit exceeded bit 1: 1 = Static data authentication failed on last transaction and transaction declined offline Byte 4: bits 8-6: Number of Issuer Script Commands containing secure messaging processed on last transaction bit 5: 1 = DDA failed on last online transaction and transaction declined offline. bit 4: Issuer Script processing failed on last transaction bit 3: 1 = Lower Consecutive Offline Limit or Lower Cumulative Offline Transaction Amount Exceeded bit 2: 1 = Upper Consecutive Offline Limit or Upper Cumulative Offline Transaction Amount Exceeded bit 1: 1 = Maximum Offline transaction amount exceeded Note: If only one GENERATE AC command is issued for a transaction, byte 2, bits 6-5 shall indicate that a TC or AAC is returned in the first GENERATE AC command and bits 8-7 shall be indicate that a second GENERATE AC command was not requested.

Name	Description	Source	Format	Length	Values
Certification Authority Public Key Modulus	Part of the RSA Certification Authority Public Key				
Cryptogram Version Number	Data element indicating the version of the TC, AAC or ARQC algorithm used by the application.	ICC	b	1	Transmitted in the Issuer Application Data.
Cumulative Offline Transaction	MCI/EPI proprietary data element indicating the ICC internal counter accumulating the consecutive offline approved payment transaction amounts.	ICC	n	6	
Hash Algorithm Indicator	Algorithm used to compress data prior to signing.	ICC		1	Value is '01' for SHA-1
ICC PIN Encipherment Private Key	ICC private key to verify the enciphered offline PIN			$80 = N_{PE} < N_I$	
ICC PIN Encipherment Public Key Modulus	Part of the RSA ICC Public Key used for offline PIN encipherment			$80 = N_{PE} < N_I$	
ICC Private Key	ICC private key used for creating the Signed Dynamic Application Data (or verifying the enciphered offline PIN)			$80 = N_{IC} < N_I$	The memory occupied is implementation dependent. For an implementation based on the Chinese Remainder theorem, the memory required is 5Nic/2. For an implementation based on standard exponentiation the memory required is 2Nic.
ICC Public Key Modulus	Part of the RSA public key used for DDA			$80 = N_{IC} < N_I$	Can also be used for enciphered offline PIN verification is no dedicated key pair is present for that purpose.

Name	Description	Source	Format	Length	Values
Integrated Circuit Card (ICC) Dynamic Data	Part of the digital signature generated by the ICC with the ICC private key on critical application parameters	ICC	var.	$L_{DD <}$ Nic-25	For this specification the ICC Dynamic Data is limited to the concatenation of hex '08' and an ICC Dynamic Number
Issuer Authentication Response Code	Response Code in Issuer Authentication Data used to validate ARPC	Issuer	an 2	2	Values '00', '01' and '08' are considered as approval.
Issuer Public Key Modulus	Part of the RSA Issuer Public Key			$80 = N_I < N_{CA}$	
Issuer Script Command Counter	This Script Counter is incremented after each script processing command performed successfully.	ICC	b	1	It is reset during the second Generate AC after a successful Issuer Authentication if no critical script ('71') has been performed during the current transaction.
Key Derivation Index	Index of the key used in Application Cryptogram	ICC	b	1	Value assigned by the Issuer/payment system
Lower Cumulative Domestic Offline Transaction Amount	MCI/EPI proprietary data element indicating the lower maximum value which the 'Offline Cumulative Transaction Amount' can reach. Once this value is exceeded, the payment transaction will seek online authorization.	ICC	n	6	

Name	Description	Source	Format	Length	Values
Maximum Domestic Offline Transaction Amount	MCI/EPI proprietary data element indicating the maximum transaction amount. Once this value is exceeded, the payment transaction will seek online authorization.	ICC	n	6	
MK_{AC}	ICC Master key for Cryptogram calculation and Issuer Authentication	ICC	b	16	The Issuer Master-keys IMK from which the ICC Master keys MK are derived, are not stored in the card. (CRF004.9)
MK_{IDN}	ICC Master key for the Generation of the ICC Dynamic Number	ICC	b	16	
MK_{SMC}	ICC Master key for Secure messaging for Confidentiality	ICC	b	16	
MK_{SMI}	ICC Master key for Secure messaging for Integrity	ICC	b	16	
Non Domestic Control Factor	MCI/EPI proprietary data element. A n power of 2 used to divide the maximum consecutive non domestic transactions	ICC	b	1	The Non Domestic Control Factor can be implemented easily as NDCF = 2^n
PIN Try Limit	MCI/EPI proprietary data element indicating the maximum number of consecutive wrong PIN entries.	ICC	b	1	

Name	Description	Source	Format	Length	Values
Reference Currency Conversion Table	MCI/EPI proprietary data element which contains reference currency codes, conversion rates and the exponents for currency conversion	ICC	b	up to 20	Byte 1-2: Reference Currency Code Byte 3-4: Reference Currency to Application Currency Conversion rate Byte 5: Reference Currency to Application Currency Conversion rate exponent
Reference Personal Identification Number (PIN)	MCI/EPI proprietary data element indicating the PIN which is stored in the ICC, and against which the transaction PIN is verified.	ICC	n	var.	The PIN is 4 to12 digits long.
Upper Cumulative Domestic Offline Transaction Amount	MCI/EPI proprietary data element indicating the upper maximum value which the 'Offline Cumulative Transaction Amount' can reach. Once this value is exceeded, the payment transaction will seek online authorization or be declined.	ICC	n	6	

3.2 UPDATING CARD RISK MANAGEMENT DATA

Issuers may define any card risk management function within their ICC that they choose and implement it in cooperation with their chosen card vendor(s).

The data elements which set the risk management boundaries are provided in Table 3.4. The data elements other than the ATC, and last online ATC, may be updated at the option of the issuer during the life of the ICC by using a script processing command.

The ATC and last online ATC can only be updated by the ICC internally.

TABLE 3.4: CARD RISK MANAGEMENT DATA (PSMXXX)

Value	Tag
ATC	'9F36'
Last Online ATC	'9F13'
CIAC – Decline	—
CIAC – Offline	—
CIAC – Online	—
Card TVR Action Code	—
Lower Consecutive Offline Limit	'9F14'
Upper Consecutive Offline Limit	'9F23'
Maximum Domestic Offline Transaction Amount	—
Lower Cumulative Domestic Offline Amount	—
Upper Cumulative Domestic Offline Amount	—
Non Domestic Control Factor	—
Reference Currency Conversion Table	—

3.3 CARD RISK MANAGEMENT DATA OBJECT LIST

The ICC's CDOL1 and CDOL2 contain the tags and lengths for the concatenated data objects to be transmitted from the terminal to the ICC in the GENERATE AC command. Part of the card risk management is based on these data elements and a part of these data will be included in the Application Cryptogram (see Section 2.4).

Issuers can easily check that a terminal has performed the Card Authentication process by requesting from the terminal, in CDOL1 or CDOL2, data retrieved from the terminal when performing card authentication, and inserting this data in the Issuer Application Data. For Static Data Authentication this data is the Data Authentication Code (DAC). For Dynamic Data Authentication this data is the (first two bytes of the) ICC Dynamic Number.

3.3.1 Card Risk Management Data Object List 1

A recommended list of data elements that may be used in the CDOL1 is as shown in Table 3.5.

TABLE 3.5: RECOMMENDED CDOL1 DATA OBJECT

Value	Tag
Amount, Authorized	'9F02'
Amount, Other	'9F03'
Terminal Country Code	'9F1A'
Terminal Verification Results	'95'
Transaction Currency Code	'5F2A'
Transaction Date	'9A'
Transaction Type	'9C'
Unpredictable Number	'9F37'
Terminal Type	'9F35'
Amount in Reference Currency	'9F3A'
Transaction Reference Currency Code	'9F3C'
ICC Dynamic Number	'9F4C'
Data Authentication Code	'9F45'

3.3.2 Card Risk Management Data Object List 2

A recommended list of data elements in CDOL2 is indicated in Table 3.6:

TABLE 3.6: RECOMMENDED CDOL2 DATA OBJECTS

Value	Tag
Issuer Authentication Data	'91'
Authorization Response Code	'8A'
Terminal Verification Results	'95'

- As indicated before, only Authorization Response Code and TVR are mandatory since other data elements could be retained from CDOL1.

- The Issuer Authentication Data (tag "91") is only indicated in CDOL2 if Issuer Authentication is done in the 2nd GENERATE AC and not by means of an EXTERNAL AUTHENTICATE command.

- The Authorization Response Code is not used in cryptogram generation but is used by the ICC in completion processing. In this process the ICC verifies the Authorization Code to ascertain whether the transaction succeeded in going online or not.

If the other data elements are not retained from the CDOL1, they can be added to CDOL2.

A list of data elements that can be added to the CDOL2 is as shown in Table 3.5.

TABLE 3.5: ADDITIONAL CDOL2 DATA OBJECTS

Value	Tag
Amount, Authorized	'9F02'
Amount, Other	'9F03'
Terminal Country Code	'9F1A'
Transaction Currency Code	'5F2A'
Transaction Date	'9A'
Transaction Type	'9C'
Unpredictable Number	'9F37'
ICC Dynamic Number	'9F35'
Data Authentication Code	'9F3A'

3.4 CARD LIFE CYCLE DATA

Card Life Cycle Data (CLCD) can be used by the Issuer for management of the ICC when the specification is implemented. The format and content of the Card Life Cycle Data is under the control of the Application Provider, and the Card Manufacturer. If the Application Provider wants to delegate part of the ICC management to the Payment System, the Payment System will be involved also.

APPENDIX A NETWORK DATA ELEMENT REQUIREMENTS

NETWORK DATA ELEMENT REQUIREMENTS

EMV '96 Integrated Circuit Card Terminal Specification for Payment Systems, Version 3.1.1 describes message content requirements associated with ICC transactions and chip data elements.

This annex summarizes by message the list of new data elements that the Acquirer is to pass to the Issuer. Some of these data elements may not be provided/requested by the ICC or the Issuer, however the acquirer should support them and pass them when present. Data element 55 has been established for these chip data in the Authorization Messages.

Table A.1 lists the data elements that should be supported by acquirer for authorization request.

Table A.2 lists the data elements that should be supported by acquirer for authorization response.

Table A.3 lists the data elements that should be supported by acquirer in clearing message.

TABLE A.1: CHIP DATA ELEMENTS IN AUTHORIZATION REQUEST MESSAGE

Field #	Name	Tag	Format	Length	Usage
1	Application Cryptogram (ARQC)	'9F26'	b	8	Allows Card Authentication processing
2	Cryptogram Information Data	'9F27'	b	1	Indicates the type of cryptogram returned (approved, online or denied) by the card and the actions to be performed by the terminal (Advice or not, reason for advice)
3	Issuer Application Data (IAD)	'9F10'	b	Var. up 32	Provides data elements the Issuer has elected to have in the authorization message, for example derivation key and algorithm version numbers. Present if provided by ICC in GENERATE AC command response.
4	Unpredictable Number	'9F37'	b	4	Provides variability and uniqueness of the ARQC in order to reduce fraud risk. Present if input to application cryptogram calculation.
5	Application transaction counter (ATC)	'9F36'	b	2	Indicates sequential order of transaction performed by the application. Provides variability and uniqueness of ARQC in order to reduce fraud risk
6	Terminal Verification Result	'95'	b	5	Informs Issuer of the reason(s) the transaction went online
7	Transaction Date	'9A'	n	3	Indicates the date the transaction was conducted and the authorization requested
8	Transaction Type	'9C'	n	1	Indicates the transaction type used for application usage control
9	Amount Authorized	'9F02'	n	6	Indicates the amount sent to the Issuer for authorization
10	Transaction Currency Code	'5F2A'	n	2	Indicates the currency code of the merchant/terminal associated with the transaction
11	Application Interchange Profile	'82'	b	2	Indicates the specified chip application capabilities

Field #	Name	Tag	Format	Length	Usage
12	Terminal Country Code	'9F1A'	n	2	Indicates the country code of the merchant/terminal associated with the transaction
13	Amount Other	'9F03'	n	6	Indicates the cashback amount sent to the Issuer for authorization. Present if cashback used for current transaction.

It is assumed that Track2 Equivalent Data (PAN, PAN Sequence Number and expiration date) are already present in Authorization Message.

Table A.2: Chip Data Elements in Authorization Request Response Message

Field #	Name	Tag	Format	Length	Usage
1	Issuer Authentication Data	'91'	b	Var 8 to 16	Provide data to be transmitted to the card for Issuer authentication
2	• Issuer Script Template 1 • Issuer Script Template 2	'71' '72'	b	Var up 128	Allows the Issuer to provide a command for transmission to the card. Present if commands to ICC are sent by issuer. Acquirer network shall support a minimum of 24 Bytes

TABLE A.3: CHIP DATA ELEMENTS IN CLEARING MESSAGE

Field #	Name	Tag	Format	Length	Usage
1	Application Cryptogram (TC/ARQC or AAC)	'9F26'	b	8	Allows authentication of transaction data in support of chargeback processing, for single message terminal ARQC may be used as TC substitute
2	Cryptogram Information Data	'9F27'	b	1	Indicates the type of cryptogram returned (approved, online or denied) by the card and the actions to be performed by the terminal (Advice or not, reason for advice)
3	Issuer Application Data (IAD)	'9F10'	b	Var. up 32	Provides data elements the Issuer has elected to have in the authorization message such as derivation key and algorithm version numbers. Present if provided by ICC in GENERATE AC command response.
4	Unpredictable Number	'9F37'	b	4	Provides variability and uniqueness of the application cryptogram in order to reduce fraud risk. Present if input to application cryptogram calculation.
5	Application transaction counter (ATC)	'9F36'	b	2	Indicates sequential order of transaction performed by the application. Provides variability and uniqueness of the application cryptogram in order to reduce fraud risk
6	Terminal Verification Result	'95'	b	5	Informs Issuer of the results of transaction processing performed by the terminal and may be used for dispute management
7	Transaction Date	'9A'	n	3	Indicates the date the transaction was conducted
8	Transaction Type	'9C'	n	1	Indicates the type of transaction conducted by the cardholder
9	Amount Authorized	'9F02'	n	6	Indicates the amount authorized for the transaction and used to generate the application cryptogram). This amount may be different from the transaction amount
10	Transaction Currency Code	'5F2A'	n	2	Indicates the currency code associated with the transaction
11	Application Interchange Profile	'82'	b	2	Indicates the specified chip application capabilities

Field #	Name	Tag	Format	Length	Usage
12	Terminal Country Code	'9F1A'	n	2	Indicates the country code of the merchant/terminal associated with the transaction. Provides liability protection
13	Amount Other	'9F03'	n	6	Indicates the cashback amount sent to the Issuer for authorization. Present if cashback used for current transaction

It is assumed that Track2 Equivalent Data (PAN, PAN Sequence Number and expiration date) are already present in Clearing Message.

APPENDIX B DIGITAL SIGNATURE SCHEME GIVING MESSAGE RECOVERY

B.1 OVERVIEW

This section describes the digital signature scheme giving message recovery using a hash function according to ISO/IEC CD 9796-2. The main features of the scheme are the following.

- Adding of redundancy in the signature by applying a hash function to the data to be signed
- Adding of a header and trailer byte to obtain a unique recovery procedure and to prevent certain attacks

The scheme uses the following two algorithms:

- The reversible asymmetric algorithm RSA/Rabin as specified in Appendix C, consisting of a signing function $Sign(S_K)[\ ]$ depending on a Private Key S_K and a recovery function $Recover(P_K)[\ ]$ depending on a Public Key P_K. Both functions map N-byte numbers onto N-byte numbers and have the property that

$$Recover(P_K)[Sign(S_K)[X]] = X,$$

for any N-byte number X.

- The secure hash function SHA-1 as specified in Appendix C, that maps a message M of arbitrary length onto an 20-byte hash code H:

$$H: = SHA(M)$$

In the following two sections the signature generation and verification process is described.

B.2 SIGNATURE GENERATION

The computation of a signature S on a message M, consisting of an arbitrary number L of bytes, takes place in the following way:

CASE 1*: The length L of the message to be signed is at most N - 22 bytes*

1. Compute the 20-byte hash value H: = SHA[M] of the message M

2. Define the (N – 21 – L)-byte block B as follows:

$$B: = \text{‘4A’ if } L = N - 22,$$

$$B: = (\text{‘4B’} \,||\, \text{‘BB’} \,||\, \text{‘BB’} \,||\, \ldots \,||\, \text{‘BB’} \,||\, \text{‘BA’}) \text{ if } L < N - 22$$

3. Define the byte E: = ‘BC’.

4. Define the N-byte block X as the concatenation of the blocks B, M, H and E, hence

$$X: = (B \,||\, M \,||\, H \,||\, E)$$

5. The digital signature is then defined as the N-byte number

$$S: = \text{Sign}(S_K)[X]$$

CASE 2: *The length L of the message to be signed is larger than N - 22 bytes*

1. Compute the 20-byte hash value H: = SHA[M] of the message M.

2. Split M into two parts $M = (M_1 \,||\, M_2)$, where M_1 consists of the N – 22 leftmost (most significant bytes) of M and M_2 of the remaining (least significant) L – N + 22 bytes of M.

3. Define the byte B: = ‘6A’.

4. Define the byte E: = ‘BC’.

5. Define the N-byte block X as the concatenation of the blocks B, M_1, H and E, hence

$$X: = (B \,||\, M_1 \,||\, H \,||\, E)$$

6. The digital signature S is then defined as the N-byte number

$$S: = \text{Sign}(S_K)[X]$$

B.3 SIGNATURE VERIFICATION

Signature verification takes place in the following way:

CASE 1: *The length L of the message signed is at most N – 22 bytes*

1. Check whether the digital signature S consists of N bytes.

2. Retrieve the N-byte number X from the digital signature S:

$$X = \text{Recover}(P_K)[S].$$

3. Partition X as X = (B || M || H || E), where
   - B = '4A' or B is a leading byte-string of the form ('4B' || 'BB' || 'BB' || . . . || 'BB' || 'BA'). If none of these cases occur, the signature is rejected
   - H is 20 bytes long
   - E is one byte long
   - M consists of the remaining bytes

4. Check whether the byte E is equal to 'BC'

5. Check whether H = SHA[M]

If and only if these checks are correct is the message accepted as genuine.

CASE 2: *The length L of the message signed is larger than N – 22 bytes*

1. Check whether the digital signature S consists of N bytes.

2. Retrieve the N-byte number X from the digital signature S:

$$X = \text{Recover}(P_K)[S].$$

3. Partition X as X= (B || M_1 || H || E), where
   - B is one byte long
   - H is 20 bytes long
   - E is one byte long
   - M_1 consists of the remaining N – 22 bytes

4. Check whether the byte B is equal to '6A'

5. Check whether the byte E is equal to ‘BC’

6. Compute $M = (M_1 \| M_2)$ and check whether H = SHA[M].

If and only if these checks are correct is the message accepted as genuine.

APPENDIX C CRYPTOGRAPHIC ALGORITHMS

C.1 DES AND TRIPLE-DES

The *Data Encryption Standard* (DES) standardized in ISO 8731-1 and ISO 8372 is the block cipher used for Application Cryptogram generation, Issuer Authentication and Secure Messaging.

More precisely, both single-DES and its triple-DES (DES3) version described below are used in the session key derivation, MAC and encryption mechanisms described in Section 2.

The DES algorithm takes as input an 8-byte plain text block X and an 8-byte secret key K and produces an 8-byte ciphertext block Y:

$$Y := DES(K)[X]$$

Decryption is denoted as

$$X := DES^{-1}(K)[Y]$$

Triple DES encryption involves encrypting an 8-byte plain text block X in an 8-byte ciphertext block Y with a double-length (16-byte) secret key $K = (K_L \,||\, K_R)$ as follows:

$$Y := DES3(K)[X] := DES(K_L)[DES^{-1}(K_R)[DES(K_L)[X]]]$$

Decryption takes place as follows:

$$X := DES3^{-1}(K)[Y] := DES^{-1}(K_L)[DES(K_R)[DES^{-1}(K_L)[Y]]]$$

C.2 RSA/RABIN

The RSA algorithm and its Rabin variant with exponent 2 is the asymmetric reversible cryptographic algorithm used in the digital signature scheme described in Annex B for Static and Dynamic Data Authentication.

The RSA/Rabin algorithm produces a digital signature whose length equals the size of the modulus used. The lengths in bytes of the moduli and the values of the public exponents involved in Static Data Authentication, Dynamic Data Authentication and PIN Encipherment are specified in Table C.1 and Table C.2. The bit length of all moduli shall be a multiple of 8, the leftmost bit of its leftmost byte being 1.

TABLE C.1: LENGTHS IN BYTES OF PUBLIC KEY MODULI

Description	Length
Certification Authority Public Key Modulus	$96 \leq N_{CA} \leq 128$
Issuer Public Key Modulus	$80 \leq N_I < N_{CA}$
ICC Public Key Modulus	$80 \leq N_{IC} < N_I$
ICC PIN Encipherment Public Key Modulus	$80 \leq N_{PE} < N_I$

TABLE C.2: VALUES OF PUBLIC KEY EXPONENTS

Description	Value
Certification Authority Public Key Exponent	2 or 3
Issuer Public Key Exponent	2 or 3 or $2^{16}+1$
ICC Public Key Exponent	2 or 3 or $2^{16}+1$
ICC PIN Encipherment Public Key Exponent	3 or $2^{16}+1$

The Public Key Algorithm Indicator for the RSA/Rabin algorithm shall be coded as hexadecimal '01'.

The keys and signing and recovery functions for the RSA/Rabin algorithm are specified below. The cases for odd public key exponent and public key exponent equal to 2 are considered separately.

C.2.1 Odd Public Key Exponent

C.2.1.1 Keys

The private key S_K of the RSA digital signature scheme with an odd public key exponent e consists of two prime numbers p and q such that p – 1 and q – 1 are co-prime to e and a private exponent d such that

$$ed \equiv 1 \bmod (p - 1)(q - 1).$$

The corresponding public key P_K consists of the public key modulus n = pq and the public key exponent e.

C.2.1.2 Signing Function

The signing function for RSA with an odd public key exponent is defined as

$$S = Sign(S_K)[X]: = X^d \bmod n,\ 0 < X < n,$$

where X is the data to be signed and S the corresponding digital signature.

C.2.1.3 Recovery Function

The recovery function for RSA with an odd public key exponent is equal to

$$X = Recover(P_K)[S]: = S^e \bmod n.$$

C.2.2 Public Key Exponent 2

C.2.2.1 Keys

The private key S_K of the Rabin variant of the RSA digital signature scheme with public key exponent 2 consists of two prime numbers p and q such that $p \equiv 3 \bmod 8$ and $q \equiv 7 \bmod 8$ and a private exponent d equal to

$$d = (pq - p - q + 5)/8.$$

The corresponding public key P_K consists of the public key modulus n = pq and the public key exponent 2.

C.2.2.2 Signing Function

The signing function for Rabin is defined as

$$S = Sign(S_K)[X]: = X^d \bmod n, \text{ if the Jacobi symbol } (X \mid n) = 1,$$

$$S = Sign(S_K)[X]: = (X/2)^d \bmod n, \text{ if the Jacobi symbol } (X \mid n) = -1,$$

where X is the data to be signed and S the corresponding digital signature.

The data block X is a nonnegative integer less than n with the additional property that its least significant byte is equal to 'BC'. This is to ensure that the recovery process is unique. Note that this is always the case when RSA is used in the digital signature scheme described in Annex B.

C.2.2.3 Recovery Function

The recovery function for Rabin is as follows.

1. Compute $Y: = S^e \bmod n$

2. The recovered data is obtained by one of the following cases
   a. If the least significant byte of Y is equal to 'BC', X: = Y
   b. If the least significant byte of 2Y is equal 'BC', X: = 2Y
   c. If the least significant byte of n – Y is equal to 'BC', X: = n – Y
   d. If the least significant byte of 2(n – Y) is equal to 'BC', X: = 2(n – Y)

If none of these cases occur, the signature shall be rejected

C.3 SHA-1

The *Secure Hash Algorithm* (SHA-1) standardized in ISO/IEC 10118-3 and FIPS 180-1 is used for the generation of the TC Hash Value and in the digital signature scheme described in Annex B for Static and Dynamic Data Authentication. SHA-1 takes as input messages M of arbitrary length and produces a 20-byte hash value H:

$$H: = SHA(M).$$

The Hash Algorithm Indicator for this hashing algorithm shall be coded as hexadecimal '01'.

INDEX

The page references in this index reflect the pages where significant descriptions or procedures begin. The page numbers separated by a dash (-) refer to a section or appendix (before the dash) and a page number within the section or appendix (after the dash). For example: 3-2 refers to section 3, page 2; A-4 refers to page 4 of appendix A Page numbers in **bold** type refer to figures or illustrated text.

The following conventions are used:

- Entries are listed alphabetically, in word order
- Entries that are abbreviations have the full name following in italics
- *See* references are supplied from non-preferred to preferred terms
- *See also* references are supplied between related terms

NUMERICS

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W

Y

Z