EMV®
Integrated Circuit Card Specifications for Payment Systems

Book 3

Application Specification

Version 4.4
October 2022

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

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

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

Revision Log – Version 4.4

The following changes have been made to Book 3 since the publication of Version 4.3. Numbering and cross references in this version have been updated to reflect changes introduced by the published bulletins.

Incorporated changes described in the following Specification Bulletins:

- Specification Bulletin no. 113: SDA selected TVR bit
- Specification Bulletin no. 117, Second Edition: Order of Data Elements in Templates
- Specification Bulletin no. 118: Clarification on the Definition of Payment System
- Specification Bulletin no. 120: Reserve AIP Bits for Contactless
- Specification Bulletin no. 140: Clarification on the Format of AFL, Byte 1
- Specification Bulletin no. 141: Clarification on Random Transaction Selection
- Specification Bulletin no. 147: Clarification on the Format of Exponent Data Elements
- Specification Bulletin no. 159: Incorrectly Formatted ICC Data Objects
- Specification Bulletin no. 161: Reserve TVR and AIP Bits for Contactless
- Specification Bulletin no. 163: Changes to PIN Pad requirements
- Specification Bulletin no. 164: Electronic Signature Capture and Electronic Receipt Delivery
- Specification Bulletin no. 175: Application Selection Registered Proprietary Data
- Specification Bulletin no. 178, Third Edition: Tokenisation Data Objects – Payment Account Reference (PAR)
- Specification Bulletin no. 185, Second Edition: Biometric Terminal Specification
- Specification Bulletin no. 197: Tokenisation Data Objects – Token Requestor ID and Last 4 Digits of PAN
- Specification Bulletin no. 209: Clarifications for EMV® Book 3 (Version 4.3), offline-only terminals processing
- Specification Bulletin no. 231: Issuer Identification Number Extended (IINE)
- Specification Bulletin no. 234: Data Object Lists (DOLs) and Constructed Data Objects
- Specification Bulletin no. 243: Introduction of XDA/ODE for EMV Specifications
- Specification Bulletin no. 260: Terminal Risk Management Clarification

Specification Bulletin no. 267: Clarification on Handling of Unknown ICC Data and Track 1 Discretionary Data

Specification Bulletin no. 273: Clarification on Setting TSI bit for CDA (Mode 1 and 2) Transaction

Minor editorial clarifications and corrections, including those described in the following:

Specification Bulletin no. 175: Application Selection Registered Proprietary Data

Specification Bulletin no. 243: Introduction of XDA/ODE for EMV Specifications

Contents

Part I – General

Part II – Data Elements and Commands

Tables

Figures

Part I

General

1 Scope

This document, the *Integrated Circuit Card Specifications for Payment Systems – Book 3, Application Specification* defines the terminal and integrated circuit card (ICC) procedures necessary to effect a payment system transaction in an international interchange environment.

The *Integrated Circuit Card Specifications for Payment Systems* includes the following additional documents, all available on http://www.emvco.com:

- Book 1 – Application Independent ICC to Terminal Interface Requirements
- Book 2 – Security and Key Management
- Book 4 – Cardholder, Attendant, and Acquirer Interface Requirements

1.1 Changes in Version 4.4

This release incorporates all relevant Specification Bulletins, Application Notes, amendments, etc. published up to the date of this release.

The Revision Log at the beginning of the Book provides additional detail about changes to this specification.

1.2 Structure

Book 3 consists of the following parts:

Part I	–	**General**
Part II	–	**Data Elements and Commands**
Part III	–	**Debit and Credit Application Specification**
Part IV	–	**Annexes**
Part V	–	**Common Core Definitions**

Part I includes this introduction, as well as data applicable to all Books: normative references, definitions, abbreviations, notations, data element format convention, and terminology.

Part II describes data elements and files as well as commands for financial transaction.

Part III specifies the debit and credit application functions including:

- Transaction flow (the sequence of events and the commands issued to the card)
- Exception processing
- Definition of data elements and commands as they apply to the exchange of information between an ICC and a terminal. In particular,
  - Structure and referencing of files
  - The usage of commands between the ICC and the terminal to achieve application level functions.

The functions described are those necessary to ensure that payment system cards conforming to this specification can perform a set of core functions in terminals conforming to this specification. Application functions unique to individual payment systems and those functions not performed in interchange are not described, but are not precluded.

Part IV includes a complete data elements dictionary, rules for BER-TLV data objects, instructions for coding of specific data elements, and transaction log information. It discusses TVR and TSI bit settings following script processing, and Status Words returned in response to an EXTERNAL AUTHENTICATE command.

Part V defines an optional extension to be used when implementing a card complying to the Common Core Definitions (CCD).

The Book also includes a revision log and an index.

This specification does not address clearing and settlement or any transactions where the ICC is not present.

1.3 Underlying Standards

This specification is based on the ISO/IEC 7816 series of standards and should be read in conjunction with those standards. However, if any of the provisions or definitions in this specification differ from those standards, the provisions herein shall take precedence.

1.4 Audience

This specification is intended for use by manufacturers of ICCs and terminals, system designers in payment systems, and financial institution staff responsible for implementing financial applications in ICCs.

2 Normative References

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

EMV Contact Interface Specification	EMV Level 1 Specifications for Payment Systems, EMV Contact Interface Specification
EMV Tokenisation Framework	EMV Payment Tokenisation Specification – Technical Framework Framework specification for an interoperable Payment Tokenisation solution.
FIPS 202	SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions
IEEE P1363	Standard Specifications For Public-Key Cryptography
ISO 639-1	Codes for the representation of names of languages – Part 1: Alpha-2 Code **Note:** This standard is updated continuously by ISO. Additions/changes to ISO 639-1:1988: Codes for the Representation of Names of Languages are available on: http://www.loc.gov/standards/iso639-2/php/code_changes.php
ISO 3166	Codes for the representation of names of countries and their subdivisions
ISO 4217	Codes for the representation of currencies and funds
ISO/IEC 7812-1	Identification cards – Identification of issuers — Part 1: Numbering System
ISO/IEC 7813	Identification cards – Financial transaction cards
ISO/IEC 7816-4	Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange
ISO/IEC 7816-5	Identification cards — Integrated circuit cards — Part 5: Registration of application providers
ISO/IEC 7816-6	Identification cards – Integrated circuit cards – Part 6: Interindustry data elements for interchange

ISO/IEC 7816-11	Identification cards – Integrated circuit cards – Personal verification through biometric methods
ISO 8583:1987	Bank card originated messages – Interchange message specifications – Content for financial transactions
ISO 8583:1993	Financial transaction card originated messages – Interchange message specifications
ISO/IEC 8825-1	Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)
ISO/IEC 8859	Information processing – 8-bit single-byte coded graphic character sets
ISO 9362	Banking – Banking telecommunication messages – Bank identifier codes
ISO 9564-1	Financial services – Personal Identification Number (PIN) management and security – Part 1: Basic principles and requirements for PINs in card-based systems
ISO/IEC 9796-2	Information technology – Security techniques – Digital signature schemes giving message recovery – Part 2: Integer factorization based mechanisms
ISO/IEC 9797-1	Information technology – Security techniques – Message Authentication Codes – Part 1: Mechanisms using a block cipher
ISO/IEC 9797-2	Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a dedicated hash-function
ISO/IEC 10116	Information technology – Security techniques – Modes of operation for an n-bit block cipher
ISO/IEC 10118-3	Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions
ISO/IEC 11770-6	Information technology – Security techniques – Key management — Part 6: Key derivation
ISO 13616	Banking and related financial services – International bank account number (IBAN)

ISO/IEC 14888-3	Information technology – Security techniques – Digital signatures with appendix — Part 3: Discrete logarithm based mechanisms
ISO/IEC 15946-1	Information technology – Security techniques – Cryptographic techniques based on elliptic curves — Part 1: General
ISO/IEC 15946-5	Information technology – Security techniques – Cryptographic techniques based on elliptic curves — Part 5: Elliptic curve generation
ISO 16609	Banking – Requirements for message authentication using symmetric techniques
ISO/IEC 18031	Information technology – Security techniques – Random bit generation
ISO/IEC 18033-2	Information technology – Security techniques – Encryption algorithms – Part 2: Asymmetric ciphers
ISO/IEC 18033-3	Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers
ISO/IEC 19772	Information technology – Security techniques – Authenticated encryption
ISO/IEC 19785-3	Information technology – Common Biometric Exchange Formats Framework – Patron format specifications
ISO/IEC 19794	Information technology – Biometric data interchange formats
ISO/IEC 19794-2	Information technology – Biometric data interchange formats – Part 2: Finger minutiae data
SEC 1	Elliptic Curve Cryptography (available at http://www.secg.org)

3 Definitions

The following terms are used in one or more books of these specifications.

Application	The application protocol between the card and the terminal and its related set of data.
Application Authentication Cryptogram	An Application Cryptogram generated by the card when declining a transaction.
Application Cryptogram	A cryptogram generated by the card in response to a GENERATE AC command. See also: • Application Authentication Cryptogram • Authorisation Request Cryptogram • Transaction Certificate
Authentication	The provision of assurance of the claimed identity of an entity or of data origin.
Authorisation Request Cryptogram	An Application Cryptogram generated by the card when requesting online authorisation.
Authorisation Response Cryptogram	A cryptogram generated by the issuer in response to an Authorisation Request Cryptogram.
Biometric Data Block	A block of data with a specific format that contains information captured from a biometric capture device and that could be used as follows: • stored in the card as part of the biometric reference template • sent to the ICC in the data field of the PIN CHANGE/UNBLOCK command • sent to the ICC in the data field of the VERIFY command for offline biometric verification • sent online for verification The format of the BDB is outside the scope of this specification.

Biometric Reference Template	Biometric data stored in the card as reference. Data provided by a biometric capture device would be compared against the biometric reference template to determine a match.
Biometric Verification	The process of determining that the biometrics presented, such as finger, palm, iris, voice, or facial, are valid.
Byte	8 bits.
Card	A payment card as defined by a payment system.
Certificate	The public key and identity of an entity together with some other information, rendered unforgeable by signing with the private key of the certification authority which issued that certificate.
Certification Authority	Trusted third party that establishes a proof that links a public key and other relevant information to its owner.
Ciphertext	Enciphered information.
Combined DDA/Application Cryptogram Generation	A form of offline dynamic data authentication.
Command	A message sent by the terminal to the ICC that initiates an action and solicits a response from the ICC.
Command Chaining	A mechanism where consecutive command-response pairs can be chained.
Compromise	The breaching of secrecy or security.
Concatenation	Two elements are concatenated by appending the bytes from the second element to the end of the first. Bytes from each element are represented in the resulting string in the same sequence in which they were presented to the terminal by the ICC, that is, most significant byte first. Within each byte bits are ordered from most significant bit to least significant. A list of elements or objects may be concatenated by concatenating the first pair to form a new element, using that as the first element to concatenate with the next in the list, and so on.

Contact	A conducting element ensuring galvanic continuity between integrated circuit(s) and external interfacing equipment.
Cryptogram	Result of a cryptographic operation.
Cryptographic Algorithm	An algorithm that transforms data in order to hide or reveal its information content.
Data Integrity	The property that data has not been altered or destroyed in an unauthorised manner.
Decipherment	The reversal of a corresponding encipherment.
DEM1	A family of data encapsulation mechanisms defined in ISO/IEC 18033-2.
Digital Signature	An asymmetric cryptographic transformation of data that allows the recipient of the data to prove the origin and integrity of the data, and protect the sender and the recipient of the data against forgery by third parties, and the sender against forgery by the recipient.
Dynamic Data Authentication	A form of offline dynamic data authentication
Elliptic Curve Cryptography	Public key cryptography based on the algebraic structure of elliptic curves over finite fields.
Encipherment	The reversible transformation of data by a cryptographic algorithm to produce ciphertext.
Exclusive-OR	Binary addition with no carry, giving the following values: 0 + 0 = 0 0 + 1 = 1 1 + 0 = 1 1 + 1 = 0
Extended Data Authentication	A form of offline dynamic data authentication.
Facial Verification	The process of determining that the face presented is valid.
Financial Transaction	The act between a cardholder and a merchant or acquirer that results in the exchange of goods or services against payment.

Finger Verification	The process of determining that the finger presented is valid.
Function	A process accomplished by one or more commands and resultant actions that are used to perform all or part of a transaction.
Hash Function	A function that maps strings of bits to fixed-length strings of bits, satisfying the following two properties: • It is computationally infeasible to find for a given output an input which maps to this output. • It is computationally infeasible to find for a given input a second input that maps to the same output. Additionally, if the hash function is required to be collision-resistant, it must also satisfy the following property: • It is computationally infeasible to find any two distinct inputs that map to the same output.
Hash Result	The string of bits that is the output of a hash function.
I2OSP	An integer to octet string conversion primitive function defined in ISO/IEC 18033-2.
Integrated Circuit(s)	Electronic component(s) designed to perform processing and/or memory functions.
Integrated Circuit(s) Card	A card into which one or more integrated circuits are inserted to perform processing and memory functions.
Interface Device	That part of a terminal into which the ICC is inserted, including such mechanical and electrical devices as may be considered part of it.
Iris Verification	The process of determining that the iris presented is valid.
Issuer Action Code	Any of the following, which reflect the issuer-selected action to be taken upon analysis of the TVR: • Issuer Action Code – Default • Issuer Action Code – Denial • Issuer Action Code – Online

Kernel	The set of functions required to be present on every terminal implementing a specific interpreter. The kernel contains device drivers, interface routines, security and control functions, and the software for translating from the virtual machine language to the language used by the real machine. In other words, the kernel is the implementation of the virtual machine on a specific real machine.
Key	A sequence of symbols that controls the operation of a cryptographic transformation.
Key Introduction	The process of generating, distributing, and beginning use of a key pair.
Key Withdrawal	The process of removing a key from service as part of its revocation.
Keypad	Arrangement of numeric, command, and, where required, function and/or alphanumeric keys laid out in a specific manner.
Library	A set of high-level software functions with a published interface, providing general support for terminal programs and/or applications.
Logical Compromise	The compromise of a key through application of improved cryptanalytic techniques, increases in computing power, or combination of the two.
Magnetic Stripe	The stripe containing magnetically encoded information.
Message	A string of bytes sent by the terminal to the card or vice versa, excluding transmission-control characters.
Message Authentication Code	A symmetric cryptographic transformation of data that protects the sender and the recipient of the data against forgery by third parties.
Nibble	The four most significant or least significant bits of a byte.
Offline Data Encipherment	Offline encipherment of data, in particular for cardholder PIN and biometric data. See Book 2.
Padding	Appending extra bits to either side of a data string.
Palm Verification	The process of determining that the palm presented is valid.

Path	Concatenation of file identifiers without delimitation.
Payment System Environment	A logical construct within the ICC, the entry point to which is a Directory Definition File (DDF) named '1PAY.SYS.DDF01'. This DDF contains a Payment System Directory which in turn contains entries for one or more Application Definition Files (ADFs) which are formatted according to this specification.
Physical Compromise	The compromise of a key resulting from the fact that it has not been securely guarded, or a hardware security module has been stolen or accessed by unauthorised persons.
PIN Pad	Arrangement of numeric and command keys to be used for personal identification number (PIN) entry. Also known as a “PIN Entry Device” (PED).
Plaintext	Unenciphered information.
Potential Compromise	A condition where cryptanalytic techniques and/or computing power has advanced to the point that compromise of a key of a certain length is feasible or even likely.
Private Key	That key of an entity’s asymmetric key pair that should only be used by that entity. In the case of a digital signature scheme, the private key defines the signature function.
Public Key	That key of an entity’s asymmetric key pair that can be made public. In the case of a digital signature scheme, the public key defines the verification function.
Public Key Certificate	The public key information of an entity signed by the certification authority and thereby rendered unforgeable.
Response	A message returned by the ICC to the terminal after the processing of a command message received by the ICC.
RSA-KEM	A family of key encapsulation mechanisms defined in ISO/IEC 18033-2.
RSATransform	The RSA exponentiation that is used for encryption and decryption, and generating and verifying a signature.

Script	A command or a string of commands transmitted by the issuer to the terminal for the purpose of being sent serially to the ICC as commands.
Secret Key	A key used with symmetric cryptographic techniques and usable only by a set of specified entities.
Socket	An execution vector defined at a particular point in an application and assigned a unique number for reference.
Static Data Authentication	Offline static data authentication
Symmetric Cryptographic Technique	A cryptographic technique that uses the same secret key for both the originator's and recipient's transformation. Without knowledge of the secret key, it is computationally infeasible to compute either the originator's or the recipient's transformation.
Template	Value field of a constructed data object, defined to give a logical grouping of data objects.
Terminal	The device used in conjunction with the ICC at the point of transaction to perform a financial transaction. The terminal incorporates the interface device and may also include other components and interfaces such as host communications.
Terminal Action Code	Any of the following, which reflect the acquirer-selected action to be taken upon analysis of the TVR: • Terminal Action Code – Default • Terminal Action Code – Denial • Terminal Action Code – Online
Terminate Card Session	End the card session by deactivating the IFD contacts according to EMV Contact Interface Specification and displaying a message indicating that the ICC cannot be used to complete the transaction.
Terminate Transaction	Stop the current application and deactivate the card.
Transaction	An action taken by a terminal at the user's request. For a POS terminal, a transaction might be payment for goods, etc. A transaction selects among one or more applications as part of its processing flow.

Transaction Certificate	An Application Cryptogram generated by the card when accepting a transaction.
Virtual Machine	A theoretical microprocessor architecture that forms the basis for writing application programs in a specific interpreter software implementation.
Voice Verification	The process of determining that the voice presented is valid.

4 Abbreviations, Notations, Conventions, and Terminology

4.1 Abbreviations

a	Alphabetic (see section 4.3, Data Element Format Conventions)
AAC	Application Authentication Cryptogram
AAD	Additional Authenticated Data
AC	Application Cryptogram
ADF	Application Definition File
AEF	Application Elementary File
AES	Advanced Encryption Standard
AFL	Application File Locator
AID	Application Identifier
AIP	Application Interchange Profile
an	Alphanumeric (see section 4.3)
ans	Alphanumeric Special (see section 4.3)
APDU	Application Protocol Data Unit
API	Application Program Interface
ARC	Authorisation Response Code
ARPC	Authorisation Response Cryptogram
ARQC	Authorisation Request Cryptogram
ASI	Application Selection Indicator
ASN	Abstract Syntax Notation
ATC	Application Transaction Counter
ATM	Automated Teller Machine
ATR	Answer to Reset

AUC	Application Usage Control
b	Binary (see section 4.3)
BCD	Binary Coded Decimal
BDB	Biometric Data Block
BEK	Biometric Encryption Key
BER	Basic Encoding Rules (defined in ISO/IEC 8825-1)
BHT	Biometric Header Template
BIC	Bank Identifier Code
BIT	Biometric Information Template
BMK	Biometric MAC Key
CA	Certification Authority
CAD	Card Accepting Device
C-APDU	Command APDU
CBC	Cipher Block Chaining
CBEFF	Common Biometric Exchange Formats Framework
CCD	Common Core Definitions
CCI	Common Core Identifier
CCYYMMDD	Year (4 digits), Month, Day
CDA	Combined DDA/Application Cryptogram Generation
CDOL	Card Risk Management Data Object List
CID	Cryptogram Information Data
CLA	Class Byte of the Command Message
cn	Compressed Numeric (see section 4.3)
CPU	Central Processing Unit
CRL	Certificate Revocation List
CSU	Card Status Update
CV	Cryptogram Version
CV Rule	Cardholder Verification Rule

CVM	Cardholder Verification Method
CVR	Card Verification Results
DDA	Dynamic Data Authentication
DDF	Directory Definition File
DDOL	Dynamic Data Authentication Data Object List
DES	Data Encryption Standard
DF	Dedicated File
DIR	Directory
DOL	Data Object List
ECB	Electronic Code Book
EC-SDSA	Elliptic Curve Schnorr Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
EF	Elementary File
EN	European Norm
FC	Format Code
FCI	File Control Information
Hex	Hexadecimal
HHMMSS	Hours, Minutes, Seconds
HMAC	Keyed-hash Message Authentication Code
I/O	Input/Output
IAC	Issuer Action Code (Denial, Default, Online)
IAD	Issuer Application Data
IBAN	International Bank Account Number
IC	Integrated Circuit
ICC	Integrated Circuit(s) Card
ICCD	Issuer Certified Card Data
IEC	International Electrotechnical Commission
IFD	Interface Device

IIN	Issuer Identification Number
IINE	Issuer Identification Number Extended
INS	Instruction Byte of Command Message
ISO	International Organization for Standardization
KD	Key Derivation
KDF	Key Derivation Function
K_M	Master Key
K_S	Session Key
L	Length
l.s.	Least Significant
Lc	Exact Length of Data Sent by the TAL in a Case 3 or 4 Command
LCOL	Lower Consecutive Offline Limit
L_{DD}	Length of the ICC Dynamic Data
Le	Maximum Length of Data Expected by the TAL in Response to a Case 2 or 4 Command
Lr	Length of Response Data Field
LRC	Longitudinal Redundancy Check
M	Mandatory
m.s.	Most Significant
MAC	Message Authentication Code
max.	Maximum
MF	Master File
MK	ICC Master Key for session key generation
MMDD	Month, Day
MMYY	Month, Year
n	Numeric (see section 4.3)
N_{CA}	Length of the Certification Authority Public Key Modulus
NF	Norme Française

N_{FIELD}	Length of a finite field element
N_{HASH}	Output length of a hash function
N_I	Length of the Issuer Public Key Modulus
N_{IC}	Length of the ICC Public Key Modulus
NIST	National Institute for Standards and Technology
N_{PE}	Length of the ICC PIN Encipherment Public Key Modulus
N_{SIG}	Length of an ECC Digital Signature
O	Optional
O/S	Operating System
ODA	Offline Data Authentication
ODE	Offline Data Encipherment
P1	Parameter 1
P2	Parameter 2
PAN	Primary Account Number
PAR	Payment Account Reference
PC	Personal Computer
P_{CA}	Certification Authority Public Key
PDOL	Processing Options Data Object List
P_I	Issuer Public Key
P_{IC}	ICC Public Key
PIN	Personal Identification Number
PIX	Proprietary Application Identifier Extension
POS	Point of Service
pos.	Position
PSE	Payment System Environment
R-APDU	Response APDU
RFU	Reserved for Future Use
RID	Registered Application Provider Identifier

RSA	Rivest, Shamir, Adleman Algorithm
S_{CA}	Certification Authority Private Key
SDA	Static Data Authentication
SDAD	Signed Dynamic Application Data
SFI	Short File Identifier
SHA-1	Secure Hash Algorithm 1
SHA-2	Secure Hash Algorithm 2 (includes SHA-256 and SHA-512)
SHA-256	Secure Hash Algorithm 256
SHA-3	Secure Hash Algorithm 3
S_I	Issuer Private Key
S_{IC}	ICC Private Key
SK	Session Key
SW1	Status Byte One
SW2	Status Byte Two
TAA	Terminal Action Analysis
TAC	Terminal Action Code(s) (Default, Denial, Online)
TAL	Terminal Application Layer
TC	Transaction Certificate
TDOL	Transaction Certificate Data Object List
TLV	Tag Length Value
TPDU	Transport Protocol Data Unit
TSI	Transaction Status Information
TVR	Terminal Verification Results
UCOL	Upper Consecutive Offline Limit
UL	Underwriters Laboratories Incorporated
UN	Unpredictable Number
var.	Variable (see section 4.3)
XDA	Extended Data Authentication

YYMM	Year, Month
YYMMDD	Year, Month, Day

4.2 Notations

'0' to '9' and 'A' to 'F'	16 hexadecimal characters
xx	Any value
A := B	A is assigned the value of B
A = B	Value of A is equal to the value of B
A ≡ B mod n	Integers A and B are congruent modulo the integer n, that is, there exists an integer d such that $(A - B) = dn$
A mod n	The reduction of the integer A modulo the integer n, that is, the unique integer r, $0 \leq r < n$, for which there exists an integer d such that $A = dn + r$
A / n	The integer division of A by n, that is, the unique integer d for which there exists an integer r, $0 \leq r < n$, such that $A = dn + r$
Y := ALG(K)[X]	Encipherment of a data block X with a block cipher as specified in Book 2 section A1, using a secret key K
$X = ALG^{-1}(K)[Y]$	Decipherment of a data block Y with a block cipher as specified in Book 2 section A1, using a secret key K
$Y := Sign\ (S_K)[X]$	The signing of a data block X with an asymmetric reversible algorithm as specified in Book 2 section A2, using the private key S_K
$X = Recover(P_K)[Y]$	The recovery of the data block X with an asymmetric reversible algorithm as specified in Book 2 section A2, using the public key P_K
C := (A || B)	The concatenation of an n-bit number A and an m-bit number B, which is defined as $C = 2^m A + B$.
Leftmost	Applies to a sequence of bits, bytes, or digits and used interchangeably with the term "most significant". If C = (A || B) as above, then A is the leftmost n bits of C.

Rightmost	Applies to a sequence of bits, bytes, or digits and used interchangeably with the term “least significant”. If C = (A || B) as above, then B is the rightmost m bits of C.
H := Hash[MSG]	Hashing of a message MSG of arbitrary length using a 160-bit hash function
X ⊕ Y	The symbol '⊕' denotes bit-wise exclusive-OR and is defined as follows: X ⊕ Y — The bit-wise exclusive-OR of the data blocks X and Y. If one data block is shorter than the other, then it is first padded to the left with sufficient binary zeros to make it the same length as the other.
MIN (x, y)	The smaller of values x and y.

4.3 Data Element Format Conventions

The EMV specifications use the following data element formats:

a	Alphabetic data elements contain a single character per byte. The permitted characters are alphabetic only (a to z and A to Z, upper and lower case).
an	Alphanumeric data elements contain a single character per byte. The permitted characters are alphabetic (a to z and A to Z, upper and lower case) and numeric (0 to 9). There is one exception: The permitted characters for Payment Account Reference are alphabetic *upper case* (A to Z) and numeric (0 to 9).
ans	Alphanumeric Special data elements contain a single character per byte. The permitted characters and their coding are shown in the Common Character Set table in Book 4 Annex B. There is one exception: The permitted characters for Application Preferred Name are the non-control characters defined in the ISO/IEC 8859 part designated in the Issuer Code Table Index associated with the Application Preferred Name.
b	These data elements consist of either unsigned binary numbers or bit combinations that are defined elsewhere in the specification. Binary example: The Application Transaction Counter (ATC) is defined as “b” with a length of two bytes. An ATC value of 19 is stored as Hex '00 13'. Bit combination example: Processing Options Data Object List (PDOL) is defined as “b” with the format shown in Book 3 section 5.4.
cn	Compressed numeric data elements consist of two numeric digits (having values in the range Hex '0'–'9') per byte. These data elements are left justified and padded with trailing hexadecimal 'F's. Example: The Application Primary Account Number (PAN) is defined as “cn” with a length of up to ten bytes. A value of 1234567890123 may be stored in the Application PAN as Hex '12 34 56 78 90 12 3F FF' with a length of 8.
n	Numeric data elements consist of two numeric digits (having values in the range Hex '0' – '9') per byte. These digits are right justified and padded with leading hexadecimal zeroes. Other specifications sometimes refer to this data format as Binary Coded Decimal (“BCD”) or unsigned packed. Example: Amount, Authorised (Numeric) is defined as “n 12” with a length of six bytes. A value of 12345 is stored in Amount, Authorised (Numeric) as Hex '00 00 00 01 23 45'.
var.	Variable data elements are variable length and may contain any bit combination. Additional information on the formats of specific variable data elements is available elsewhere.

4.4 Terminology

business agreement	An agreement reached between a payment system and its business partner(s).
proprietary	Not defined in this specification and/or outside the scope of this specification
shall	Denotes a mandatory requirement
should	Denotes a recommendation

Part II

Data Elements and Commands

5 Data Elements and Files

An application in the Integrated Circuit Card (ICC) includes a set of items of information. These items of information may be accessible to the terminal after a successful application selection (see Book 1 section 12).

An item of information is called a data element. A data element is the smallest piece of information that may be identified by a name, a description of logical content, a format, and a coding.

5.1 Data Elements Associated with Financial Transaction Interchange

The data elements dictionary defined in Annex A includes those data elements that may be used for financial transaction interchange. Data elements not specified in Annex A are outside the scope of these specifications.

Any additional data element transmitted in the response to the SELECT command (for example, O/S Manufacturer proprietary data) is placed in the field “FCI Issuer Discretionary Data” (tag 'BF0C').

5.2 Data Objects

A data object consists of a tag, a length, and a value. A tag uniquely identifies a data object within the environment of an application. The length is the length of the value field of the data object. The value field of a data object may consist of either a single data element or one or more data objects. When a data object encapsulates a single data element, it is called a primitive data object. When a data object encapsulates one or more data objects, it is called a constructed data object. Specific tags are assigned to the constructed data objects with a specific meaning in the environment of an application according to this specification. The value field of such constructed data objects is a context-specific template. Rules for the coding of context-specific data objects and templates are given in Annex B.

Upon receipt, the terminal shall parse all the data elements following the rules described in Annex B. The retrieved value fields of the data elements shall be stored in the terminal buffer for possible later use in the application.

The actual order of tagged data elements within templates encountered by the terminal may differ from that described in the Specifications. The terminal shall not treat different ordering of data elements as an error.

The terminal shall be capable of correctly interpreting Tag Length Value (TLV) data objects with a length field coded '00' as defined in ISO/IEC 7816. This situation can occur when a data element is personalised on a card without an actual value field. A data element with length '00' shall be treated as not present. The data element length indicated in Annex A reflects the length of the data elements when actually present on the card.

Annex A describes the mapping of data elements onto data objects and the mapping of data objects into templates when applicable.

Records are templates containing one or more primitive and/or constructed data objects.

The mapping of data objects into records is left to the discretion of the issuer. The manner in which data elements are to be used is described in Part III.

Annex B defines the tags that are reserved by this specification for EMVCo, the payment systems, and issuers. All ICC applications conforming to this specification shall comply with this coding and allocation scheme in accordance with ISO/IEC 7816-6.

5.2.1 Classes of Data Objects

Identification and coding of different classes of data objects are defined in Annex B. The tag definitions specified in that annex are according to ISO/IEC 8825 and the ISO/IEC 7816 series and apply to applications conforming to this specification.

5.3 Files

The data objects contained in data files accessible from the ICC are stored in records. The file structure and referencing method depend on the purpose of the file. The following sections describe structures and referencing methods. The layout of the data files accessible from the ICC is left to the discretion of the issuer except for the directory files described in the following section.

5.3.1 Application Elementary Files

An Application Elementary File (AEF) in the range 1-10, contains one or more primitive Basic Encoding Rules – TLV (BER-TLV) data objects grouped into constructed BER-TLV data objects (records) according to Annex B. After selecting the application, an AEF in the range 1-10 is referred to only by its SFI as described in section 5.3.2.2.

A data file referred to in this specification consists of a sequence of records addressed by record number. The data files referred to by SFIs in the range 1-10 contain only data not interpreted by the card, that is, data that is not used by the card in its internal processes. This file structure is defined as linear. It can be either linear fixed or linear variable according to ISO/IEC 7816-4. The choice is left to the issuer and does not affect the reading of the file according to this specification.

5.3.2 File Referencing

A file may be referred to by a name or a SFI depending on its type.

5.3.2.1 Referencing by Name

Any Application Definition File (ADF) or Directory Definition File (DDF) in the card is referenced by its Dedicated File (DF) name. A DF name for an ADF corresponds to the Application Identifier (AID) or contains the AID as the beginning of the DF name. Each DF name shall be unique within a given card.

5.3.2.2 Referencing by SFI

SFIs are used for the selection of AEFs. Any AEF within a given application is referenced by a SFI coded on 5 bits in the range 1 to 30. The coding of the SFI is described in every command that uses it.

Table 1 describes the assignment of SFIs for an EMV application:

Value	Meaning
1-10	Governed by this specification
11-20	Payment system-specific
21-30	Issuer-specific

Table 1: Structure of SFI

An SFI shall be unique within an application. The coding of SFIs within the range 1 to 10 is used to address AEFs governed by this specification.

5.4 Rules for Using a Data Object List (DOL)

In several instances, the terminal is asked to build a flexible list of data elements to be passed to the card under the card's direction. To minimise processing within the ICC, such a list is not TLV encoded but is a single constructed field built by concatenating several data elements together. Since the elements of the constructed field are not TLV encoded, it is imperative that the ICC knows the format of this field when the data is received. This is achieved by including a Data Object List (DOL) in the ICC, specifying the format of the data to be included in the constructed field. DOLs currently used in this specification include:

- The Processing Options Data Object List (PDOL) used with the GET PROCESSING OPTIONS command
  Note: An ICC supporting XDA or ECC ODE may use PDOL to obtain Terminal Capabilities (tag '9F33') to determine whether the terminal supports XDA.
- The Card Risk Management Data Object Lists (CDOL1 and CDOL2) used with the GENERATE APPLICATION CRYPTOGRAM (AC) command
- The Transaction Certificate Data Object List (TDOL) used to generate a TC Hash Value
- The Dynamic Data Authentication Data Object List (DDOL) used with the INTERNAL AUTHENTICATE command

This section describes the rules for constructing a field using a DOL supplied by the card.

A DOL is a concatenated list of entries, with each entry representing a single data element to be included in the constructed field. The format of each entry is a one- or two-byte tag identifying the desired data object, followed by a one-byte length which represents the number of bytes the field shall occupy in the command data. Only tags representing primitive data objects constructed according to Annex B shall be used in DOLs. Primitive data objects contained within terminal sourced constructed data objects cannot be requested using DOLs.

The terminal shall complete the following steps to build the constructed field:

1. Read the DOL from the ICC.
2. Concatenate all data elements listed in the DOL. The following rules apply to this concatenation:
    - If the tag of any data object identified in the DOL is unknown to the terminal or represents a constructed data object, the terminal shall provide a data element with the length specified and a value of all hexadecimal zeroes.
    - If a data object is in the list and is meaningful to the terminal but represents optional static data that is absent from the terminal, the portion of the command field representing the data object shall be filled with hexadecimal zeroes.
    - If the length specified in the DOL entry is less than the length of the actual data object, the leftmost bytes of the data element shall be truncated if the data object has numeric (n) format,[1] or the rightmost bytes of the data shall be truncated for any other format.
    - If the length specified in the DOL entry is greater than the length of the actual data, the actual data shall be padded:
        - with leading hexadecimal zeroes if the data has numeric format
        - with trailing hexadecimal 'FF's if the data has compressed numeric (cn [1]) format
        - with trailing hexadecimal zeroes for any other format (an, ans, or b including bit combination data)[1]
    - If a data object is in the list and is meaningful to the terminal but represents data that is not applicable to the current transaction, the portion of the command field representing the data object shall be filled with hexadecimal zeroes.

The completed list of data elements shall be concatenated in the sequence in which the corresponding data objects appear in the DOL.

[1] See section 4.3, Data Element Format Convention.

6 Commands for Financial Transaction

6.1 Command APDU Format

The command Application Protocol Data Unit (APDU) consists of a mandatory header of four bytes followed by a conditional body of variable length, as shown in Figure 1:

CLA	INS	P1	P2	Lc	Data	Le
← Mandatory Header [2] →				← Conditional Body →		

Figure 1: Command APDU Structure

The number of data bytes sent in the command APDU (C-APDU) is denoted by Lc (length of command data field).

The maximum number of data bytes expected in the response APDU is denoted by length of expected data (Le). When Le is present and contains the value zero, the maximum number of available data bytes (up to 256) is expected. When required in a command message, Le shall always be set to '00'.

6.2 Response APDU Format

The response APDU format consists of a conditional body of variable length followed by a mandatory trailer of two bytes, as shown in Figure 2:

Data	SW1	SW2
← Body →	← Trailer →	

Figure 2: Response APDU Structure

The data field of a response APDU is an object structured as defined in Annex B. The detailed coding of the objects is provided with the commands described in subsequent sub-clauses.

[2] CLA = Class Byte of the Command Message

INS = Instruction Byte of Command Message

P1 = Parameter 1

P2 = Parameter 2

6.3 Coding Conventions

This section defines the coding of the header and the body of the messages (command and response).

6.3.1 Coding of the Class Byte

The most significant nibble of the class byte indicates the type of command as shown in Table 2:

Hex	Meaning
'0'	Inter-industry command
'8'	Proprietary to this specification
Any other value	Outside the scope of this specification

Table 2: Most Significant Nibble of the Class Byte

A command proprietary to this specification is introduced by the most significant nibble of the class byte set to 8; in other words, the structure of the command and response messages is according to ISO/IEC 7816-4, and the coding of the messages is defined within the context of these specifications.

The least significant nibble of the class byte indicates secure messaging and logical channel mechanisms, according to ISO/IEC 7816-4.

6.3.2 Coding of the Instruction Byte

The INS byte of a command is structured according to EMV Contact Interface Specification. Table 3 indicates the coding of INS and its relationship to CLA:

CLA	INS	Meaning
'8x'	'1E'	APPLICATION BLOCK
'8x'	'18'	APPLICATION UNBLOCK
'8x'	'16'	CARD BLOCK
'0x'	'82'	EXTERNAL AUTHENTICATE
'8x'	'AE'	GENERATE APPLICATION CRYPTOGRAM
'0x'	'84'	GET CHALLENGE
'8x'	'CA'	GET DATA
'8x'	'A8'	GET PROCESSING OPTIONS
'0x'	'88'	INTERNAL AUTHENTICATE
'8x'	'24'	PERSONAL IDENTIFICATION NUMBER (PIN) CHANGE/UNBLOCK
'0x'	'B2'	READ RECORD
'0x'	'A4'	SELECT
'0x'	'20'	VERIFY
'8x'	'Dx'	RFU for the payment systems
'8x'	'Ex'	RFU for the payment systems
'9x'	'xx'	RFU for manufacturers for proprietary INS coding
'Ex'	'xx'	RFU for issuers for proprietary INS coding

Table 3: Coding of the Instruction Byte

Note: Additional INS codes may be assigned in the future by EMVCo using class '8x'. It is strongly recommended that application providers not define proprietary commands in class '8x' when they are to be used in the context of these specifications, so that collision is avoided.

6.3.3 Coding of Parameter Bytes

The parameter bytes P1 P2 may have any value. If not used, a parameter byte has the value '00'.

6.3.4 Coding of Data Field Bytes

When present, a command APDU message data field consists of a string of data elements.

When present, a response APDU message data field consists of a data object or a string of data objects encapsulated in a template according to Annex B.

6.3.5 Coding of the Status Bytes

The status bytes SW1 SW2 are returned by the transport layer to the application layer in any response message and denote the processing state of the command. The coding of the status words is structured as illustrated in Figure 3:

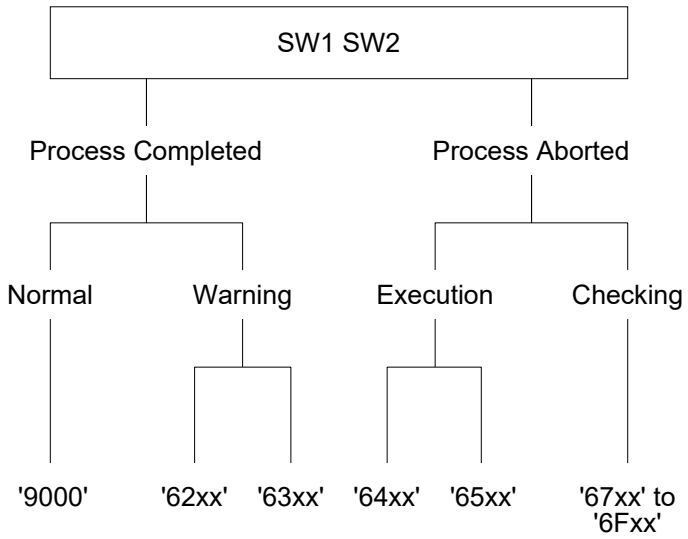


Figure 3: Structural Scheme of Status Bytes

SW1	SW2	Meaning
		Normal processing
'90'	'00'	Process completed (any other value for SW2 is RFU)
		Warning processing
'62'	'83'	State of non-volatile memory unchanged; selected file invalidated
'63'	'00'	State of non-volatile memory changed; authentication failed
'63'	'Cx'	State of non-volatile memory changed; counter provided by 'x' (from 0-15)
		Checking errors
'68'	'00'	Command chaining failed
'68'	'83'	Last command of chain was expected but not received
'68'	'84'	Command chaining not supported
'69'	'83'	Command not allowed; authentication method blocked
'69'	'84'	Command not allowed; referenced data invalidated
'69'	'85'	Command not allowed; conditions of use not satisfied
'6A'	'81'	Wrong parameter(s) P1 P2; function not supported
'6A'	'82'	Wrong parameter(s) P1 P2; file not found
'6A'	'83'	Wrong parameter(s) P1 P2; record not found
'6A'	'88'	Referenced data (data objects) not found

Table 4: Coding of Status Bytes SW1 SW2

The following values of SW1 SW2 are described in EMV Contact Interface Specification as they apply to the Transport Protocol Data Unit (TPDU) and are not returned to the APDU:

- '61xx': SW2 indicates the number of response bytes still available.
- '6Cxx': Wrong length Le, SW2 indicates the exact length.

SW1 = '6x' or '90' denotes a normal, warning, or error condition coded according to ISO/IEC 7816-4.

Other values of SW1 returned by the ICC are not supported by EMV Contact Interface Specification.

Table 5 shows the coding of the SW1 SW2 status bytes which this specification requires to be returned in response to specific conditions. The card may generate status bytes not listed in this table for error and warning conditions not specified in Part III.

SW1	SW2	APPLICATION BLOCK	APPLICATION UNBLOCK	CARD BLOCK	EXTERNAL AUTHENTICATE	GENERATE APPLICATION CRYPTOGRAM	GET CHALLENGE	GET DATA	GET PROCESSING OPTIONS	INTERNAL AUTHENTICATE	PIN CHANGE/UNBLOCK	READ RECORD	SELECT	VERIFY
'62'	'83'												X	
'63'	'00'				X									
'63'	'Cx'													X
'68'	'00'													X
'68'	'83'													X
'68'	'84'													X
'69'	'83'													X
'69'	'84'													X
'69'	'85'				X	X			X					
'6A'	'81'							X					X	
'6A'	'82'												X	
'6A'	'83'											X		
'6A'	'88'							X						

Table 5: Allocation of Status Bytes

The following convention is used in the table:

X = Allowed response code, for which a dedicated action shall be taken or which has a special meaning for an EMV compliant application. The meaning of the action is explained in section 10.

If during transaction processing as described in Part III, the card returns a value for SW1 SW2 other than those specified in Table 5, and no other action is indicated elsewhere in these specifications, the transaction shall be terminated. For example, if the application reads records in a file that contains four records and, in response to the READ RECORD command for record 5, the card returns SW1 SW2 = '6400' instead of SW1 SW2 = '6A83', then the transaction would be terminated.

If during the processing of the GET DATA command, defined in section 6.5.7, the card returns an error condition, the terminal shall proceed as indicated in section 10.6.3 (for terminal velocity checking) or in Book 4 section 6.3.4.1 (for cardholder verification processing).

If during the processing of an issuer script command, as defined in section 10.10, the card returns a warning condition (SW1 SW2 = '62XX' or '63xx'), the terminal shall continue with the next command from the Issuer Script (if any).

For the EXTERNAL AUTHENTICATE command, SW1 SW2 = '6300' means 'Authentication Failed'.

6.3.6 Coding of RFU Data

The coding of data (bits and bytes) indicated as RFU and marked as 'x' in the tables of the specifications shall be set to zero unless otherwise stated.

To allow for migration and support of new functionality, the ICC and the terminal shall not verify the data indicated as RFU.

6.4 Logical Channels

A logical channel establishes and maintains the link between an application in the terminal and an application in the card.

A card may support more than one logical channel but only the basic logical channel is supported by this specification. This limits to one the number of concurrent applications according to this specification.

6.5 Commands

This section describes the following APDU command-response pairs:

- APPLICATION BLOCK (post-issuance command)
- APPLICATION UNBLOCK (post-issuance command)
- CARD BLOCK (post-issuance command)
- EXTERNAL AUTHENTICATE
- GENERATE APPLICATION CRYPTOGRAM
- GET CHALLENGE
- GET DATA
- GET PROCESSING OPTIONS
- INTERNAL AUTHENTICATE
- PIN CHANGE/UNBLOCK (post-issuance command)
- READ RECORD
- VERIFY

The post-issuance commands shall only be sent using script processing (see section 10.10) and secure messaging as specified in Book 2.

6.5.1 APPLICATION BLOCK Command-Response APDUs

6.5.1.1 Definition and Scope

The APPLICATION BLOCK command is a post-issuance command that invalidates the currently selected application.

Following the successful completion of an APPLICATION BLOCK command:

- An invalidated application shall return the status bytes SW1 SW2 = '6283' ('Selected file invalidated') in response to a SELECT command.
- An invalidated application shall return only an Application Authentication Cryptogram (AAC) as AC in response to a GENERATE AC command.

6.5.1.2 Command Message

The APPLICATION BLOCK command message is coded as shown in Table 6:

Code	Value
CLA	'8C' or '84'; coding according to the secure messaging specified in Book 2
INS	'1E'
P1	'00'; all other values are RFU
P2	'00'; all other values are RFU
Lc	Number of data bytes
Data	Message Authentication Code (MAC) data component; coding according to the secure messaging specified in Book 2
Le	Not present

Table 6: APPLICATION BLOCK Command Message

6.5.1.3 Data Field Sent in the Command Message

The data field of the command message contains the MAC data component coded according to the secure messaging format specified in Book 2.

6.5.1.4 Data Field Returned in the Response Message

No data field is returned in the response message.

6.5.1.5 Processing State Returned in the Response Message

'9000' indicates a successful execution of the command, whether the application was already invalidated or not.

6.5.2 APPLICATION UNBLOCK Command-Response APDUs

6.5.2.1 Definition and Scope

The APPLICATION UNBLOCK command is a post-issuance command that rehabilitates the currently selected application.

Following the successful completion of an APPLICATION UNBLOCK command, the restrictions imposed by the APPLICATION BLOCK command are removed.

6.5.2.2 Command Message

The APPLICATION UNBLOCK command message is coded as shown in Table 7.

Code	Value
CLA	'8C' or '84'; coding according to the secure messaging specified in Book 2
INS	'18'
P1	'00'; all other values are RFU
P2	'00'; all other values are RFU
Lc	Number of data bytes
Data	MAC data component; coding according to the secure messaging specified in Book 2
Le	Not present

Table 7: APPLICATION UNBLOCK Command Message

6.5.2.3 Data Field Sent in the Command Message

The data field of the command message contains the MAC data component coded according to the secure messaging format specified in Book 2.

6.5.2.4 Data Field Returned in the Response Message

No data field is returned in the response message.

6.5.2.5 Processing State Returned in the Response Message

'9000' indicates a successful execution of the command, whether the application was previously invalidated or not.

6.5.3 CARD BLOCK Command-Response APDUs

6.5.3.1 Definition and Scope

The CARD BLOCK command is a post-issuance command that permanently disables all applications in the ICC.

The CARD BLOCK command shall disable all applications in the ICC, including applications that may be selected implicitly.

Following the successful completion of a CARD BLOCK command, all subsequent SELECT commands shall return the status bytes SW1 SW2 = '6A81' ('Function not supported') and perform no other action.

6.5.3.2 Command Message

The CARD BLOCK command message is coded as shown in Table 8.

Code	Value
CLA	'8C' or '84'; coding according to the secure messaging specified in Book 2
INS	'16'
P1	'00'; all other values are RFU
P2	'00'; all other values are RFU
Lc	Number of data bytes
Data	MAC data component; coding according to the secure messaging specified in Book 2
Le	Not present

Table 8: CARD BLOCK Command Message

6.5.3.3 Data Field Sent in the Command Message

The data field of the command message contains the MAC data component coded according to the secure messaging format specified in Book 2.

6.5.3.4 Data Field Returned in the Response Message

No data field is returned in the response message.

6.5.3.5 Processing State Returned in the Response Message

'9000' indicates a successful execution of the command, whether the card was already blocked or not.

6.5.4 EXTERNAL AUTHENTICATE Command-Response APDUs

6.5.4.1 Definition and Scope

The EXTERNAL AUTHENTICATE command asks the application in the ICC to verify a cryptogram.

The ICC returns the processing state of the command.

6.5.4.2 Command Message

The EXTERNAL AUTHENTICATE command message is coded as shown in Table 9:

Code	Value
CLA	'00'
INS	'82'
P1	'00'
P2	'00'
Lc	8-16
Data	Issuer Authentication Data
Le	Not present

Table 9: EXTERNAL AUTHENTICATE Command Message

The reference of the algorithm (P1) of the EXTERNAL AUTHENTICATE command is coded '00', which means that no information is given. The reference of the algorithm either is known before issuing the command or is provided in the data field.

6.5.4.3 Data Field Sent in the Command Message

The data field of the command message contains the value field of tag '91' coded as follows:

- mandatory first 8 bytes containing the cryptogram
- optional additional 1-8 bytes are proprietary

6.5.4.4 Data Field Returned in the Response Message

No data field is returned in the response message.

6.5.4.5 Processing State Returned in the Response Message

'9000' indicates a successful execution of the command.

'6300' indicates ‘Issuer authentication failed’.

For further information, see Annex F.

6.5.5 GENERATE APPLICATION CRYPTOGRAM Command-Response APDUs

6.5.5.1 Definition and Scope

The GENERATE AC command sends transaction-related data to the ICC, which computes and returns a cryptogram. This cryptogram shall either be an Application Cryptogram (AC) as specified in this specification or a proprietary cryptogram. In both cases, the cryptogram shall be of a type specified in Table 10 (for more details, see section 9).

This command is also used when performing the Combined DDA/Application Cryptogram Generation (CDA) function as described in Book 2 section 6.6 and when performing the XDA offline data authentication function as described in Book 2 section 12.

Type	Abbreviation	Meaning
Application Authentication Cryptogram	AAC	Transaction declined
Authorisation Request Cryptogram	ARQC	Online authorisation requested
Transaction Certificate	TC	Transaction approved

Table 10: GENERATE AC Cryptogram Types

The cryptogram returned by the ICC may differ from that requested in the command message according to an internal process in the ICC (as described in section 9).

6.5.5.2 Command Message

The GENERATE AC command message is coded as shown in Table 11:

Code	Value
CLA	'80'
INS	'AE'
P1	Reference control parameter (see Table 12)
P2	'00'
Lc	var.
Data	Transaction-related data
Le	'00'

Table 11: GENERATE AC Command Message

The reference control parameter of the GENERATE AC command is coded as shown in Table 12:

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0							AAC
0	1							TC
1	0							ARQC
1	1							RFU
		x						RFU
			0	0				No CDA/XDA signature requested
			1	0				CDA signature requested
			0	1				XDA signature requested
			1	1				RFU
					x	x	x	RFU

Table 12: GENERATE AC Reference Control Parameter

6.5.5.3 Data Field Sent in the Command Message

The content of the data field of the command message is coded according to the rules for the data object list as defined in section 5.4.

6.5.5.4 Data Field Returned in the Response Message

The data field of the response message consists of a BER-TLV coded data object. The coding of the data object shall be according to one of the following two formats.

- **Format 1:** The data object returned in the response message is a primitive data object with tag equal to '80'. The value field consists of the concatenation without delimiters (tag and length) of the value fields of the data objects specified in Table 13:

Value	Presence
Cryptogram Information Data (CID)	M
Application Transaction Counter (ATC)	M
Application Cryptogram (AC)	M
Issuer Application Data (IAD)	O

Table 13: Format 1 GENERATE AC Response Message Data Field

- **Format 2:** The data object returned in the response message is a constructed data object with tag equal to '77'. The value field may contain several BER-TLV coded objects, but shall always include the Cryptogram Information Data, the Application Transaction Counter, and the cryptogram computed by the ICC (either an AC or a proprietary cryptogram). The utilisation and interpretation of proprietary data objects which may be included in this response message are outside the scope of these specifications.

  Format 2 shall be used if the response is being returned with a signature as specified for the CDA or XDA functions described in Book 2 sections 6.6 and 12. The data elements for the response are shown in Table 14. These data elements may be in any order.

Value	Presence
Cryptogram Information Data (CID)	M
Application Transaction Counter (ATC)	M
Application Cryptogram (AC)	C1
Issuer Application Data (IAD)	O
Other data objects	O
Signed Dynamic Application Data (SDAD)	C2

Table 14: Format 2 GENERATE AC Response Message Data Field

Note 1: Application Cryptogram (AC) is not present when CDA signature is returned.

Note 2: When the card does not perform XDA or CDA, Signed Dynamic Application Data is not included in the response.

For both Format 1 and Format 2, the Cryptogram Information Data returned by the GENERATE AC response message is coded as shown in Table 15:

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0							AAC
0	1							TC
1	0							ARQC
1	1							RFU
		x	x					Payment System-specific cryptogram
				0				No advice required
				1				Advice required
					x	x	x	Reason/advice code
					0	0	0	No information given
					0	0	1	Service not allowed
					0	1	0	PIN Try Limit exceeded
					0	1	1	Issuer authentication failed
					1	x	x	Other values RFU

Table 15: Coding of Cryptogram Information Data

For the Format 2 response template, if any mandatory data element is missing, the terminal shall terminate the transaction.

6.5.5.5 Processing State Returned in the Response Message

'9000' indicates a successful execution of the command.

6.5.6 GET CHALLENGE Command-Response APDUs

6.5.6.1 Definition and Scope

The GET CHALLENGE command is used to obtain an unpredictable number from the ICC for use in a security-related procedure.

The challenge shall be valid only for the next issued command.

6.5.6.2 Command Message

The GET CHALLENGE command message is coded as shown in Table 16:

Code	Value
CLA	'00'
INS	'84'
P1	'00'
P2	'00'
Lc	Not present
Data	Not present
Le	'00'

Table 16: GET CHALLENGE Command Message

6.5.6.3 Data Field Sent in the Command Message

The data field of the command message is not present.

6.5.6.4 Data Field Returned in the Response Message

The data field of the response message contains an 8-byte unpredictable number generated by the ICC.

6.5.6.5 Processing State Returned in the Response Message

'9000' indicates a successful execution of the command.

6.5.7 GET DATA Command-Response APDUs

6.5.7.1 Definition and Scope

The GET DATA command is used to retrieve a primitive or constructed data object not encapsulated in a record within the current application.

The usage of the GET DATA command in this specification is limited to the retrieval of the following primitive or constructed data objects that are defined in Annex A and interpreted by the application in the ICC:

- ATC (tag '9F36')
- Last Online ATC Register (tag '9F13')
- PIN Try Counter (tag '9F17')
- Log Format (tag '9F4F')
- Biometric Try Counters Template (tag 'BF4C')
- Preferred Attempts Template (tag 'BF4D')

6.5.7.2 Command Message

The GET DATA command message is coded as shown in Table 17:

Code	Value
CLA	'80'
INS	'CA'
P1 P2	'9F36', '9F13', '9F17', '9F4F', 'BF4C' or 'BF4D'
Lc	Not present
Data	Not present
Le	'00'

Table 17: GET DATA Command Message

6.5.7.3 Data Field Sent in the Command Message

The data field of the command message is not present.

6.5.7.4 Data Field Returned in the Response Message

The data field of the response message contains the primitive or constructed data object referred to in P1 P2 of the command message (in other words, including its tag and its length).

6.5.7.5 Processing State Returned in the Response Message

'9000' indicates a successful execution of the command.

6.5.8 GET PROCESSING OPTIONS Command-Response APDUs

6.5.8.1 Definition and Scope

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

The ICC returns the Application Interchange Profile (AIP) and the Application File Locator (AFL).

6.5.8.2 Command Message

The GET PROCESSING OPTIONS command message is coded as shown in Table 18:

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

Table 18: GET PROCESSING OPTIONS Command Message

6.5.8.3 Data Field Sent in the Command Message

The data field of the command message is a data object coded according to the PDOL provided by the ICC, as defined in section 5.4, and is introduced by the tag '83'. When the data object list is not provided by the ICC, the terminal sets the length field of the template to zero. Otherwise, the length field of the template is the total length of the value fields of the data objects transmitted to the ICC.

Note: An ICC supporting XDA or ECC ODE may return PDOL indicating Terminal Capabilities (tag '9F33') in response to SELECT command.

6.5.8.4 Data Field Returned in the Response Message

The data field of the response message consists of a BER-TLV coded data object. The coding of the data object shall be according to one of the following two formats.

- **Format 1:** The data object returned in the response message is a primitive data object with tag equal to '80'. The value field consists of the concatenation without delimiters (tag and length) of the value fields of the AIP and the AFL. The coding of the data object returned in the response message is shown in Figure 4:

'80'	Length	Application Interchange Profile	Application File Locator

Figure 4: Format 1 GET PROCESSING OPTIONS Response Message Data Field

- **Format 2:** The data object returned in the response message is a constructed data object with tag equal to '77'. The value field may contain several BER-TLV coded objects, but shall always include the AIP and the AFL. The utilisation and interpretation of proprietary data objects which may be included in this response message are outside the scope of these specifications.

The AIP specifies the application functions that are supported by the application in the ICC and is coded according to Part III.

The AFL consists of the list, without delimiters, of files and related records for the currently selected application that shall be read according to section 10.2.

For the Format 2 response template, if any mandatory data element is missing, the terminal shall terminate the transaction.

6.5.8.5 Processing State Returned in the Response Message

'9000' indicates a successful execution of the command.

6.5.9 INTERNAL AUTHENTICATE Command-Response APDUs

6.5.9.1 Definition and Scope

The INTERNAL AUTHENTICATE command initiates the computation of the Signed Dynamic Application Data by the card using:

- the challenge data sent from the terminal and
- ICC data and
- a relevant private key stored in the card.

The ICC returns the Signed Dynamic Application Data to the terminal.

6.5.9.2 Command Message

The INTERNAL AUTHENTICATE command message is coded as shown in Table 19:

Code	Value
CLA	'00'
INS	'88'
P1	'00'
P2	'00'
Lc	Length of authentication-related data
Data	Authentication-related data
Le	'00'

Table 19: INTERNAL AUTHENTICATE Command Message

The reference of the algorithm (P1) of the INTERNAL AUTHENTICATE command is coded '00', which means that no information is given. The reference of the algorithm either is known before issuing the command or is provided in the data field.

6.5.9.3 Data Field Sent in the Command Message

The data field of the command message contains the authentication-related data proprietary to an application. It is coded according to the DDOL as defined in Book 2.

6.5.9.4 Data Field Returned in the Response Message

The data field of the response message consists of a BER-TLV coded data object. The coding of the data object shall be according to one of the following two formats.

- **Format 1:** The data object returned in the response message is a primitive data object with tag equal to '80'. The value field consists of the value field of the Signed Dynamic Application Data as specified in Book 2.
- **Format 2:** The data object returned in the response message is a constructed data object with tag equal to '77'. The value field may contain several BER-TLV coded objects, but shall always include the Signed Dynamic Application Data as specified in Book 2. The utilisation and interpretation of proprietary data objects which may be included in this response message are outside the scope of these specifications.

For the Format 2 response template, if any mandatory data element is missing, the terminal shall terminate the transaction.

Note: To ensure that the INTERNAL AUTHENTICATE response data length is within the 256 byte limit, the length of the Signed Dynamic Application Data plus the length of the TLV encoded optional data (if present) shall remain within the limits as specified in Book 2 Annex D.

6.5.9.5 Processing State Returned in the Response Message

'9000' indicates a successful execution of the command.

6.5.10 PIN CHANGE/UNBLOCK Command-Response APDUs

6.5.10.1 Definition and Scope

The PIN CHANGE/UNBLOCK command is a post-issuance command. Its purpose is to provide the issuer the capability either to unblock the PIN or to simultaneously change and unblock the reference PIN. It also provides the issuer the capability either to unblock the specified Biometric Type, or to simultaneously update the specified biometric reference template and unblock the associated Biometric Type, or to store the biometric reference templates in the card during enrolment.

If the PIN CHANGE/UNBLOCK command is used either to unblock the PIN or to simultaneously change and unblock the reference PIN, upon successful completion of the PIN CHANGE/UNBLOCK command, the card shall perform the following functions:

- The value of the PIN Try Counter shall be reset to the value of the PIN Try Limit.
- If requested, the value of the reference PIN shall be set to the new PIN value.

If PIN data is transmitted in the command it shall be enciphered for confidentiality.

Note: The reference PIN, which is stored within the card, is the one that is associated with the application and which the card uses to check the Transaction PIN Data transmitted within the VERIFY command.

If the PIN CHANGE/UNBLOCK command is used either to unblock the specified Biometric Type, or to simultaneously enroll or update the specified biometric reference template and unblock the associated Biometric Type, then upon successful completion of the command the card shall perform the following functions:

- In order to unblock the Biometric Type, the value of the Biometric (Facial, Finger, Iris, Palm, or Voice) Try Counter for the Biometric Type being updated shall be reset to the value of the associated Biometric (Facial, Finger, Iris, Palm, or Voice) Try Limit.
- If requested, the biometric reference templates in the card shall be set to or replaced by the new template in the Biometric Data Block (BDB), received in the PIN CHANGE/UNBLOCK command.

The BDB transmitted in the PIN CHANGE/UNBLOCK command shall be enciphered for confidentiality.

Due to the large size of the BDB, multiple PIN CHANGE/UNBLOCK commands are typically required, as described in section 6.5.13.

Note: The biometric reference template, which is stored within the card, is the one that is associated with the application and which the card uses to check the template captured by the biometric capture device and transmitted within the VERIFY command.

6.5.10.2 Command Message

The PIN CHANGE/UNBLOCK command message is coded as shown in Table 20.

Code	Value
CLA	'8C', '84', '9C', or '94'; coding according to the secure messaging specified in Book 2 and command chaining specified in section 6.5.13
INS	'24'
P1	'00'
P2	'00', '01', '02', '03', or '04'
Lc	Number of data bytes
Data	If P2 = '00', then the data field contains the MAC data component coded according to the secure messaging specified in Book 2. If P2 = '03', then the data field contains the Biometric Type, as shown in Table 49, and the MAC data component coded according to the secure messaging specified in Book 2. The P2 values '01', '02', and '04' are reserved for the payment systems, but the data field should contain one of the following: • Enciphered PIN data component, if present, and MAC data component; coding according to the secure messaging specified in Book 2 • The following enciphered biometric data: o Biometric Type, as shown in Table 49 o Biometric Subtype, as shown in Table 50 o Biometric Solution ID, as shown in Table 48 o Biometric Data Block (BDB) followed by the MAC data component coded according to the secure messaging specified in Book 2. If the length of the enciphered biometric data plus MAC is greater than 255 bytes, then the data should be sent to the card over several PIN CHANGE/UNBLOCK commands. The data encipherment and MACing should be applied to each command.
Le	Not present

Table 20: PIN CHANGE/UNBLOCK Command Message

P2: If P2 is equal to '00', the reference PIN is unblocked and the PIN Try Counter is reset to the PIN Try Limit. There is no PIN update, since the PIN CHANGE/UNBLOCK command does not contain a new PIN value.

If P2 is equal to '03', the try counter associated with the Biometric Type specified in the data field of the command is reset to the try limit of the specified Biometric Type. There is no biometric update, since the PIN CHANGE/UNBLOCK command does not contain a new BDB.

The usage of P2 equal to '01', '02', or '04' is reserved for payment systems.

Any other value of P2 allowing PIN/Biometric Type unblocking and/or PIN/biometric reference template changing is out of the scope of this specification and shall be described for each payment system individually.

6.5.10.3 Data Field Sent in the Command Message

The data field of the command message contains one of the following:

- The PIN data component, if present, followed by the MAC data component coded according to the secure messaging format specified in Book 2.
- The Biometric Type, Biometric Subtype, Biometric Solution ID and BDB followed by the MAC data component coded according to the secure messaging format specified in Book 2.
- The Biometric Type followed by the MAC data component coded according to the secure messaging format specified in Book 2.

6.5.10.4 Data Field Returned in the Response Message

No data field is returned in the response message.

6.5.10.5 Processing State Returned in the Response Message

'9000' indicates a successful execution of the command.

6.5.11 READ RECORD Command-Response APDUs

6.5.11.1 Definition and Scope

The READ RECORD command reads a file record in a linear file.

The response from the ICC consists of returning the record.

6.5.11.2 Command Message

The READ RECORD command message is coded as shown in Table 21:

Code	Value
CLA	'00'
INS	'B2'
P1	Record number
P2	Reference control parameter (see Table 22)
Lc	Not present
Data	Not present
Le	'00'

Table 21: READ RECORD Command Message

Table 22 defines the reference control parameter of the command message:

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x	x	x	x				SFI
					1	0	0	P1 is a record number

Table 22: READ RECORD Command Reference Control Parameter

6.5.11.3 Data Field Sent in the Command Message

The data field of the command message is not present.

6.5.11.4 Data Field Returned in the Response Message

The data field of the response message of any successful READ RECORD command contains the record read. For SFIs in the range 1-10, the record is a BER-TLV constructed data object as defined in Annex B and coded as shown in Figure 5:

'70'	Length	Record Template

Figure 5: READ RECORD Response Message Data Field

The response message to READ RECORD for SFIs in the range 11-30 is outside the scope of this specification, except as specified in section 10.3 and in Annex D.

6.5.11.5 Processing State Returned in the Response Message

'9000' indicates a successful execution of the command.

6.5.12 VERIFY Command-Response APDUs

6.5.12.1 Definition and Scope

If the CVM chosen from the CVM List is an offline PIN, as described in section 10.5, the VERIFY command initiates in the ICC the comparison of the Transaction PIN Data sent in the data field of the command with the reference PIN data associated with the application. The manner in which the comparison is performed is proprietary to the application in the ICC.

If the CVM chosen from the CVM List is any of the offline biometric verification methods, the VERIFY command initiates in the ICC the comparison of the biometric template captured by the biometric capture device and sent in the data field of the command with the biometric reference template(s) associated with the application. The manner in which the comparison is performed is proprietary to the application in the ICC.

The biometric data to be sent in the VERIFY command shall be enciphered for confidentiality, as described in Book 2 section 7.3 (for RSA) or Book 2 section 13.4 (for ECC).

Due to the large size of the biometric data to be sent in the VERIFY command, command chaining is typically required, as described in section 6.5.13.

6.5.12.2 Command Message

The VERIFY command message is coded as shown in Table 23:

Code	Value
CLA	'00' or '10'; coding according to the command chaining specified in section 6.5.13
INS	'20'
P1	'00'
P2	Qualifier of the reference data (see Table 24)
Lc	var.
Data	Transaction PIN Data or Biometric Verification Data Template
Le	Not present

Table 23: VERIFY Command Message

Table 24 defines the qualifier of the reference data (P2):

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	0	0	0	0	0	As defined in ISO/IEC 7816-4 [3]
1	0	0	0	0	0	0	0	Plaintext PIN, format as defined below
1	0	0	0	0	x	x	x	RFU for this specification
1	0	0	0	1	0	0	0	Enciphered PIN (RSA), format as defined in Book 2 section 7
1	0	0	0	1	0	0	1	Enciphered Biometric (RSA), format as defined in Book 2 section 7
1	0	0	0	1	0	1	0	Enciphered PIN (ECC), format as defined in Book 2 section 13
1	0	0	0	1	0	1	1	Enciphered Biometric (ECC), format as defined in Book 2 section 13
1	0	0	0	1	1	x	x	RFU for the individual payment systems
1	0	0	1	x	x	x	x	RFU for the issuer

Table 24: VERIFY Command Qualifier of Reference Data (P2)

The processing of the VERIFY command in the ICC will be defined in combination with the CVM rules as specified in section 10.5.

[3] The value of P2 = '00' is not used by this specification.

The plaintext offline PIN block shall be formatted as follows:

C	N	P	P	P	P	P/F	P/F	P/F	P/F	P/F	P/F	P/F	P/F	F	F

where:

	Name	Value
C	Control field	4 bit binary number with value of 0010 (Hex '2')
N	PIN length	4 bit binary number with permissible values of 0100 to 1100 (Hex '4' to 'C')
P	PIN digit	4 bit binary number with permissible values of 0000 to 1001 (Hex '0' to '9')
P/F	PIN/filler	Determined by PIN length
F	Filler	4 bit binary number with a value of 1111 (Hex 'F')

Table 25: Plaintext Offline PIN Block Format

P2 = '00' indicates that no particular qualifier is used. The processing of the VERIFY command in the ICC should know how to find the PIN data unambiguously.

6.5.12.3 Data Field Sent in the Command Message

If the Transaction PIN Data is sent in the VERIFY command, then the data field of the command message contains the value field of tag '99' (Transaction PIN Data).

If offline biometric verification data is sent in the VERIFY command, then the data field of the command message contains the value field of tag 'BF4E' (Biometric Verification Data Template).

6.5.12.4 Data Field Returned in the Response Message

No data field is returned in the response message.

6.5.12.5 Processing State Returned in the Response Message

'9000' indicates a successful execution of the command.

If command chaining is required, when the VERIFY command has CLA = '10' and the SW1 SW2 ≠ '9000', the terminal shall discontinue sending the remainder of the chained VERIFY commands.

When the VERIFY command has CLA = '10' and command chaining fails, the ICC shall return SW1 SW2 = '6800' to indicate command chaining failed (due to a failure other than the specific error conditions described in section 6.5.13).

For the VERIFY command with CLA = '00', when for the currently selected application the comparison between the Transaction PIN Data and the reference PIN data, or the comparison between the biometric template captured on the biometric capture device and contained in the command and the biometric reference template stored on the card, performed by the VERIFY command fails, the ICC shall return SW2 = 'Cx', where 'x' represents the number of retries still possible. When the card returns 'C0', no more retries are left, and the CVM shall be blocked. Any subsequent VERIFY command applied in the context of that application shall then fail with SW1 SW2 = '6983'.

6.5.13 Command Chaining

When more than one command is required to perform a function, multiple VERIFY or PIN CHANGE/UNBLOCK commands need to be sent to the card, in which case the consecutive command-response pairs can be chained. As defined in ISO/IEC 7816-4, bit 5 of the class byte CLA of the command is used to indicate command chaining, as shown in Table 26.

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
-	-	-	x	-	-	-	-	Command chaining control
-	-	-	0	-	-	-	-	– The command is the last or only command of a chain
-	-	-	1	-	-	-	-	– The command is not the last command of a chain

Table 26: Values of CLA in Command Chaining

As defined in ISO/IEC 7816-4, in response to a command that is not the last command of a chain, SW1 SW2 is set to '9000', meaning that the process has been completed so far. Moreover, the following specific error conditions may occur.

- If SW1 SW2 is set to '6883', then the last command of the chain was expected but was not received.
- If SW1 SW2 is set to '6884', then command chaining is not supported.

Part III

Debit and Credit Application Specification

7 Files for Financial Transaction Interchange

The description of the file structure and commands for accessing the files is found in Part III of Book 1 (for application selection) and Part II of this book (for the application elementary files). The definition of each of the data objects is defined in Annex A.

7.1 Mapping Data Objects

The payment system or issuer will map the appropriate data objects to files according to their needs, subject to the following restrictions:

- All files accessible using the READ RECORD command as defined in this specification containing data objects defined in this specification shall use SFIs in the range 1 to 10. These files:
  - Shall be linear files readable using the READ RECORD command as described in this specification.
  - May contain multiple records. Each record is limited to 254 bytes, including tag and length.
  - Each record shall be coded as a constructed data object. The tag of the constructed data object shall be '70' indicating a template proprietary to this specification, and the length field shall contain the total length of the encapsulated data objects.
  - Shall contain only data objects defined in this specification and coded in accordance with the BER-TLV described in Annex B.
  - May have access conditions to be satisfied for updates, but must be readable unconditionally.
- Files with SFIs in the range 11 to 20 are reserved for proprietary data to be specified by the individual payment systems.
- Files with SFIs in the range 21 to 30 are reserved for proprietary data to be specified by the issuer.

- The AFL determines the files and records to be used for processing a transaction. The use of the AFL is described in section 10.2. The data objects listed in Table 27 are used by the offline data authentication algorithm and, when present, should be located in the first record referenced by the AFL.[4]

Tag	Value
'8F'	Certification Authority Public Key Index
'90'	Issuer Public Key Certificate

Table 27: Data Objects Used by the Offline Data Authentication Algorithm

Additional information may be found in complementary payment system documentation.

7.2 Mandatory Data Objects

Table 28 lists the data objects that must be present in the ICC in files read using the READ RECORD command. All other data objects defined in this specification to be resident in such files in the card are optional.

Tag	Value	Presence
'5F24'	Application Expiration Date	M
'5A'	Application Primary Account Number (PAN)	M
'8C'	Card Risk Management Data Object List 1	M
'8D'	Card Risk Management Data Object List 2	M

Table 28: Mandatory Data Objects

Table 29 lists the data objects that must be present if the ICC supports SDA. Table 30 lists the data objects that must be present if the ICC supports DDA and/or CDA.[5] Table 31 lists the data objects that must be present if the ICC supports XDA. Offline data authentication is required to support offline transactions but is optional in cards that support only online transactions.

[4] This allows the terminal to optionally perform the hashing necessary for data authentication in parallel with reading and parsing of data for other purposes.

[5] The exception may be that the Issuer Public Key Remainder or the ICC Public Key Remainder could be absent. This is because if the public key modulus can be recovered in its entirety from the public key certificate there is no need for a remainder.

Tag	Value
'8F'	Certification Authority Public Key Index
'90'	Issuer Public Key Certificate
'93'	Signed Static Application Data
'92'	Issuer Public Key Remainder
'9F32'	Issuer Public Key Exponent

Table 29: Data Required for SDA

Tag	Value
'8F'	Certification Authority Public Key Index
'90'	Issuer Public Key Certificate
'92'	Issuer Public Key Remainder
'9F32'	Issuer Public Key Exponent
'9F46'	ICC Public Key Certificate
'9F47'	ICC Public Key Exponent
'9F48'	ICC Public Key Remainder
'9F49'	Dynamic Data Authentication Data Object List (DDOL) [6]

Table 30: Data Required for DDA and/or CDA

Tag	Value
'8F'	Certification Authority Public Key Index
'90'	Issuer Public Key Certificate
'9F46'	ICC Public Key Certificate

Table 31: Data Required for XDA

[6] In case the DDOL is not present in the card, the Default DDOL shall be used instead.

7.3 Data Retrievable by GET DATA Command

Data objects listed in Table 32 are not retrievable by the READ RECORD command but are retrieved by the terminal using the GET DATA command as described in this specification.

Of the objects listed here, only the Application Transaction Counter (ATC) is a mandatory data object, and it can be retrieved by either the GET DATA command or in the response to a GENERATE AC command. The terminal retrieves the ATC via the GET DATA command only if the ICC contains the Lower Consecutive Offline Limit (LCOL) and Upper Consecutive Offline Limit (UCOL) data objects. If the issuer does not wish terminal velocity checking to be performed and omits these data objects, the ICC does not need to support the GET DATA command, unless the card supports retrieval of the PIN Try Counter or the Log Format using GET DATA.

Tag	Value	Presence
'9F36'	Application Transaction Counter (ATC)	M
'9F17'	PIN Try Counter	O
'9F13'	Last Online ATC Register	O
'9F4F'	Log Format	O

Table 32: Data Objects Retrievable by GET DATA Command

7.4 Data Retrievable by GET PROCESSING OPTIONS

Data objects listed in Table 33 are not retrievable by the READ RECORD command but are retrieved by the terminal using the GET PROCESSING OPTIONS command as described in section 6.5.8. Table 33 defines the data returned, not the format of the response; section 6.5.8.4 describes the format of the data when returned by the GET PROCESSING OPTIONS command.

Tag	Value	Presence
'82'	Application Interchange Profile	M
'94'	Application File Locator	M

Table 33: Data Retrievable by GET PROCESSING OPTIONS

7.5 Erroneous or Missing Data in the ICC

Data objects in the card are classified in section 7.2 as either mandatory or optional. However, some optional data objects must be present to support optional functions selected by the issuer or must be present to avoid inconsistencies if other related data objects are present.

When any mandatory data object is missing, the terminal terminates the transaction unless otherwise specified in these specifications. When an optional data object that is required because of the existence of other data objects or that is required to support functions that must be performed due to the setting of bits in the Application Interchange Profile is missing, the terminal shall set the 'ICC data missing' indicator in the Terminal Verification Results (TVR) to 1.

Table 35 summarises the conditions under which this bit should be set to 1. If none of the conditions in Table 35 applies, the terminal shall set the bit to 0. The setting of this bit is in addition to any other actions specified in other sections of this Book.

Note: If data is missing during CVM processing, the CVM method fails. The 'ICC data missing' TVR bit is not set.

During a transaction the terminal shall ignore any data object coming from the ICC which is designated in the EMV data elements dictionary (Table 37 in Annex A) as terminal-sourced or issuer-sourced. The data elements dictionary defines data as being sourced from any of three places: the ICC, the terminal, or the issuer. If the terminal receives a primitive data element that is not within the templates as designated in the data dictionary, the terminal should consider that data as unknown and should ignore it.

The correct formatting of certain card-sourced data objects is not critical for completion of a transaction. If during normal processing the terminal recognises that any of the following data objects are incorrectly formatted, it shall ignore the formatting error and continue processing.

- Cardholder Name ('5F20')
- Cardholder Name Extended ('9F0B')
- Issuer Identification Number ('42')
- Issuer Identification Number Extended ('9F0C')
- Issuer URL ('5F50')
- Log Entry ('9F4D')
- Log Format ('9F4F')
- Track 1 Discretionary Data ('9F1F')

The format errors, if detected, may prevent a particular service to be delivered, but shall not prevent financial transactions to be processed and to complete successfully.

The following table lists proposed terminal behaviours when format errors are detected for these data elements. Other behaviours are allowed as long as the previous requirement is met.

Data	Proposed Terminal Behaviour
Cardholder Name ('5F20') Cardholder Name Extended ('9F0B')	• Continue processing as if the data is not present, or • Print/Display the printable characters, or • Print/Display error message in place of cardholder name
Issuer URL ('5F50')	• Continue processing as if the data is not present
Log Entry ('9F4D')	• Continue processing as if the data is not present, or • Continue processing as if no Transaction Log file is supported by the card, or • Abort Transaction Log file reading, or • Display error message when attempting to read the Transaction Log file
Log Format ('9F4F')	• Continue processing as if the data is not present, or • Continue processing as if no Transaction Log file is supported by the card, or • Abort Transaction Log file reading, or • Display error message when attempting to read the Transaction Log file

Table 34: Proposed Terminal Behaviours when Format Errors Detected for Selected Data Elements

It is up to the issuer to ensure that data in the card is of the correct format, and no format checking other than that specifically defined is mandated on the part of the terminal. However, if in the course of normal processing the terminal recognises that data is incorrectly formatted, the terminal shall terminate the transaction unless otherwise specified in these specifications. This rule includes (but is not limited to):

- Constructed data objects that do not parse correctly.
- Dates that are out of range (for example, months that are not in the range 1 to 12).
- Other data that must be in a specific range of values but are not.
- Multiple occurrences of a data object that should only appear once.
- An AFL with no entries.
- An AFL entry with invalid syntax, that is, any one or more of the following:
  - An SFI of 0 or 31.
  - A starting record number of 0.
  - An ending record number less than the starting record number (byte 3 < byte 2).

- Number of records participating in offline data authentication greater than the number of records (byte 4 > byte 3 – byte 2 + 1).

Name	Tag	'ICC Data Missing' Shall Be Set If ...
Application Transaction Counter (ATC)	'9F36'	ATC is not returned by GET DATA command and both Lower and Upper Consecutive Offline Limit data objects are present
Cardholder Verification Method (CVM) List	'8E'	Not present and AIP indicates that cardholder verification is supported
Certification Authority Public Key Index	'8F'	Not present and SDA, DDA, CDA or XDA selected
Issuer Public Key Certificate	'90'	Not present and SDA, DDA, CDA or XDA selected
Issuer Public Key Exponent	'9F32'	Not present and SDA, DDA or CDA selected.
Issuer Public Key Remainder	'92'	Not present and SDA, DDA or CDA selected and the 'length of the Issuer Public Key', as recovered from the Issuer Public Key Certificate, indicates that there was insufficient space for the entire Issuer Public Key in the certificate.
Last Online Application Transaction Counter (ATC) Register	'9F13'	Last Online ATC Register is not returned by GET DATA command and both Lower and Upper Consecutive Offline Limits are present
Signed Static Application Data	'93'	Not present and SDA selected
ICC Public Key Certificate	'9F46'	Not present and DDA, CDA or XDA selected
ICC Public Key Exponent	'9F47'	Not present and DDA or CDA selected
ICC Public Key Remainder	'9F48'	Not present and DDA or CDA selected and the 'length of the ICC Public Key', as recovered from the ICC Public Key Certificate, indicates that there was insufficient space for the entire ICC Public Key in the certificate

Table 35: ICC Data Missing Indicator Setting

8 Transaction Flow

The Application Interchange Profile specifies the application functions that are supported by the card. The terminal shall attempt to execute only those functions that the ICC supports. The functions shall be performed according to the Conditions of Execution as specified in section 10.

Book 1 describes all functionality outside the application layer, including the selection of the application. The functions described here begin after application selection has taken place.

The remainder of this book deals with the terminal-to-ICC dialogue on the level of the application logical functions. Section 8.2 describes a possible transaction flow.

8.1 Exception Handling

Exceptions to normal processing are described in this Book for specific status codes returned in the status bytes (SW1, SW2) or for missing data. Unless otherwise specified in these specifications, any SW1 SW2 returned by the transport layer to the application layer other than '9000', '63Cx', or '6283' shall cause termination of the transaction.[7] This requirement applies throughout these specifications except for the Application Selection process described in Book 1.

8.2 Example Flowchart

The flowchart in Figure 6 gives an example of a transaction flow that may be used by a terminal for a normal purchase transaction. This flowchart is only an example, and the order of processing may differ from that given here. All restrictions on the order of processing are provided in section 10.

[7] Other actions may be taken by prior agreement but are outside the scope of this specification.

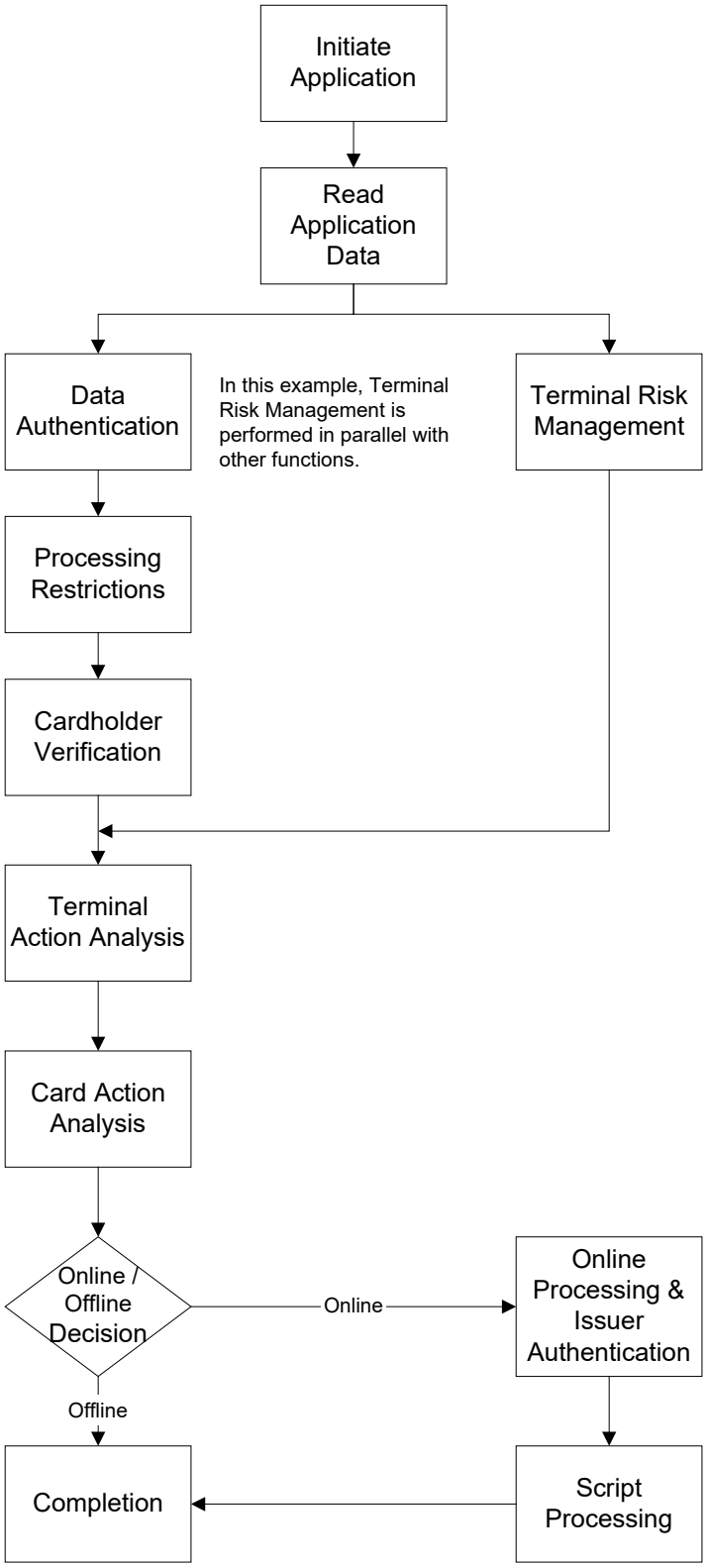


Figure 6: Transaction Flow Example

8.3 Additional Functions

Provision has been made in this specification for additional functions beyond those described here. Such additional functions may be:

- future additions to this specification
- proprietary functions implemented by local or national agreement or by the individual payment systems

The Application Interchange Profile indicates the functions supported in the ICC according to this specification. Most of the bits in this data object are reserved for future use (RFU). When a new function is added, a bit in the Application Interchange Profile will be allocated to indicate support for the new function, and this specification will be updated to specify the new function and where it fits into the transaction flow.

Proprietary functions may be added to the terminal and the ICC application as long as they do not interfere with processing of terminals and ICCs not implementing the function. For example, offline dynamic data authentication based on symmetric keys may be added at local option. Such proprietary functions, while not described in this specification, are not precluded, as long as the functions specified herein continue to be supported for all ICCs, including those not implementing the proprietary functions.

9 GENERATE AC Command Coding

The GENERATE AC command format and response codes are described fully in section 6.5.5. This section describes how the various options and data elements are used in transaction processing. Figure 7 depicts the interaction between the terminal decisions, ICC decisions, issuer approval, the GENERATE AC command, and the possible ICC responses.

The complete transaction flow is not shown in this chart, only the GENERATE AC commands, responses, and associated decisions. Furthermore, this chart applies to no ODA performed, SDA performed, and DDA performed. For CDA performed and XDA performed, please refer to Book 2 sections 6.6.3 and 12.5.4.

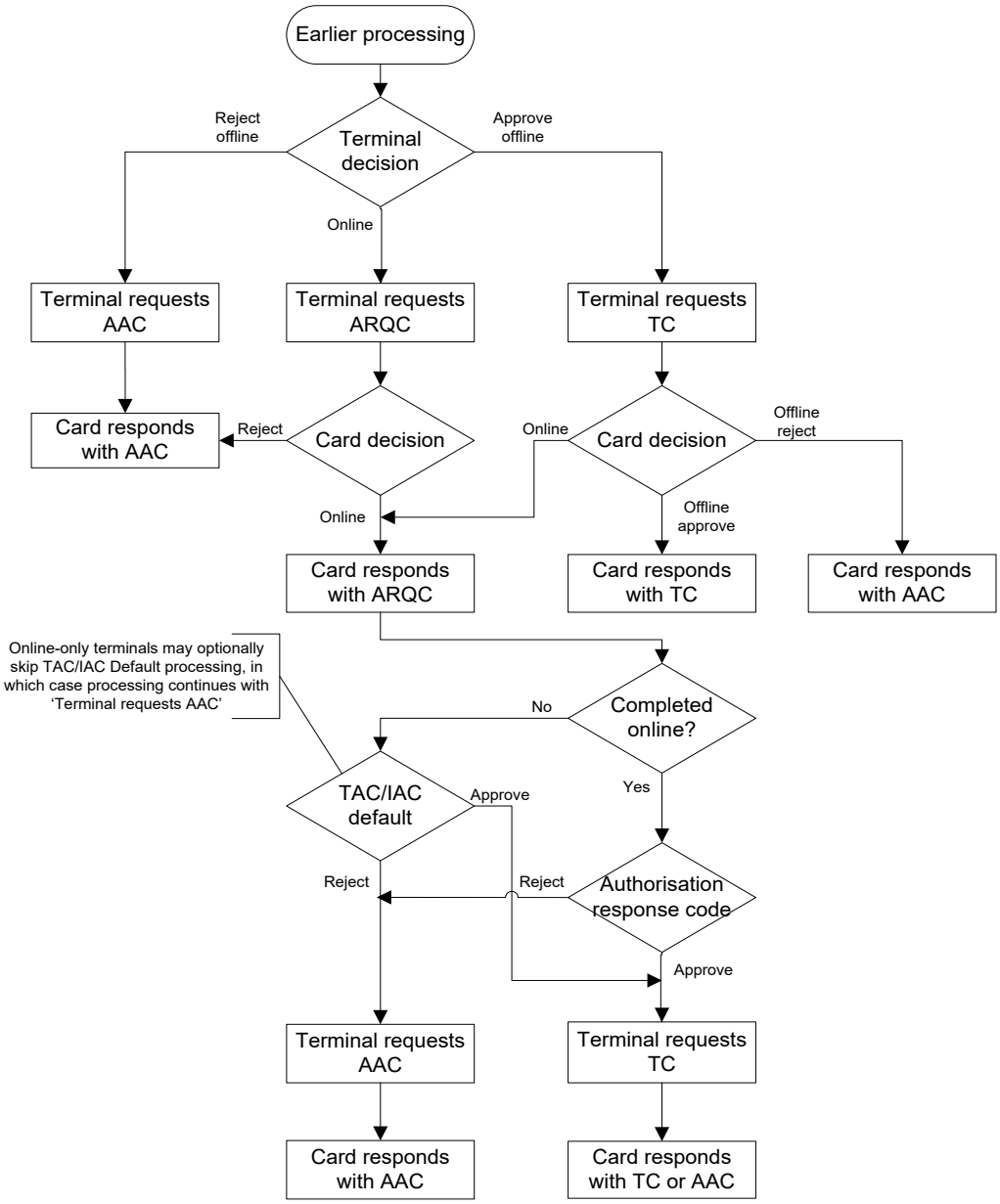


Figure 7: Use of GENERATE AC Options (no ODA, SDA, DDA)

9.1 Command Parameters

The GENERATE AC command parameters provide three different options to the terminal:

- Request for the generation of a TC
- Request for the generation of an ARQC
- Request for the generation of an AAC

9.2 Command Data

The data field of the GENERATE AC command is not TLV encoded, so it is imperative that the ICC knows the format of this data when the command is received. This is achieved by allowing the ICC to specify the format of the data to be included in the GENERATE AC command using a Card Risk Management Data Object List (CDOL).

9.2.1 Card Risk Management Data

The CDOL is a data object in the ICC that provides to the terminal a list of data objects that must be passed from the terminal to the ICC in the GENERATE AC command. There shall be two CDOLs in the ICC, referred to as CDOL1 (tag '8C') and CDOL2 (tag '8D'). CDOL1 is used with the first GENERATE AC command, and CDOL2 is used with the second GENERATE AC command (if used). The terminal uses the appropriate CDOL and the rules specified in section 5.4 to build the appropriate data field for the command. It is the responsibility of the terminal to ensure that current data is used in building the GENERATE AC command. To this end, the command data should be built immediately prior to issuing the GENERATE AC command.

9.2.2 Transaction Certificate Data

A CDOL may request a TC Hash Value to be included in the command data of a GENERATE AC command. The TDOL is a data object that provides to the terminal a list of data objects to be used in generating the TC Hash Value. The ICC may contain the TDOL, but there may be a default TDOL in the terminal, specified by the payment system, for use in case the TDOL is not present in the ICC. To create the hash value, the terminal shall use the TDOL (if present) or the default TDOL to create the appropriate value for input to the hash algorithm, applying the rules specified in section 5.4. If the default TDOL is used, the terminal shall set the 'Default TDOL used' bit in the TVR to 1. If a default TDOL is required but is not present in the terminal, a default TDOL with no data objects in the list shall be assumed. If this event occurs, since the default TDOL was not used, the terminal shall not set the 'Default TDOL used' bit in the TVR to 1.

If the terminal issues a second GENERATE AC command during the processing of a transaction, the terminal shall ensure that the data provided in the TC Hash Value is current at the time the command is issued.

9.3 Command Use

Either one or two GENERATE AC commands are issued during the processing of a transaction according to this specification.

The ICC shall respond to the first GENERATE AC command with any of the following:

- TC
- ARQC
- AAC

The ICC shall respond to a second GENERATE AC command with either a TC or an AAC.

The possible responses listed above are in hierarchical order, with a TC being the highest and an AAC being the lowest. The terminal may request a TC, an ARQC, or an AAC. If the ICC responds with a cryptogram at a higher level or with a cryptogram of an undefined type, this indicates a logic error in the ICC. If this occurs after the first GENERATE AC command in a transaction, the transaction shall be terminated. If it occurs after the second GENERATE AC command, all processing for the transaction has been completed, and the cryptogram returned shall be treated as an AAC.

If the ICC response is an approval (TC) or online authorisation request (ARQC) and the reference control parameter of the GENERATE AC command is 'CDA signature requested' (b5-b4 = 10), the ICC shall return a public key signature in the GENERATE AC response as specified in Book 2 section 6.6 (CDA).

If the reference control parameter of the GENERATE AC command is 'XDA signature requested' (b5-b4 = 01), the ICC shall return a public key signature in the GENERATE AC response as specified in Book 2 section 12 (XDA).

9.3.1 GENERATE AC (First Issuance)

The terminal completes its online/offline decision process with a GENERATE AC command (see section 6.5.5). The form of the command depends upon the decision made by the terminal:

- If the terminal decides the transaction might be completed offline, it requests a TC from the ICC. The ICC shall reply with a TC, an ARQC, or an AAC, depending upon its own analysis of the transaction.
- If the terminal decides the transaction should go online, it requests an ARQC from the ICC. The ICC shall reply with an ARQC, or an AAC.
- If the terminal decides to reject the transaction, it requests an AAC from the ICC. The ICC shall reply with an AAC.

If the ICC responds with a TC[8] or an AAC, the terminal completes the transaction offline.

If the ICC responds with an ARQC, the terminal attempts to go online, sending an authorisation request message to the issuer. Included in the authorisation request message is the ARQC for online card authentication.

9.3.2 GENERATE AC (Second Issuance)

Whether the terminal receives an authorisation response message as a result of online processing or an approval or rejection by using the Issuer Action Code – Default, it completes the transaction by requesting either a TC (in the case an approval was obtained) or an AAC (in case the issuer's instruction is to reject the transaction) from the ICC. If a TC was requested, the ICC shall reply with either a TC or an AAC. If an AAC was requested, the card shall reply with an AAC.

The ICC shall permit at most two GENERATE AC commands in a transaction. If the terminal issues more than two, the third and all succeeding GENERATE AC commands shall end with SW1 SW2 = '6985', and no cryptogram shall be returned.

[8] If ECC key recovery failed or XDA signature verification failed and depending on the TAC-Denial and/or IAC-Denial, the terminal attempts to go online after the Terminal Action Analysis with the TC returned by the ICC.

10 Functions Used in Transaction Processing

The following sections shall be read in conjunction with Book 4 Part II, which may contain additional terminal-specific requirements.

10.1 Initiate Application Processing

Purpose:

The Initiate Application Processing function:

- informs the ICC that the processing of a new transaction is beginning
- provides to the ICC the terminal-related information about the transaction
- obtains from the ICC the Application Interchange Profile and a list of files that contain the ICC data to be used in processing the transaction
- determines whether the transaction is allowed

Conditions of Execution:

The terminal shall always execute this function.

Sequence of Execution:

This is the first function performed after Application Selection.

Description:

The terminal shall set all bits in the Transaction Status Information (TSI) and the Terminal Verification Results (TVR) to 0.[9]

The PDOL is a list of tags and lengths of terminal-resident data elements needed by the ICC in processing the GET PROCESSING OPTIONS command. Only data elements having the terminal as the source of the data may be referenced in the PDOL.

If the PDOL does not exist, the GET PROCESSING OPTIONS command uses a command data field of '8300', indicating that the length of the value field in the command data is zero.

If the PDOL exists, the terminal extracts the PDOL from the FCI of the ADF and uses it to create a concatenated list of data elements without tags or lengths. The rules specified in section 5.4 apply to processing of the PDOL.

[9] There may be some exceptions in the timing for this. For example, these bits could be set to 0 at the completion of the previous transaction or prior to application selection of this transaction. The intent here is that the processing steps as described in the Application Specification presume the bits have been initialised to 0.

Note: An ICC supporting XDA or ECC ODE may request Terminal Capabilities (tag '9F33') using PDOL.

The terminal issues the GET PROCESSING OPTIONS command using either the command data field of '8300' (if there was no PDOL in the ICC) or a data object constructed with a tag of '83' and the appropriate length according to BER-TLV encoding rules and a value field that is the concatenated list of data elements resulting from processing the PDOL. The card returns either:

- The Application Interchange Profile, the Application File Locator (identifying the files and records containing the data to be used for the transaction), and status SW1 SW2 = '9000', or
- Status SW1 SW2 = '6985' ('Conditions of use not satisfied'), indicating that the transaction cannot be performed with this application.

The format of the response message is given in section 6.5.8.

If the status words '6985' are returned, the terminal shall eliminate the current application from consideration and return to the Application Selection function to select another application.

10.2 Read Application Data

Purpose:

Data contained in files in the ICC are required by the terminal to perform the various functions used in transaction processing as described in this section. The terminal must read this data from the ICC.

Conditions of Execution:

The terminal shall always execute this function.

Sequence of Execution:

The Read Application Data function is performed immediately following the Initiate Application Processing function.

Description:

The terminal shall read the files and records indicated in the AFL using the READ RECORD command identifying the file by its SFI. If an error prevents the terminal from reading data from the ICC, the transaction shall be terminated (see section 8.1).

The AFL is a list identifying the files and records to be used in the processing of a transaction. The terminal is to read only the records named in the AFL. Each element of the list corresponds to a file to be read and is structured as follows:

- The five most significant bits of the first byte indicate the SFI. The three least significant bits are RFU.
- The second byte indicates the first (or only) record number to be read for that SFI. The second byte shall never be set to zero.
- The third byte indicates the last record number to be read for that SFI. Its value is either greater than or equal to the second byte. When the third byte is greater than the second byte, all the records ranging from the record number in the second byte to and including the record number in the third byte shall be read for that SFI. When the third byte is equal to the second byte, only the record number coded in the second byte shall be read for that SFI.
- The fourth byte indicates the number of records involved in offline data authentication starting with the record number coded in the second byte. The fourth byte may range from zero to the value of the third byte less the value of the second byte plus 1.

The terminal shall process each entry in the AFL from left to right. A READ RECORD command as described in section 6.5.11 shall be issued for each record between the starting record number and the ending record number, inclusively. Any SW1 SW2 other than '9000' passed to the application layer as a result of reading any record shall cause the transaction to be terminated. Records specified in the AFL to be included in offline data authentication shall be processed as described in section 10.3.

The terminal shall store all recognised data objects read, whether mandatory or optional, for later use in the transaction processing. Data objects that are not recognised by the terminal (that is, their tags are unknown by the terminal) shall not be stored, but records containing such data objects may still participate in their entirety in offline data authentication, depending upon the coding of the AFL.[10]

All mandatory data objects shall be present in the card. If any mandatory data objects are not present, the terminal shall terminate the transaction.

Redundant primitive data objects are not permitted. If the terminal encounters more than one occurrence of a single primitive data object while reading data from the ICC, the transaction shall be terminated.

Proprietary data files may or may not conform to this specification. Records in proprietary files may be represented in the AFL and may participate in offline data authentication if they are readable without conditions by the READ RECORD command coded according to this specification. Otherwise, the reading and processing of proprietary files is beyond the scope of this specification.

[10] Payment system-specific tags are interpreted within the context of the application RID. Issuer-specific tags are interpreted within the context of the Issuer Identification Number. The Issuer Identification Number as defined in ISO/IEC 7812-1 may be present on the card in the IIN (tag '42') and/or IINE (tag '9F0C'). Additionally, to satisfy business requirements such as proprietary domestic processing, multiple issuers may agree on the definition of a private class tag. Such tags may be interpreted in the context of other data such as Issuer Country Code.

10.3 Offline Data Authentication

Purpose:

Offline data authentication is performed as specified in Book 2. This specification describes how it is determined whether offline data authentication will be performed, what kind of authentication will be performed, and how the success or failure of authentication affects the transaction flow and data recorded in the TVR and TSI.

Conditions of Execution:

Availability of data in the ICC to support offline data authentication is optional; its presence is indicated in the Application Interchange Profile (AIP). If the terminal and the ICC support a common method of offline data authentication, the terminal shall select this method and perform the offline data authentication. The terminal attempts to perform the highest priority offline data authentication method supported by both the card and terminal according to the following rules:

If both of the following are true, the terminal shall select the XDA method, set the TVR bit for 'XDA selected' to the value 1 and shall perform XDA as specified in this section, in Book 2 section 12, and in Book 4 section 6.3.2:

- The AIP indicates that the card supports XDA.
- Terminal Capabilities for the selected AID indicates that the terminal supports XDA.

Otherwise if both of the following are true then CDA is selected, the terminal shall select the CDA method and perform CDA as specified in this section, in Book 2 section 6, and in Book 4 section 6.3.2:

- The AIP indicates that the card supports CDA.
- The terminal supports CDA.

Otherwise if both of the following are true then DDA is selected, the terminal shall select the DDA method and perform DDA as specified in this section, in Book 2 section 6, and in Book 4 section 6.3.2:

- The AIP indicates that the card supports DDA.
- The terminal supports DDA.

Otherwise if both of the following are true, the terminal shall select the SDA method, set the TVR bit for 'SDA selected' to the value 1 and perform SDA as specified in this section, in Book 2 section 5, and in Book 4 section 6.3.2:

- The AIP indicates that the card supports SDA.
- The terminal supports SDA.

Otherwise (i.e. if neither XDA nor CDA nor DDA nor SDA is to be performed), the terminal shall set the 'Offline data authentication was not performed' bit in the TVR to 1.

Note: An ICC supporting both RSA based methods (SDA, DDA, CDA or RSA ODE) and ECC based methods (XDA or ECC ODE) is expected to return a PDOL indicating Terminal Capabilities (tag '9F33') in response to SELECT command. Based on the Terminal Capabilities, the ICC selects either RSA based methods or ECC based methods. Records referenced in the AFL contain either data objects needed for RSA based methods or data objects needed for ECC based methods, as selected by the ICC, but not both. In particular, if RSA methods are selected by the ICC, the certificates (tags '90' and '9F46') contained in the records referenced in the AFL are expected to be formatted as described in Book 2 Table 13 and Table 14, while if ECC methods are selected by the ICC, they are expected to be formatted as described in Book 2 Table 35 and Table 36.

Sequence of Execution:

For SDA and DDA the terminal shall perform offline data authentication in any order after Read Application Data but before completion of the terminal action analysis.

For CDA and XDA the terminal shall start offline data authentication at any time after Read Application Data, but CDA and XDA cannot be successfully completed until after the response to the GENERATE AC command. The key recovery/authentication process may be conducted at any time during this period, but the terminal shall check the presence of the payment system CA Public Key prior to Terminal Action Analysis.

For XDA the terminal shall complete the ECC key recovery and XDA signature verification processes described in Book 2 section 12 before any online authorisation request and before the final Terminal Action Analysis prior to any second GENERATE AC.

Description:

SDA authenticates static data put into the card by the issuer. DDA, CDA and XDA authenticate ICC-resident data, data from the terminal, and the card itself. CDA and XDA also authenticate that certain data passed between the terminal and card has not been altered by an intermediate device.

For SDA, DDA, and CDA, input to the authentication process is formed from the records identified by the AFL, followed by the data elements identified by the optional Static Data Authentication Tag List (tag '9F4A').

For XDA, input to the authentication process is formed from the records identified by the AFL, followed by the tag, length and value of the AIP, the tag, length and value of Application Identifier (AID) – terminal, and, if a PDOL was received from the card, the tag, length and value of the PDOL. This forms the Issuer Certified Card Data (which is the Static Data to be Authenticated for XDA).

Only those records identified in the AFL as participating in offline data authentication are to be processed. Records are processed in the same sequence in which they appear within AFL entries. The records identified by a single AFL entry are to be processed in record number sequence. The first record begins the input for the authentication process, and each succeeding record is concatenated at the end of the previous record.

The records read for offline data authentication shall be TLV-coded with tag equal to '70'.

The data from each record to be included in the offline data authentication input depends upon the SFI of the file from which the record was read.

- For files with SFI in the range 1 to 10, the record tag ('70') and the record length are excluded from the offline data authentication process. All other data in the data field of the response to the READ RECORD command (excluding SW1 SW2) is included.
- For files with SFI in the range 11 to 30, the record tag ('70') and the record length are not excluded from the offline data authentication process. Thus all data in the data field of the response to the READ RECORD command (excluding SW1 SW2) is included.

If the records read for offline data authentication are not TLV-coded with tag equal to '70' then offline data authentication shall be considered to have been performed and to have failed; that is, the terminal shall set the 'Offline data authentication was performed' bit in the TSI to 1, and shall set the appropriate 'SDA failed' or 'DDA failed' or 'CDA failed' or 'ECC key recovery failed'[11] bit in the TVR.

When CDA is performed at first GENERATE AC (CDA Mode 1 or 2), terminal shall set 'Offline data authentication was performed' bit in the TSI to 1 immediately after the first GENERATE AC.

The bytes of the record are included in the concatenation in the order in which they appear in the command response.

After all records identified by the AFL have been processed, the terminal processes the following:

- For SDA, DDA, and CDA, the Static Data Authentication Tag List is processed, if it exists. If the Static Data Authentication Tag List exists, it shall contain only the tag for the Application Interchange Profile. The tag must represent the AIP available in the current application. The value field of the AIP is to be concatenated to the current end of the input string. The tag and length of the AIP are not included in the concatenation.
- For XDA, the tag, length and value field of the AIP as received from the card are concatenated to the end of the concatenated records (whether the SDA Tag List exists or not), followed by the tag, length and value of Application Identifier (AID) – terminal. If a PDOL was received, the tag, length and the value of the PDOL as received from the card are appended to the end of the string. If no PDOL was received then nothing is appended.

[11] This bit is not set until after the first GENERATE AC command and is not set if CA ECC key is missing. See Book 4 section 6.3.2.2.2.

Building of the input list for offline data authentication is considered the first step in the offline data authentication process. If the input cannot be built because of a violation of one of the above rules but offline data authentication should be performed according to the 'Conditions of Execution' above, offline data authentication shall be considered to have been performed and to have failed; that is, the terminal shall set the 'Offline data authentication was performed' bit in the TSI to 1 and shall set the appropriate 'SDA failed' or 'DDA failed' or 'CDA failed' or 'ECC key recovery failed'[12] bit in the TVR.

See Book 2 for additional steps to be performed for offline data authentication.

If the CA ECC Public Key referred by the CA Public Key Index is not present in the terminal or the index is missing, the bit 'CA ECC key missing' in the TVR must be set before the final TAA prior to the first GENERATE AC command.

If XDA is selected but the terminal fails to authenticate and recover the Issuer Public Key or the ICC Public Key (including verification of the Issuer Certified Card Data) and the CA ECC key is not missing, the bit 'ECC key recovery failed' in the TVR shall be set but only after issuing the first GENERATE AC command and before the TAA that is performed after processing the first GENERATE AC response. (Refer to the Note in Book 4 section 6.3.2.2.2 for further information regarding this).

If SDA is performed but is unsuccessful, the 'SDA failed' bit in the TVR shall be set to 1; otherwise it shall be set to 0.

If DDA is performed but is unsuccessful, the 'DDA failed' bit in the TVR shall be set to 1; otherwise it shall be set to 0.

If CDA is performed but is unsuccessful, the 'CDA failed' bit in the TVR shall be set to 1; otherwise it shall be set to 0.

If XDA is selected, the 'XDA selected' bit in the TVR shall be set to 1 before the final TAA prior to the first GENERATE AC command.

If XDA is selected but is unsuccessful, then one (and only one) of the 'CA ECC key missing', 'ECC key recovery failed', or 'XDA signature verification failed' bits in the TVR shall be set to 1; otherwise they shall be set to 0.

If XDA is selected then ECC key recovery shall have ended and XDA signature processing shall have ended before the TAA that is performed after processing the first GENERATE AC response. Note that ECC key recovery may have ended before or after the first GENERATE AC either by failing or succeeding, or by aborting due to CA ECC key missing. The bit 'XDA signature verification failed' in the TVR shall be set only if CA ECC key retrieval and ECC key recovery were successful but XDA signature verification failed.

Upon completion of the offline data authentication function, the terminal shall set the 'Offline data authentication was performed' bit in the TSI to 1.

[12] This bit is not set until after the first GENERATE AC command and is not set if CA ECC key is missing. See Book 4 section 6.3.2.2.2.

10.4 Processing Restrictions

Purpose:

The purpose of the Processing Restrictions function is to determine the degree of compatibility of the application in the terminal with the application in the ICC and to make any necessary adjustments, including possible rejection of the transaction.

Conditions of Execution:

The terminal shall always execute this function.

Sequence of Execution:

Functions described here may be performed at any time after Read Application Data and prior to completion of the terminal action analysis.

Description:

The Processing Restrictions function comprises the following compatibility checks:

- Application Version Number
- Application Usage Control
- Application Effective/Expiration Dates Checking

10.4.1 Application Version Number

The application within both the terminal and the ICC shall maintain an Application Version Number assigned by the payment system. The terminal shall use the version number in the ICC to ensure compatibility. If the Application Version Number is not present in the ICC, the terminal shall presume the terminal and ICC application versions are compatible, and transaction processing shall continue. If the Application Version Number is present in the ICC, it shall be compared to the Application Version Number maintained in the terminal. If they are different, the terminal shall set the 'ICC and terminal have different application versions' bit in the TVR to 1.

10.4.2 Application Usage Control

The Application Usage Control indicates restrictions limiting the application geographically or to certain types of transactions. If this data object is present, the terminal shall make the following checks:

- If the transaction is being conducted at an ATM, the 'Valid at ATMs' bit must be on in Application Usage Control.
- If the transaction is not being conducted at an ATM, the 'Valid at terminals other than ATMs' bit must be on in Application Usage Control.

If the Application Usage Control and Issuer Country Code are both present in the ICC, the terminal shall make the checks described in Table 36.

If:	and if Issuer Country Code:	then the following bit must be set to 1 in Application Usage Control:
Transaction Type indicates cash transaction	matches Terminal Country Code	'Valid for domestic cash transactions'
	does not match Terminal Country Code	'Valid for international cash transactions'
Transaction Type indicates purchase (of goods/services)	matches Terminal Country Code	'Valid for domestic goods' and/or 'Valid for domestic services'
	does not match Terminal Country Code	'Valid for international goods' and/or 'Valid for international services'
transaction has a cashback amount	matches Terminal Country Code	'Domestic cashback allowed'
	does not match Terminal Country Code	'International cashback allowed'

Table 36: Terminal Action Regarding Application Usage Control

If any of the above tests fail, the terminal shall set the 'Requested service not allowed for card product' bit in the TVR to 1.

10.4.3 Application Effective/Expiration Dates Checking

If the Application Effective Date is present in the ICC, the terminal shall check that the current date is greater than or equal to the Application Effective Date. If it is not, the terminal shall set the 'Application not yet effective' bit in the TVR to 1.

The terminal shall check that the current date is less than or equal to the Application Expiration Date. If it is not, the terminal shall set the 'Expired application' bit in the TVR to 1.

10.5 Cardholder Verification

Purpose:

Cardholder verification is performed to ensure that the person presenting the ICC is the person to whom the application in the card was issued.

Conditions of Execution:

Ability of the ICC to support at least one cardholder verification method is indicated in the Application Interchange Profile, as shown in section C1. If this bit is set to 1, the terminal shall use the cardholder verification related data in the ICC to determine whether one of the issuer-specified cardholder verification methods (CVMs) shall be executed. This process is described below.

Sequence of Execution:

This function may be performed any time after Read Application Data and before completion of the terminal action analysis.

Description:

The CVM List (tag '8E') is a composite data object consisting of the following:

1. An amount field (4 bytes, binary format), referred to as 'X' in Table 44: CVM Condition Codes. 'X' is expressed in the Application Currency Code with implicit decimal point. For example, 123 (hexadecimal '7B') represents £1.23 when the currency code is '826'.
2. A second amount field (4 bytes, binary format), referred to as 'Y' in Table 44. 'Y' is expressed in Application Currency Code with implicit decimal point. For example, 123 (hexadecimal '7B') represents £1.23 when the currency code is '826'.
3. A variable-length list of two-byte data elements called Cardholder Verification Rules (CV Rules). Each CV Rule describes a CVM and the conditions under which that CVM should be applied (see section C3).

If the CVM List is not present in the ICC, the terminal shall terminate cardholder verification without setting the 'Cardholder verification was performed' bit in the TSI.

Note: A CVM List with no Cardholder Verification Rules is considered to be the same as a CVM List not being present.

If the CVM List is present in the ICC, the terminal shall process each rule in the order in which it appears in the list according to the following specifications. Cardholder verification is completed when any one CVM is successfully performed or when the list is exhausted.

If the terminal encounters formatting errors in the CVM List such as a list with an odd number of bytes (that is, with an incomplete CVM Rule), the terminal shall terminate the transaction as specified in section 7.5.

If any of the following is true:

- the conditions expressed in the second byte of a CV Rule are not satisfied, or
- data required by the condition (for example, the Application Currency Code or Amount, Authorised) is not present, or
- the CVM Condition Code is outside the range of codes understood by the terminal (which might occur if the terminal application program is at a different version level than the ICC application),

then the terminal shall bypass the rule and proceed to the next. If there are no more CV Rules in the list, cardholder verification has not been successful, and the terminal shall set the 'Cardholder verification was not successful' bit in the TVR to 1.

If the conditions expressed in the second byte of the CV Rule are satisfied, the terminal next checks whether it recognises the CVM coded in the first byte of the CV Rule and proceeds according to the following steps:

1. If the CVM is recognised, the terminal next checks to determine whether it supports the CVM.
   - If the CVM is supported, the terminal shall attempt to perform it.
     - If the CVM is performed successfully, cardholder verification is complete and successful. If the CVM just processed was 'Fail CVM Processing', the terminal shall set the 'Cardholder verification was not successful' bit in the TVR (b8 of byte 3) to 1 and no further CVMs shall be processed regardless of the setting of b7 of byte 1 in the first byte of the CV Rule.[13]
     - If the CVM is not performed successfully, processing continues at step 2.
   - If the CVM is not supported, processing continues at step 2. In addition, the terminal shall perform the following:
     - Set the 'PIN entry required and PIN pad not present or not working' bit (b5 of byte 3) of the TVR to 1 for the following cases:
       - The CVM was online PIN and online PIN was not supported
       - The CVM included any form of offline PIN, and neither form of offline PIN was supported.
     - Set the 'A selected Biometric Type not supported' bit (b2 of byte 4) of the TVR to 1 for the following case:
       - The CVM included any Biometric Type, and no common Biometric Type was supported.

   If the CVM is not recognised, the terminal shall set the 'Unrecognised CVM' bit in the TVR (b7 of byte 3) to 1 and processing continues at step 2.

[13] If a card CVM List contains an entry for 'Fail CVM Processing,' it is strongly recommended that byte 1 bit 7 be set to 0 for this entry.

2. If cardholder verification was not completed in step 1 (that is, the CVM was not recognised, was not supported, or failed), the terminal examines b7 of byte 1 of the CV Rule.
   - If b7 is set to 1, processing continues with the next CV Rule, if one is present.
   - If b7 is set to 0, or there are no more CV Rules in the list, cardholder verification is complete and unsuccessful. The terminal shall set the 'Cardholder verification was not successful' bit in the TVR (b8 of byte 3) to 1.

When cardholder verification is completed, the terminal shall:

- set the CVM Results according to Book 4 section 6.3.4.5
- set the 'Cardholder verification was performed' bit in the TSI to 1.

Note: If an ICC supports a CVM Code using either RSA ODE or ECC ODE, but does not support both the RSA and the ECC mode for this CVM Code, it is expected to return a PDOL indicating Terminal Capabilities (tag '9F33') in response to SELECT command. Based on the Terminal Capabilities, the ICC selects either RSA based methods or ECC based methods. Records referenced in the AFL contain a CVM List that reflects the ICC capabilities for the transaction. For instance, if the ICC supports Enciphered PIN verification performed by ICC with ECC but not with RSA, the CVM List in the records is expected to contain the Enciphered PIN verification performed by ICC CVM Code only if the ICC has selected ECC based methods for the transaction.

10.5.1 Offline PIN Processing

This section applies to the verification by the ICC of a plaintext or enciphered PIN presented by the terminal.

If an offline PIN is the selected CVM as determined by the above process, offline PIN processing may not be successfully performed for any one of the following reasons:

- The terminal does not support offline PIN.[14] In this case, the terminal shall set the 'PIN entry required and PIN pad not present or not working' bit in the TVR to 1.
- The terminal supports offline PIN, but the PIN pad is malfunctioning. In this case, the terminal shall set the 'PIN entry required and PIN pad not present or not working' bit in the TVR to 1.
- The terminal bypassed PIN entry at the direction of either the merchant or the cardholder.[15] In this case, the terminal shall set the 'PIN entry required, PIN pad present, but PIN was not entered' bit in the TVR to 1. The terminal shall consider this CVM unsuccessful and shall continue cardholder verification processing in accordance with the card's CVM List.
- The PIN is blocked upon initial use of the VERIFY command or if recovery of the enciphered PIN Block has failed (the ICC returns SW1 SW2 = '6983' or '6984' in response to the VERIFY command). In this case, the terminal shall set the 'PIN Try Limit exceeded' bit in the TVR to 1.
- The number of remaining PIN tries is reduced to zero (indicated by an SW1 SW2 of '63C0' in the response to the VERIFY command). In this case, the terminal shall set the 'PIN Try Limit exceeded' bit in the TVR to 1.

The only case in which offline PIN processing is considered successful is when the ICC returns an SW1 SW2 of '9000' in response to the VERIFY command.

[14] This means that the terminal does not support either offline plaintext PIN verification or offline enciphered PIN verification. If the terminal supports at least one of these functions, it is considered to support offline PIN for the purposes of setting the TVR bits.

[15] Especially for a new cardholder or during conversion to PINs, it is likely that a cardholder will realise that he or she does not know the PIN. In this case, it is better to bypass PIN processing with an indication to the issuer of the circumstances than it is to either terminate the transaction or try numbers until the PIN try count is exhausted. If the transaction goes online, the issuer can decide whether to accept the transaction without the PIN.

10.5.2 Online PIN Processing

If online PIN processing is a required CVM as determined by the above process, the processing may not be successfully performed for any one of the following reasons:

- The terminal does not support online PIN. In this case, the terminal shall set the 'PIN entry required and PIN pad not present or not working' bit in the TVR to 1.
- The terminal supports online PIN, but the PIN pad is malfunctioning. In this case, the terminal shall set the 'PIN entry required and PIN pad not present or not working' bit in the TVR to 1.
- The terminal bypassed PIN entry at the direction of either the merchant or the cardholder. In this case, the terminal shall set the 'PIN entry required, PIN pad present, but PIN was not entered' bit in the TVR to 1.

If the online PIN is successfully entered, the terminal shall set the 'Online CVM captured' bit in the TVR to 1. In this case, cardholder verification is considered successful and complete.

10.5.3 Signature Processing

If a signature is a required CVM as determined by the above process, the terminal shall determine success based upon the terminal's capability to support the signature process (see complementary payment systems documentation for additional information). If the terminal is able to support signature, the process is considered successful, and cardholder verification is complete.

10.5.4 Combination CVMs

Some CVMs require multiple verification methods (for example, offline PIN plus signature). For these CVMs, all methods in the CVM must be successful for cardholder verification to be considered successful.

10.5.5 CVM Processing Logic

The flows on the following pages illustrate the CVM processing logic but do not contain all the details of the execution of the selected CVM.

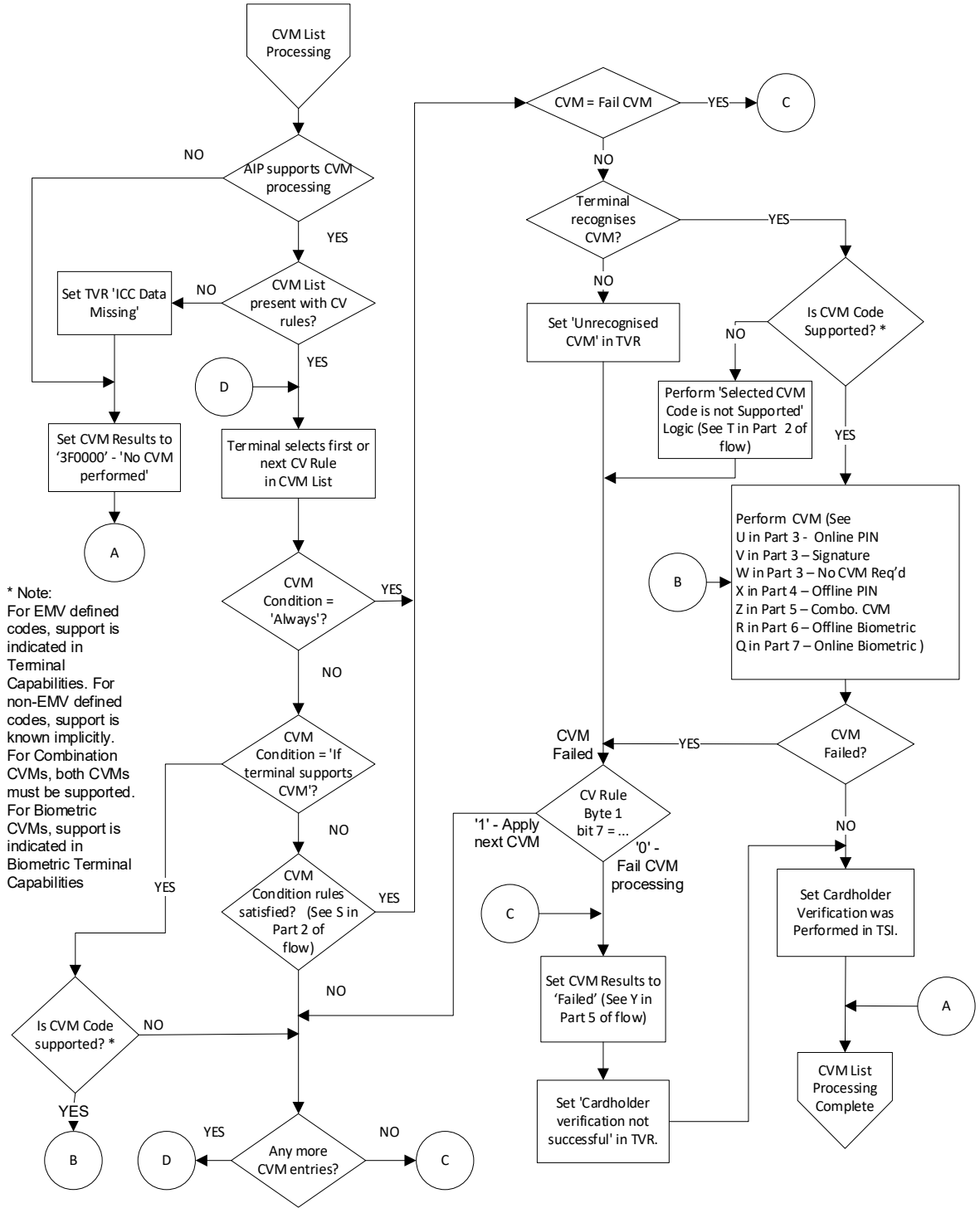


Figure 8: CVM Processing (Part 1 of 7)

Note: When a biometric card is presented to a terminal without biometric support, the Biometric CVM Code is either not recognised or recognised but not supported by the terminal. As described in section 10.5, the terminal will process the next CVM if CVM Code byte 1 b7 is set to 1, or will complete cardholder verification, set CVM Results to 'Failed' and set 'Cardholder verification not successful' bit in the TVR, if CVM Code byte 1 b7 is set to 0.

A terminal without biometric support may choose whether to recognise or not recognise the Biometric CVM Codes. If the terminal without biometric support chooses to recognise the Biometric CVM Codes, then the terminal shall implement the biometric cardholder verification processing in section 10.5 and shall treat biometric data objects as recognised data objects.

When a non-biometric card is presented to a terminal with biometric support, the terminal will not select a Biometric CVM Code, as described in section 10.5.

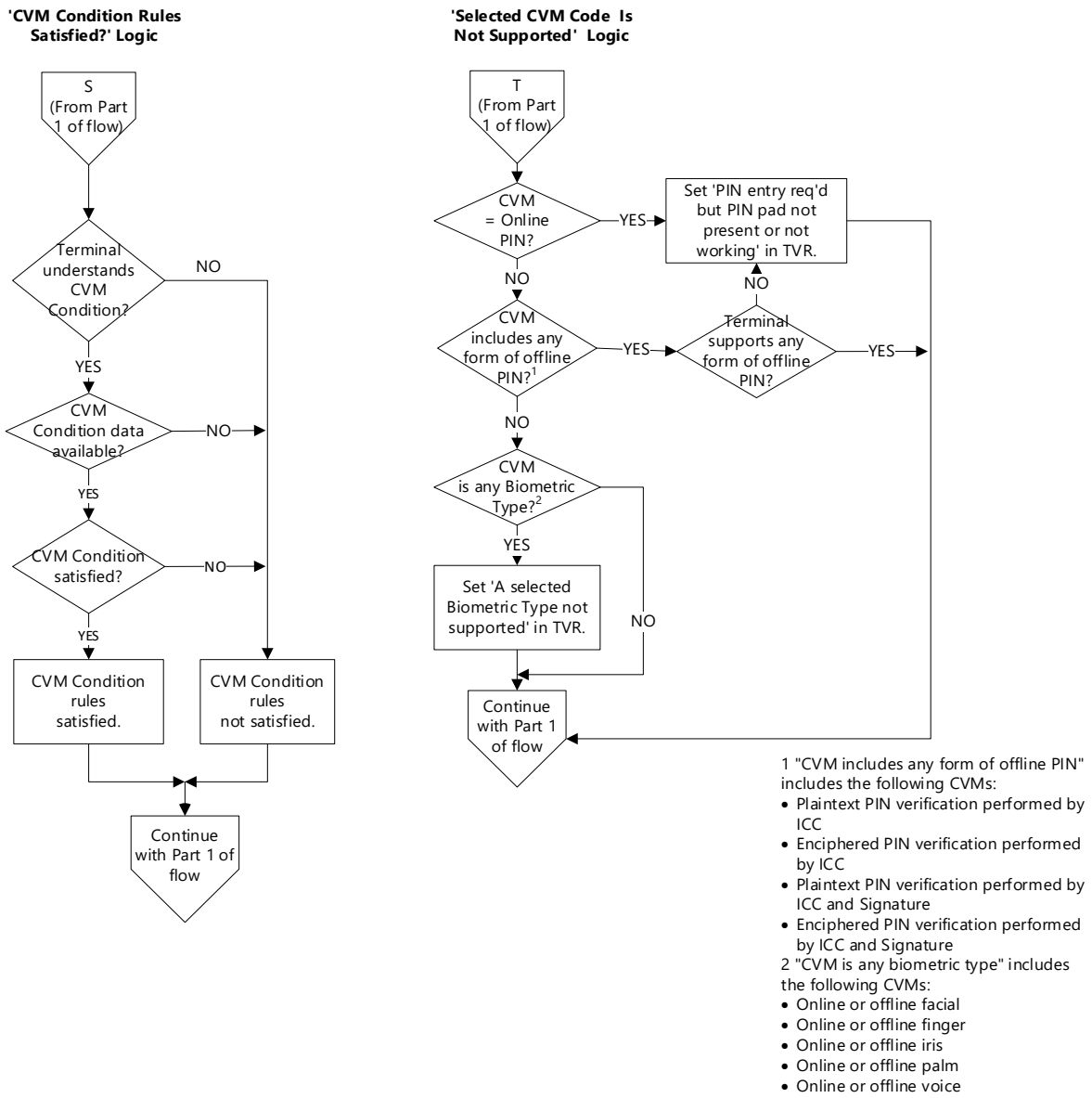


Figure 9: CVM Processing (Part 2 of 7)

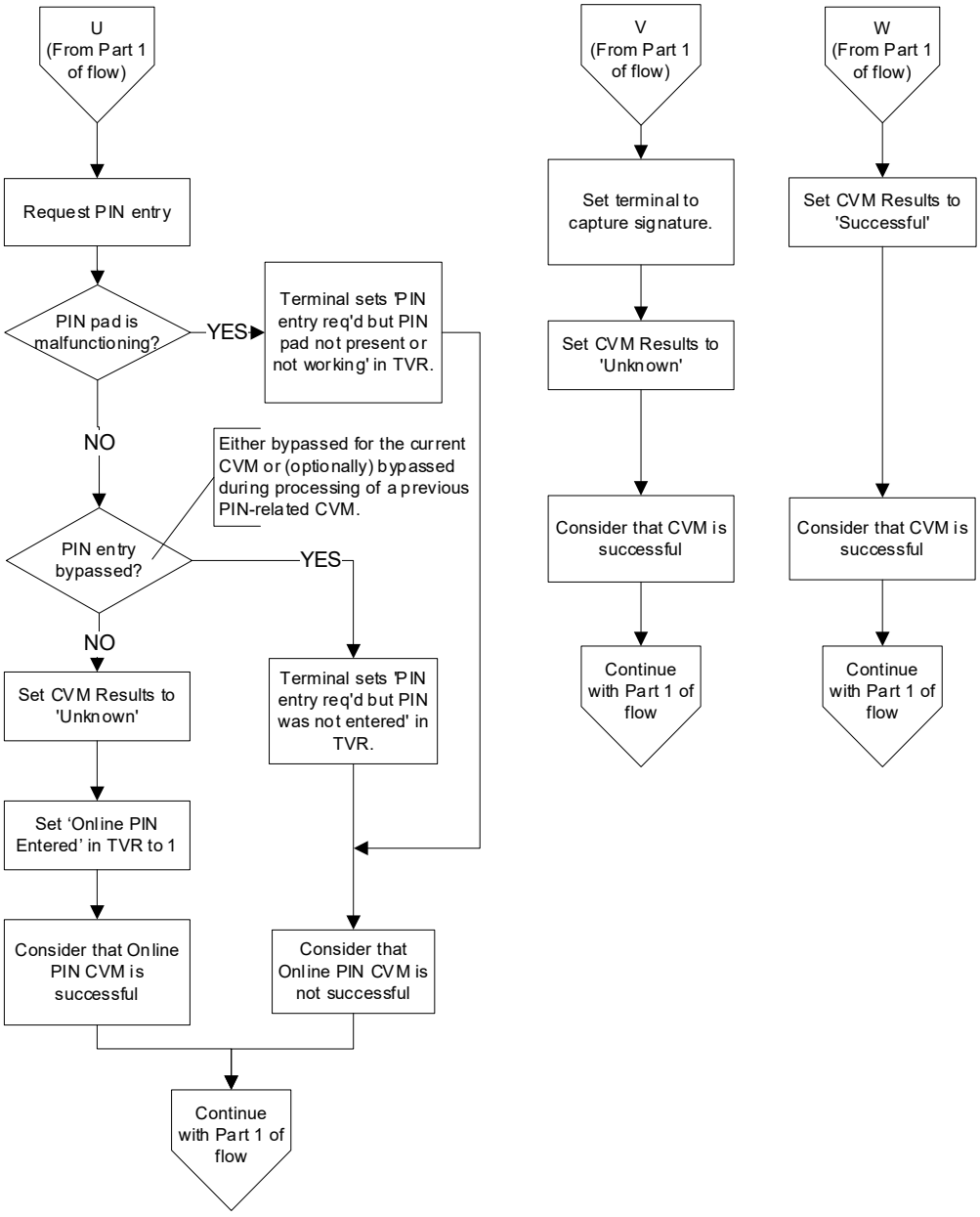


Figure 10: CVM Processing (Part 3 of 7)

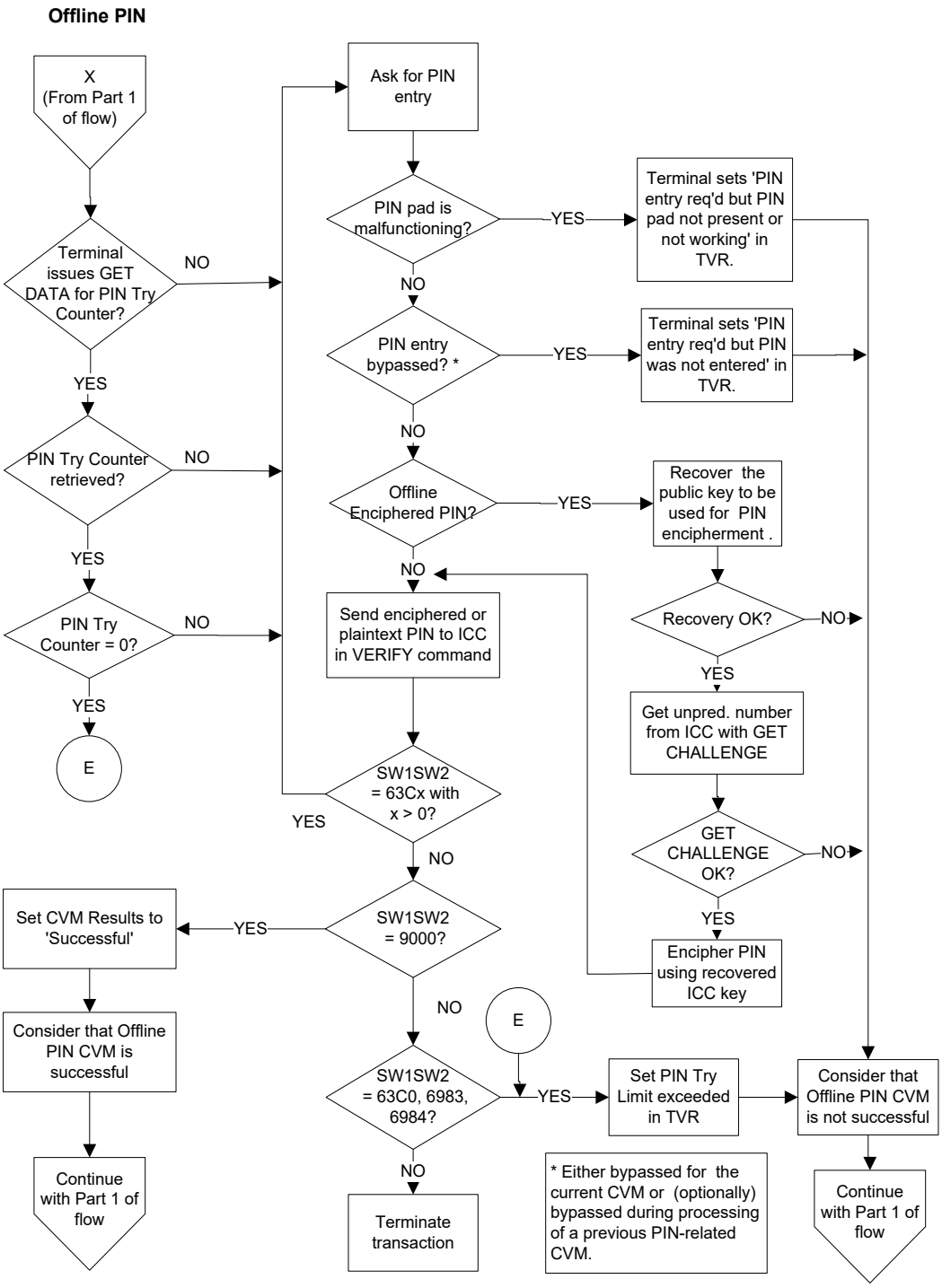


Figure 11: CVM Processing (Part 4 of 7)

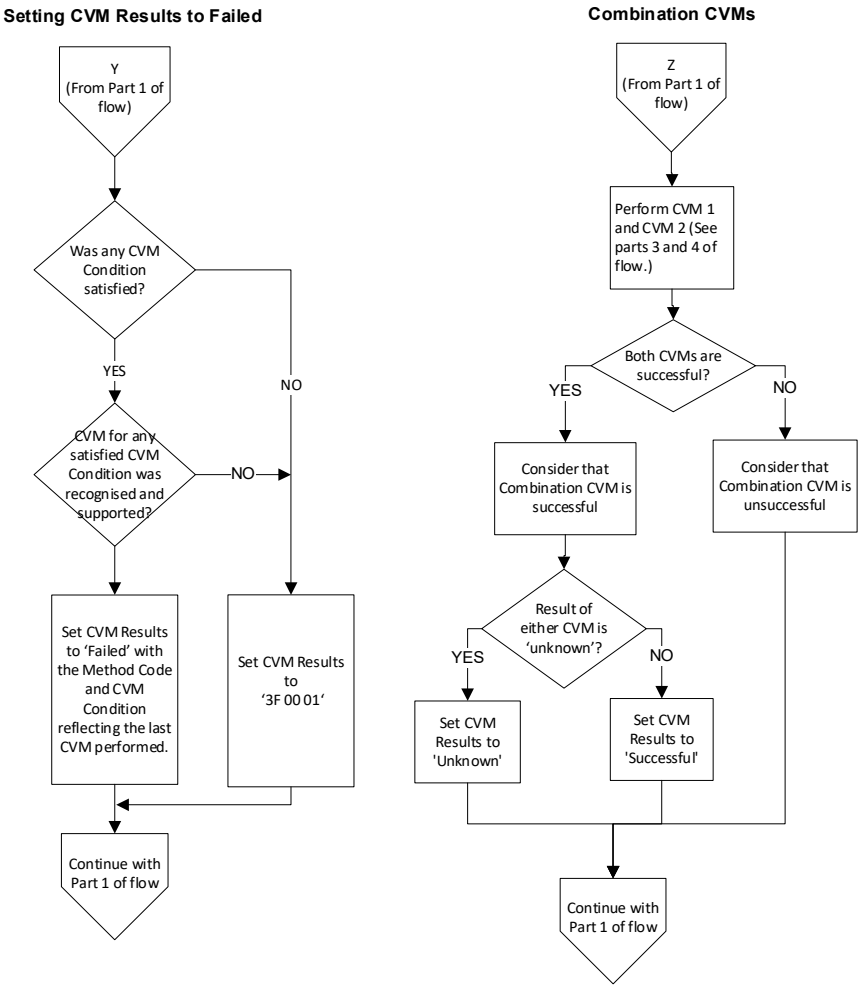
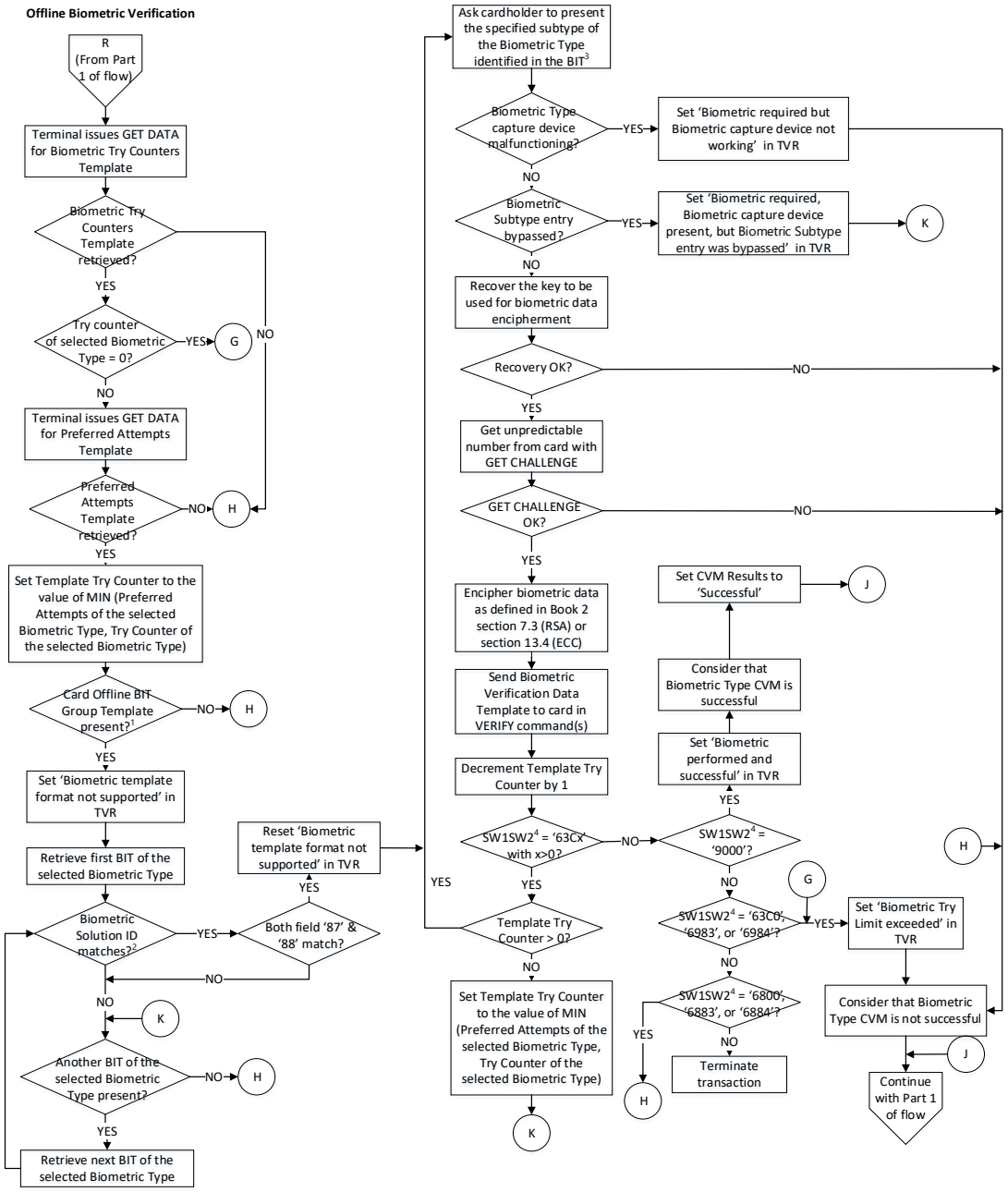


Figure 12: CVM Processing (Part 5 of 7)



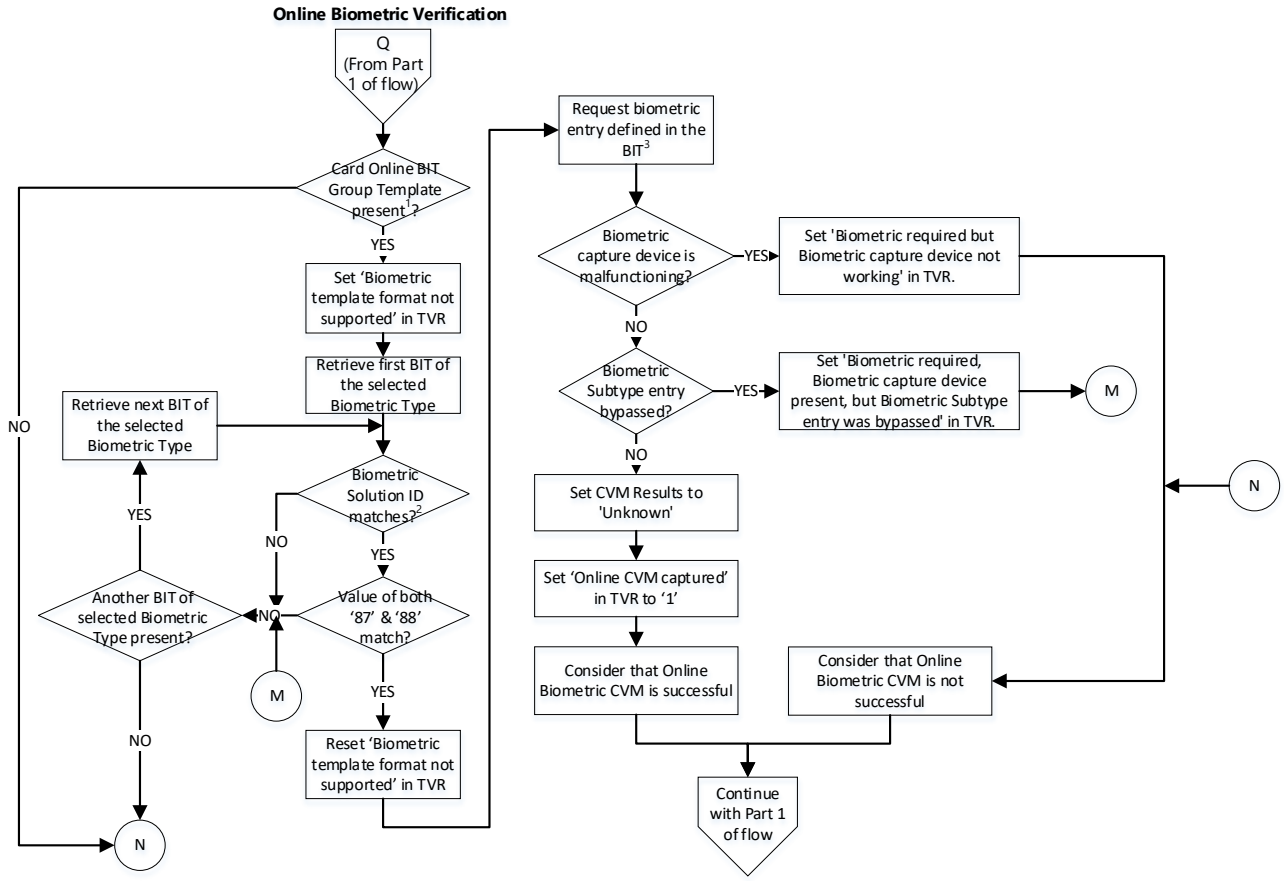
1 The Offline BIT Group Template shall be present and contain at least one BIT.

2 The terminal shall determine if the Biometric Solution ID of the BIT being processed equals to the Biometric Solution ID of any BIT in Terminal BIT Group Template.

3 It is assumed that the biometric template is captured after the cardholder is advised on which Biometric Subtype is required, and the BDB is obtained from the Biometric Processing Application.

4 The SW1 SW2 shall be in response to the final VERIFY command.

Figure 13: CVM Processing (Part 6 of 7)



[1] The Online BIT Group Template shall present and contain at least one BIT.

[2] The terminal shall determine if the Biometric Solution ID of the BIT being processed equals to the Biometric Solution ID of any BIT in Terminal BIT Group Template.

[3] It is assumed that the biometric template is captured after the cardholder is advised on which Biometric Subtype is required, and the BDB is obtained from the Biometric Processing Application.

Figure 14: CVM Processing (Part 7 of 7)

10.5.6 Offline Biometric Verification Processing

This section applies to the verification by the ICC of an enciphered biometric template presented by the terminal. The offline biometric verification methods include offline facial, finger, iris, palm, and voice.

If any of the offline biometric verification methods is the selected CVM, offline biometric verification processing may not be successfully performed for any of the following reasons:

- The terminal does not support the selected offline Biometric CVM. In this case, the terminal shall set the 'A selected Biometric Type not supported' bit in the TVR to 1.
- The terminal supports the selected offline Biometric CVM, but the biometric capture device is malfunctioning. In this case, the terminal shall set the 'Biometric required but Biometric capture device not working' bit in the TVR to 1.
- The terminal bypassed the selected offline Biometric Subtype entry at the direction of either the merchant or the cardholder. In this case, the terminal shall set the 'Biometric required, Biometric capture device present, but Biometric Subtype entry was bypassed' bit in the TVR to 1.
- The selected offline Biometric CVM is blocked upon initial use of the VERIFY command or if recovery of the Enciphered Biometric Data has failed (the ICC returns SW1 SW2 = '6983' or '6984' in response to the final VERIFY command). In this case, the terminal shall set the 'Biometric Try Limit exceeded' bit in the TVR to 1.
- The number of remaining tries of the selected offline Biometric CVM is reduced to zero (indicated by an SW1 SW2 of '63C0' in the response to the final VERIFY command). In this case, the terminal shall set the 'Biometric Try Limit exceeded' bit in the TVR to 1.
- The terminal does not support the format specified in any of the Biometric Information Templates (BITs), as specified in section C7, retrieved from the card for the selected Biometric Type. In this case, the terminal shall set 'Biometric template format not supported' bit in the TVR to 1.

When the VERIFY command is chained, the SW1 SW2 in response to the final VERIFY command indicates the outcome of biometric verification processing. If SW1 SW2 = '6800', '6883', or '6884', the terminal shall consider this CVM unsuccessful and shall continue cardholder verification processing in accordance with the card's CVM List.

The only case in which the selected offline biometric verification processing is considered successful is when the ICC returns an SW1 SW2 of '9000' in response to a single or all of multiple VERIFY command(s). In this case, the terminal shall set 'Biometric performed and successful' bit in the TVR to 1.

As shown in Figure 13, the terminal shall determine if tag '87' and '88' (format owner and format type) of the BIT in the Card BIT Group Template, as specified in section C8, match tag '87' and '88' of any BIT in the Terminal BIT Group Template, as specified in section C8, using the following procedures:

- If the BIT in the Card BIT Group Template does not contain level 2 Biometric Header Template (BHT) 1 and BHT 2, as shown in Table 48, and if the values of tag '87' and '88' of the BIT in the Card BIT Group Template are equal to the values of tag '87' and '88' (respectively) of any BIT in the Terminal BIT Group Template, the terminal shall determine that there is a match.
- If the BIT in the Card BIT Group Template contains level 2 BHT 1 and BHT 2, as shown in Table 48, and if the values of tag '87' and '88' in the BHT 1 of the BIT in the Card BIT Group Template are equal to the values of tag '87' and '88' (respectively) of any BIT in the Terminal BIT Group Template, the terminal shall determine that there is a match.
- Otherwise, the terminal shall determine that there is no match.

Moreover, the terminal shall use the sequence of BITs in the Card BIT Group Template and the Preferred Attempts Template (tag 'BF4D') to determine which Biometric Subtype the terminal shall ask the user to present for verification.

As an example, if the Preferred Finger Attempts is configured as 3 and the Card BIT Group Template is configured as following:

- BIT 1 – right index finger
- BIT 2 – right middle finger

Then, if Finger is selected by the terminal as the CVM, the terminal shall ask the user to present the right index finger first since it is first in the list. If the verification of the right index finger fails after 3 tries and the Finger Try Counter is not zero, the terminal shall move to the next finger on the list and ask the user to present the right middle finger up to 3 tries. The process continues until either the ICC returns SW1 SW2 = '9000' or the Finger Try Counter is zero.

10.5.7 Online Biometric Verification Processing

If online biometric verification is a required CVM as determined by the process described in Figure 8, the processing may not be successfully performed for any one of the following reasons:

- The terminal does not support the selected online Biometric CVM. In this case, the terminal shall set the 'A selected Biometric Type not supported' bit in the TVR to 1.
- The terminal supports the selected online Biometric CVM, but the biometric capture device is malfunctioning. In this case, the terminal shall set the 'Biometric required but Biometric capture device not working' bit in the TVR to 1.
- The terminal bypassed the selected online Biometric Subtype entry at the direction of either the merchant or the cardholder. In this case, the terminal shall set the 'Biometric required, Biometric capture device present, but Biometric Subtype entry was bypassed' bit in the TVR to 1.
- The terminal does not support the format specified in any of the BITs retrieved from the card for the selected Biometric Type. In this case, the terminal shall set the 'Biometric template format not supported' bit in the TVR to 1.

If the online biometric verification is successfully captured, the terminal shall set the 'Online CVM captured' bit in the TVR to 1. In this case, cardholder verification is considered successful and complete.

10.6 Terminal Risk Management

Purpose:

Terminal risk management is that portion of risk management performed by the terminal to protect the acquirer, issuer, and system from fraud. It provides positive issuer authorisation for high-value transactions and ensures that transactions initiated from ICCs go online periodically to protect against threats that might be undetectable in an offline environment. The result of terminal risk management is the setting of appropriate bits in the TVR.

Conditions of Execution:

Terminal risk management shall always be performed regardless of the setting of the 'Terminal risk management is to be performed' bit in the Application Interchange Profile. The following exceptions from Book 4 apply:

- An offline-only terminal and an online-only terminal need not support random transaction selection.
- An online-only terminal need not support any of the terminal risk management functions.
- A cardholder-controlled terminal (Terminal Type = '3x') need not support terminal risk management.

Sequence of Execution:

Terminal risk management may be performed at any time after Read Application Data but before issuing the first GENERATE AC command.

Description:

Terminal risk management consists of:

- Floor limit checking
- Random transaction selection
- Velocity checking

Upon completion of terminal risk management, the terminal shall set the 'Terminal risk management was performed' bit in the TSI to 1.

10.6.1 Floor Limits

To prevent split sales, the terminal may have a transaction log of approved transactions stored in the terminal consisting of at least the Application PAN and transaction amount and possibly the Application PAN Sequence Number and Transaction Date. The number of transactions to be stored and maintenance of the log are outside the scope of this specification, although to prevent split sales the number of transactions stored may be quite small.

During terminal risk management floor limit checking, the terminal checks the transaction log (if available) to determine if there is a log entry with the same Application PAN, and, optionally, the same Application PAN Sequence Number. If there are several log entries with the same PAN, the terminal selects the most recent entry. The terminal adds the Amount, Authorised for the current transaction to the amount stored in the log for that PAN to determine if the sum exceeds the Terminal Floor Limit. If the sum is greater than or equal to the Terminal Floor Limit, the terminal shall set the 'Transaction exceeds floor limit' bit in the TVR to 1.

If the terminal does not have a transaction log available or if there is no log entry with the same PAN, the Amount, Authorised is compared to the appropriate floor limit. If the amount authorised is equal to or greater than the floor limit, the terminal sets the 'Transaction exceeds floor limit' bit to 1 in the TVR.

10.6.2 Random Transaction Selection

For each application the relevant payment system specifies, in addition to the floor limit:

- 'Target Percentage to be Used for Random Selection' (in the range of 0 to 99)
- Threshold Value for Biased Random Selection (which must be zero or a positive number less than the floor limit)
- 'Maximum Target Percentage to be Used for Biased Random Selection' (also in the range of 0 to 99 but at least as high as the previous 'Target Percentage to be Used for Random Selection'). This is the desired percentage of transactions 'just below' the floor limit that will be selected by this algorithm.

Any transaction with a transaction amount less than the Threshold Value for Biased Random Selection will be subject to selection at random without further regard for the value of the transaction. The terminal shall generate a random number in the range of 1 to 99. If this random number is less than or equal to the 'Target Percentage to be used for Random Selection', the transaction shall be selected.

Any transaction with a transaction amount equal to or greater than the Threshold Value for Biased Random Selection but less than the floor limit will be subject to selection with bias toward sending higher value transactions online more frequently (biased random selection). For these transactions, the terminal shall compare its generated random number against a Transaction Target Percent, which is a linear interpolation of the target percentages provided by the payment system ('Target Percentage to be used for Random Selection' and 'Maximum Target Percentage to be used for Biased Random Selection').[16] If the random number is less than or equal to the Transaction Target Percent, the transaction shall be selected.

Figure 15 illustrates the probability of selection as a function of the transaction amount:

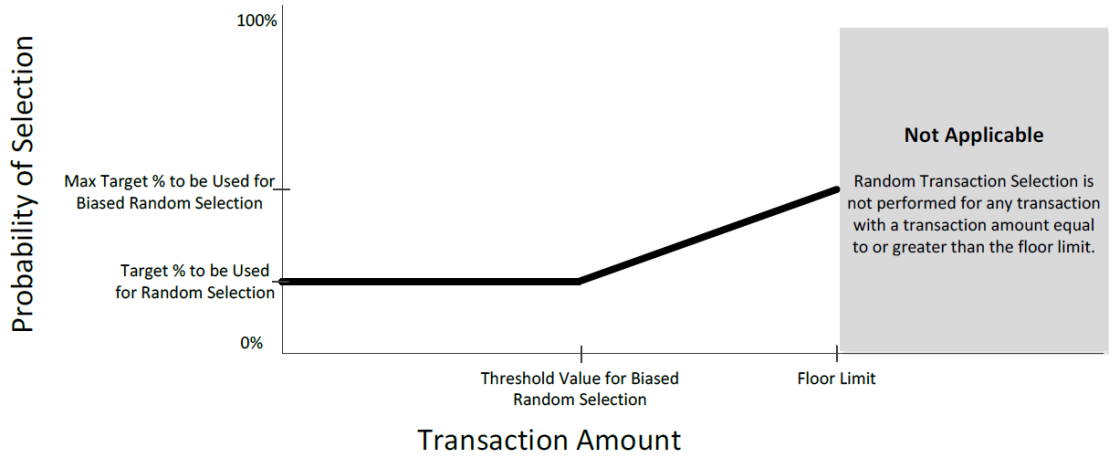


Figure 15: Random Transaction Selection Example

If the transaction is selected through the process described in this section, the terminal shall set the 'Transaction selected randomly for online processing' bit in the TVR to 1.

[16] The Transaction Target Percent is calculated as follows:

$$\text{Interpolation factor} = \frac{\text{Amount, Authorised} - \text{Threshold Value}}{\text{Floor Limit} - \text{Threshold Value}}$$

$$\text{Transaction Target Percent} = \left(\left(\text{Maximum Target Percent} - \text{Target Percent}\right) \times \text{Interpolation factor}\right) + \text{Target Percent}$$

10.6.3 Velocity Checking

If both the Lower Consecutive Offline Limit (tag '9F14') and Upper Consecutive Offline Limit (tag '9F23') exist, the terminal shall perform velocity checking as described in this section.[17] If either of these data objects is not present in the ICC application, the terminal shall skip this section.

The ATC and Last Online ATC Register shall be read from the ICC using GET DATA commands. If either of the required data objects is not returned by the ICC in response to the GET DATA command, or if the value of the ATC is less than or equal to the value in the Last Online ATC Register, the terminal shall:

- Set both the 'Lower consecutive offline limit exceeded' and the 'Upper consecutive offline limit exceeded' bits in the TVR to 1.
- Not set the 'New card' indicator in the TVR unless the Last Online ATC Register is returned and equals zero.
- End velocity checking for this transaction.

If the required data objects are available, the terminal shall compare the difference between the ATC and the Last Online ATC Register with the Lower Consecutive Offline Limit to see if the limit has been exceeded. If the difference is equal to the Lower Consecutive Offline Limit, this means that the limit has not yet been exceeded. If the limit has been exceeded, the terminal shall set the 'Lower consecutive offline limit exceeded' bit in the TVR to 1 and also compare the difference with the Upper Consecutive Offline Limit to see if the upper limit has been exceeded. If it has, the terminal shall set the 'Upper consecutive offline limit exceeded' bit in the TVR to 1.

The terminal shall also check the Last Online ATC Register for a zero value. If it is zero, the terminal shall set the 'New card' bit in the TVR to 1.

[17] The purpose of velocity checking is to allow an issuer to request that, after a certain number of consecutive offline transactions (the Lower Consecutive Offline Limit), transactions should be completed online. However, if the terminal is incapable of going online, transactions may still be completed offline until a second (Upper Consecutive Offline Limit) limit is reached. After the upper limit is reached, the recommendation of the issuer might be to reject any transaction that cannot be completed online. Once a transaction has been completed online with successful issuer authentication, the count begins anew, so that transactions may be processed offline until the lower limit is again reached.

10.7 Terminal Action Analysis

Purpose:

Once terminal risk management and application functions related to a normal offline transaction have been completed, the terminal makes the first decision as to whether the transaction should be approved offline, declined offline, or transmitted online.

- If the outcome of this decision process is to proceed offline, the terminal issues a GENERATE AC command to ask the ICC to return a TC.
- If the outcome of the decision is to go online, the terminal issues a GENERATE AC command to ask the ICC for an Authorisation Request Cryptogram (ARQC).
- If the decision is to reject the transaction, the terminal issues a GENERATE AC to ask for an Application Authentication Cryptogram (AAC).

An offline decision made here is not final. If the terminal asks for a TC from the ICC, the ICC, as a result of card risk management, may return an ARQC or AAC.

Conditions of Execution:

The terminal action analysis function is always performed.

Sequence of Execution:

The terminal action analysis function is performed after terminal risk management and cardholder and/or merchant transaction data entry has been completed. It shall be performed prior to the first use of the GENERATE AC command.

The Issuer Action Code – Default and Terminal Action Code – Default processing described below shall also be performed after online processing is attempted in the case where the terminal was unable to process the transaction online.

The terminal action analysis function may be executed at several places during a transaction to eliminate the need for unnecessary processing. If any processing results in the setting of a bit in the TVR (for example, failure of cardholder verification), it may be desirable to perform this function immediately to determine whether the transaction should be rejected offline based upon the issuer's parameters in the ICC or the acquirer's parameters in the terminal. Recognition of such a decision early in processing may allow the terminal to avoid prolonging a transaction that will ultimately be rejected. Multiple execution of this decision process is optional on the part of the terminal.

Description:

The terminal shall make a preliminary decision to reject the transaction, complete it online, or complete it offline based upon the TVR, issuer action preferences, and acquirer action preferences according to the method described in this section.

The ICC contains (optionally) three data elements to reflect the issuer's selected action to be taken based upon the content of the TVR. Each of the three data elements has defaults specified here in case any of these data elements are absent from the ICC. The three data elements are:

- Issuer Action Code – Denial
- Issuer Action Code – Online
- Issuer Action Code – Default

Collectively, these three data objects are termed the Issuer Action Codes. The purpose of each is described in this section. The format of each is identical and mirrors the TVR. Each has one bit corresponding to each bit in the TVR, and the Issuer Action Code (IAC) bit specifies an action to be taken if the corresponding bit in the TVR is set to 1. Thus, the size and format of each of the Issuer Action Codes is identical to the TVR.

Similarly, the terminal may contain three data elements to reflect the acquirer's selected action to be taken based upon the content of the TVR. These data elements are:

- Terminal Action Code – Denial
- Terminal Action Code – Online
- Terminal Action Code – Default

Collectively, these three data objects are termed the Terminal Action Codes. The purpose of each is described in this section. The format of each is identical and mirrors the TVR. Each has one bit corresponding to each bit in the TVR, and the Terminal Action Code (TAC) bit specifies an action to be taken if the corresponding bit in the TVR is set to 1. Thus, the size and format of each of the Terminal Action Codes is identical to the TVR and to the Issuer Action Codes.

The existence of each of the Terminal Action Codes is optional. In the absence of any Terminal Action Code, a default value consisting of all bits set to 0 is to be used in its place. However, it is strongly recommended that as a minimum, the Terminal Action Code – Online and Terminal Action Code – Default should be included with the bits corresponding to 'Offline data authentication was not performed', 'SDA failed', 'DDA failed', 'CDA failed', 'XDA signature verification failed', 'ECC key recovery failed', and 'CA ECC key missing' set to 1.[18]

[18] This protects against a fraudulent card with all the bits in the Issuer Action Code set to 0. Without this protection, such a card could be created with no possibility of going online or declining transactions. All transactions would be approved offline.

Processing of the action codes is done in pairs, that is, the Issuer Action Code – Denial is processed together with the Terminal Action Code – Denial, the Issuer Action Code – Online is processed together with the Terminal Action Code – Online, and the Issuer Action Code – Default is processed together with the Terminal Action Code – Default. Processing of the action codes shall be performed in the order specified here.

If the Issuer Action Code – Denial does not exist, a default value with all bits set to 0 is to be used. Together, the Issuer Action Code – Denial and the Terminal Action Code – Denial specify the conditions that cause denial of a transaction without attempting to go online. If either data object exists, the terminal shall inspect each bit in the TVR. For each bit in the TVR that has a value of 1, the terminal shall check the corresponding bits in the Issuer Action Code – Denial and the Terminal Action Code – Denial. If the corresponding bit in either of the action codes is set to 1, it indicates that the issuer or the acquirer wishes the transaction to be rejected offline. In this case, the terminal shall issue a GENERATE AC command to request an AAC from the ICC. This AAC may be presented to the issuer to prove card presence during this transaction, but details of handling a rejected transaction are outside the scope of this specification.

If the Issuer Action Code – Online is not present, a default value with all bits set to 1 shall be used in its place. Together, the Issuer Action Code – Online and the Terminal Action Code – Online specify the conditions that cause a transaction to be completed online. These data objects are meaningful only for terminals capable of online processing. Offline-only terminals may skip this test and proceed to checking the Issuer Action Code – Default and Terminal Action Code – Default, described below. For an online-only terminal, if it has not already decided to reject the transaction as described above, it shall continue transaction processing online, and shall issue a GENERATE AC command requesting an ARQC from the card. For a terminal capable of online processing, if the terminal has not already decided to reject the transaction as described above, the terminal shall inspect each bit in the TVR. For each bit in the TVR that has a value of 1, the terminal shall check the corresponding bits in both the Issuer Action Code – Online and the Terminal Action Code – Online. If the bit in either of the action codes is set to 1, the terminal shall complete transaction processing online and shall issue a GENERATE AC command requesting an ARQC from the ICC. Otherwise, the terminal shall issue a GENERATE AC command requesting a TC from the ICC.

If the Issuer Action Code – Default does not exist, a default value with all bits set to 1 shall be used in its place. Together, the Issuer Action Code – Default and the Terminal Action Code – Default specify the conditions that cause the transaction to be rejected if it might have been approved online but the terminal is for any reason unable to process the transaction online. The Issuer Action Code – Default and the Terminal Action Code – Default are used only if the Issuer Action Code – Online and the Terminal Action Code – Online were not used (for example, in case of an offline-only terminal) or indicated a desire on the part of the issuer or the acquirer to process the transaction online but the terminal was unable to go online. In the event that an online-only terminal was unable to successfully go online, it may optionally skip TAC/IAC Default processing (shown in Figure 7 for a transaction that was not completed online)[19]. If an online-only terminal does skip TAC/IAC Default processing, it shall request an AAC with the second GENERATE AC command. If the terminal has not already rejected the transaction and the terminal is for any reason unable to process the transaction online, the terminal shall use this code to determine whether to approve or reject the transaction offline. If any bit in Issuer Action Code – Default or the Terminal Action Code – Default and the corresponding bit in the TVR are both set to 1, the transaction shall be rejected and the terminal shall request an AAC to complete processing. If no such condition appears, the transaction may be approved offline, and a GENERATE AC command shall be issued to the ICC requesting a TC.

Offline only terminals may be implemented in different ways but shall always check, at least, (1) the Terminal Action Code – Denial and Issuer Action Code – Denial and (2) the Terminal Action Code – Default and Issuer Action Code – Default.

When processing the first GENERATE AC command, for each bit in the TVR that has a value of 1, the offline-only terminal shall check the corresponding bits in both the Issuer Action Code – Denial and the Terminal Action Code – Denial. If the bit in either of the action codes is set to 1, the terminal shall request an AAC from the ICC.

If an AAC is not requested based on the Issuer Action Code – Denial and the Terminal Action Code – Denial processing above, the offline-only terminal continues processing according to one of the options below:

[19] Note that if an online-only terminal is unable to successfully go online and TAC/IAC Default processing is optionally performed, this could result in a TC being requested with the second GENERATE AC command (depending upon the TAC/IAC Default settings).

(1) The offline-only terminal uses the Terminal Action Code – Online and Issuer Action Code – Online. For each bit in the TVR that has a value of 1, the terminal shall check the corresponding bits in both the Issuer Action Code – Online and the Terminal Action Code – Online. If the corresponding bit in either of the action codes is set to 1, the terminal shall request an ARQC from the ICC in the first GENERATE AC command. Otherwise, if ARQC is not requested based on the IAC-Online and TAC-Online processing above, the terminal shall use the Action Code – Default. For each bit in the TVR that has a value of 1, the terminal shall check the corresponding bits in both the Issuer Action Code – Default and the Terminal Action Code – Default. If the bit in either of the action codes is set to 1, the terminal shall request an AAC from the ICC in the first GENERATE AC command. If the ICC returns an ARQC in the first GENERATE AC response then the terminal shall use the Action Code – Default. For each bit in the TVR that has a value of 1, the terminal shall check the corresponding bits in both the Issuer Action Code – Default and the Terminal Action Code – Default. If the bit in either of the action codes is set to 1, the terminal shall request an AAC from the ICC in the second GENERATE AC command.

(2) The offline-only terminal skips the Issuer Action Code – Online and Terminal Action Code – Online check, and shall check the Issuer Action Code – Default and the Terminal Action Code – Default. For each bit in the TVR that has a value of 1, the terminal shall check the corresponding bits in both the Issuer Action Code – Default and the Terminal Action Code – Default. If the corresponding bit in either of the action codes is set to 1, the terminal shall request an AAC from the ICC.

If CDA is to be performed (as described in section 10.3 of this book and Book 2 section 6.6), the terminal shall set b5-b4 for 'CDA signature requested' in the GENERATE AC command to 10.

If XDA is to be performed (as described in section 10.3 of this book and Book 2 section 12), the terminal shall set b5-b4 for 'XDA signature requested' in the GENERATE AC command to 01.

10.8 Card Action Analysis

Purpose:

An ICC may perform its own risk management to protect the issuer from fraud or excessive credit risk. Details of card risk management algorithms within the ICC are specific to the issuer and are outside the scope of this specification, but as a result of the risk management process, an ICC may decide to complete a transaction online or offline or reject the transaction. The ICC may also decide that an advice message should be sent to the issuer to inform the issuer of an exceptional condition.

Conditions of Execution:

The card online/offline decision is specified by its response to the GENERATE AC command. Therefore, this section applies to all transactions. Whether the ICC performs any risk management tests is transparent to the terminal and outside the scope of this specification.

Sequence of Execution:

The card action analysis process is performed when the terminal issues the GENERATE AC command for a given transaction.

Description:

The result of risk management performed by the ICC is a decision for one of the following actions to be taken by the terminal:

- Approve the transaction offline. This option is available to the ICC only if the terminal has made a preliminary decision to complete the transaction offline, as described in section 10.7.
- Complete the transaction online.
- Reject the transaction.

The decision by the ICC is made known to the terminal by returning a TC, an ARQC, or an AAC to the terminal in response to a GENERATE AC command, as described in section 6.5.5.

Upon the completion of the card action analysis function, the terminal shall set the 'Card risk management was performed' bit in the TSI to 1.

10.8.1 Terminal Messages for an AAC

An AAC returned by the card indicates either a rejection of the specific transaction or a restriction that disallows use of the card in the environment of the transaction (for example, the card application may be restricted only to specific merchant categories). In both cases, the card disapproves the transaction, but the terminal may choose to display different messages in the two cases. The card may optionally distinguish the cases by the use of the code returned in the Cryptogram Information Data (see the GENERATE AC command in section 6.5.5). If an AAC is returned with b3-b1 = 001 in the Cryptogram Information Data, the AAC was returned due to card restrictions.

10.8.2 Advice Messages

The issuer may wish for an advice message, separate from either an authorisation request or a clearing message, to be sent in certain exception cases. (Currently, the only identified such case is 'PIN Try Limit exceeded', but allowance has been made for the addition of other cases later; see Table 15).

If b4 of the Cryptogram Information Data is 1, the terminal shall process the transaction as shown in Book 4 sections 6.3.7 and 12.2.5. Further information may be found in complementary payment system documentation.

10.9 Online Processing

Purpose:

Online processing is performed to ensure that the issuer can review and authorise or reject transactions that are outside acceptable limits of risk defined by the issuer, the payment system, or the acquirer.

Conditions of Execution:

Online processing shall be performed if the ICC returns an ARQC (or a TC[20] when XDA has failed) in response to the first GENERATE AC command for the transaction.

Sequence of Execution:

The online processing function is performed when the terminal receives an ARQC in response to the first GENERATE AC command or when Terminal Action Analysis results in the transaction to be completed online after the first GENERATE AC command.

Description:

In general, online processing is the same as online processing of magnetic stripe transactions and is not described here. This section is limited to the additional online processing provided in an ICC environment that is not available in a magnetic stripe environment.

The ARQC may be sent in the authorisation request message.[21] The authorisation response message from the issuer may contain the Issuer Authentication Data (tag '91'). If the Issuer Authentication Data is received in the authorisation response message and the Application Interchange Profile indicates that the ICC supports issuer authentication, the Issuer Authentication Data shall be sent to the ICC in the EXTERNAL AUTHENTICATE command. If the ICC responds with SW1 SW2 other than '9000', the terminal shall set the 'Issuer authentication failed' bit in the TVR to 1.

---

[20] If ECC key recovery or XDA signature verification failed and depending on the TAC-Denial and/or IAC-Denial, the terminal attempts to go online after the Terminal Action Analysis with the TC returned by the ICC.

[21] Actions performed by the acquirer or issuer systems are outside the scope of this specification. However, an explanation of what is expected to take place at the issuer may be useful for clarity. The ARQC is a cryptogram generated by the card from transaction data using an issuer key stored in the card and known at the issuer authorisation system. The issuer uses this key to authenticate the ARQC and thereby authenticate the card. This process is termed 'online card authentication' or simply 'card authentication'.

Subsequent to card authentication, the issuer may generate a cryptogram on selected data included in the authorisation response or already known to the card. This cryptogram is sent to the terminal in the authorisation response as part of the Issuer Authentication Data. The terminal provides the Issuer Authentication Data to the ICC in the EXTERNAL AUTHENTICATE command or the second GENERATE AC command, as described in Part I. The ICC may use the Issuer Authentication Data to authenticate that the response message originated from the issuer.

If the Issuer Authentication Data is received but the Application Interchange Profile indicates that the ICC does not support issuer authentication, this indicates that the ICC has combined the issuer authentication function with the GENERATE AC command. In this case, or if no Issuer Authentication Data is received, the terminal shall not execute the EXTERNAL AUTHENTICATE command.

The ICC shall permit at most one EXTERNAL AUTHENTICATE command in a transaction. If the terminal issues more than one, the second and all succeeding EXTERNAL AUTHENTICATE commands shall end with SW1 SW2 = '6985'.

Upon completion of online processing, if the EXTERNAL AUTHENTICATE command was sent to the card by the terminal, the terminal shall set the 'Issuer authentication was performed' bit in the TSI to 1.

Note: Annex F provides additional information about status words to be returned in response to an EXTERNAL AUTHENTICATE command.

10.10 Issuer-to-Card Script Processing

Purpose:

An issuer may provide command scripts to be delivered to the ICC by the terminal to perform functions that are not necessarily relevant to the current transaction but are important for the continued functioning of the application in the ICC. Multiple scripts may be provided with an authorisation response, and each may contain any number of Issuer Script Commands. Script processing is provided to allow for functions that are outside the scope of this specification but are nonetheless necessary.[22]

A script may contain Issuer Script Commands not known to the terminal, but the terminal shall deliver each command to the ICC individually according to this specification.

Conditions of Execution:

None.

Sequence of Execution:

Two separate script tags are available for use by the issuer. Issuer scripts with tag '71' shall be processed prior to issuing the final GENERATE AC command. Issuer scripts with tag '72' shall be processed after issuing the final GENERATE AC command.

Description:

An Issuer Script is a constructed data object (tag '71' or '72') containing (optionally) a Script Identifier and a sequence of Issuer Script Command APDUs to be delivered serially to the ICC. The Script Identifier is optional and is not interpreted by the terminal; it is meaningful only to the issuer. Figure 16 and Figure 17 illustrate an Issuer Script containing a Script Identifier and three commands.

T	L	T	L	Script ID	Commands
'71' or '72'	L(∑data, including Script ID, tags, and lengths)	'9F18'	'04'	Identifier (4 bytes)	(see Figure 17)

Figure 16: Issuer Script Format

T_1	L_1	V_1	T_2	L_2	V_2	T_3	L_3	V_3
'86'	$L(V_1)$	Command	'86'	$L(V_2)$	Command	'86'	$L(V_3)$	Command

Figure 17: Issuer Script Command Format (Shown with Three Commands)

[22] An example might be unblocking of an offline PIN, which might be done differently by various issuers or payment systems.

It is possible for multiple Issuer Scripts to be delivered with a single authorisation response. The terminal shall process each Issuer Script in the sequence in which it appears in the authorisation response according to the following rules:

- Issuer Script Commands shall be separated using the BER-TLV coding of the data objects defining the commands (tag '86').
- Each command shall be delivered to the ICC as a command APDU in the sequence in which it appeared in the Issuer Script.
- The terminal shall examine only SW1 in the response APDU and perform one of the following actions:
  - If SW1 indicates either normal processing or a ‘warning’ according to the conventions described in this specification, the terminal shall continue with the next command from the Issuer Script (if any).
  - If SW1 indicates an ‘error’ condition, the processing of the Issuer Script shall be terminated.

If an Issuer Script is processed, the terminal shall set the ‘Script processing was performed’ bit in the TSI to 1. If an error occurred in processing a script, the terminal shall set to 1 either the ‘Script processing failed before final GENERATE AC’ in the TVR if the identifying tag of the failing script was '71' or the ‘Script processing failed after final GENERATE AC’ in the TVR if the tag was '72'.

Note: Annex E discusses TVR and TSI bit settings following script processing.

10.11 Completion

Purpose:

The completion function closes processing of a transaction.

Conditions of Execution:

The terminal always performs this function unless the transaction is terminated prematurely by error processing.

Sequence of Execution:

The completion function is always the last function in the transaction processing. (Script processing may be performed after the completion function.)

Description:

The ICC indicates willingness to complete transaction processing by returning either a TC or an AAC to either the first or second GENERATE AC command issued by the terminal. If the terminal decides to go online, completion shall be done when the second GENERATE AC command is issued.

If the terminal is to perform CDA (as described in section 10.3), the terminal shall set the 'CDA signature requested' bits in the GENERATE AC command to 10.

If the terminal is to perform XDA (as described in section 10.3), the terminal shall set the 'XDA signature requested' bits in the GENERATE AC command to 01.

See section 9 for additional information on the use of the GENERATE AC command.

Part IV

Annexes

Annex A Data Elements Dictionary

Table 37 defines those data elements that may be used for financial transaction interchange and their mapping onto data objects and files. Table 38 lists the data elements in tag sequence.

The characters used in the "Format" column are described in section 4.3, Data Element Format Convention.

A1 Data Elements by Name

Table 37: Data Elements Dictionary

Name	Description	Source	Format	Template	Tag	Length
Account Type	Indicates the type of account selected on the terminal, coded as specified in Annex G	Terminal	n 2	—	'5F57'	1
Acquirer Identifier	Uniquely identifies the acquirer within each payment system	Terminal	n 6-11	—	'9F01'	6
Additional Terminal Capabilities	Indicates the data input and output capabilities of the terminal	Terminal	b	—	'9F40'	5
Amount, Authorised (Binary)	Authorised amount of the transaction (excluding adjustments)	Terminal	b	—	'81'	4
Amount, Authorised (Numeric)	Authorised amount of the transaction (excluding adjustments)	Terminal	n 12	—	'9F02'	6

Name	Description	Source	Format	Template	Tag	Length
Amount, Other (Binary)	Secondary amount associated with the transaction representing a cashback amount	Terminal	b	—	'9F04'	4
Amount, Other (Numeric)	Secondary amount associated with the transaction representing a cashback amount	Terminal	n 12	—	'9F03'	6
Amount, Reference Currency	Authorised amount expressed in the reference currency	Terminal	b	—	'9F3A'	4
Application Cryptogram	Cryptogram returned by the ICC in response of the GENERATE AC command	ICC	b	'77' or '80'	'9F26'	8
Application Currency Code	Indicates the currency in which the account is managed according to ISO 4217	ICC	n 3	'70' or '77'	'9F42'	2
Application Currency Exponent	Indicates the implied position of the decimal point from the right of the amount represented according to ISO 4217	ICC	n 1	'70' or '77'	'9F44'	1
Application Discretionary Data	Issuer or payment system specified data relating to the application	ICC	b	'70' or '77'	'9F05'	1-32
Application Effective Date	Date from which the application may be used	ICC	n 6 YYMMDD	'70' or '77'	'5F25'	3
Application Expiration Date	Date after which application expires	ICC	n 6 YYMMDD	'70' or '77'	'5F24'	3
Application File Locator (AFL)	Indicates the location (SFI, range of records) of the AEFs related to a given application	ICC	var.	'77' or '80'	'94'	var. up to 252

Name	Description	Source	Format	Template	Tag	Length
Application Dedicated File (ADF) Name	Identifies the application as described in ISO/IEC 7816-5	ICC	b	'61'	'4F'	5-16
Application Identifier (AID) – terminal	Identifies the application as described in ISO/IEC 7816-5	Terminal	b	—	'9F06'	5-16
Application Interchange Profile	Indicates the capabilities of the card to support specific functions in the application	ICC	b	'77' or '80'	'82'	2
Application Label	Mnemonic associated with the AID according to ISO/IEC 7816-5	ICC	ans with the special character limited to space	'61' or 'A5'	'50'	1-16
Application Preferred Name	Preferred mnemonic associated with the AID	ICC	ans (see section 4.3)	'61' or 'A5'	'9F12'	1-16
Application Primary Account Number (PAN)	Valid cardholder account number	ICC	cn var. up to 19	'70' or '77'	'5A'	var. up to 10
Application Primary Account Number (PAN) Sequence Number	Identifies and differentiates cards with the same PAN	ICC	n 2	'70' or '77'	'5F34'	1

Name	Description	Source	Format	Template	Tag	Length
Application Priority Indicator	Indicates the priority of a given application or group of applications in a directory	ICC	b	'61' or 'A5'	'87'	1
Application Reference Currency	1-4 currency codes used between the terminal and the ICC when the Transaction Currency Code is different from the Application Currency Code; each code is 3 digits according to ISO 4217	ICC	n 3	'70' or '77'	'9F3B'	2-8
Application Reference Currency Exponent	Indicates the implied position of the decimal point from the right of the amount, for each of the 1-4 reference currencies represented according to ISO 4217	ICC	n 1	'70' or '77'	'9F43'	1-4
Application Selection Indicator	For an application in the ICC to be supported by an application in the terminal, the Application Selection Indicator indicates whether the associated AID in the terminal must match the AID in the card exactly, including the length of the AID, or only up to the length of the AID in the terminal There is only one Application Selection Indicator per AID supported by the terminal	Terminal	At the discretion of the terminal. The data is not sent across the interface	—	—	See format

Name	Description	Source	Format	Template	Tag	Length
Application Selection Registered Proprietary Data (ASRPD)	Proprietary data allowing for proprietary processing during application selection. Proprietary data is identified using Proprietary Data Identifiers that are managed by EMVCo and their usage by the Application Selection processing is according to their intended usage, as agreed by EMVCo during registration	Card	b, also see Book 1 section 12.5	'73' 'BF0C'	'9F0A'	var.
Application Template	Contains one or more data objects relevant to an application directory entry according to ISO/IEC 7816-5	ICC	b	'70'	'61'	var. up to 252
Application Transaction Counter (ATC)	Counter maintained by the application in the ICC (incrementing the ATC is managed by the ICC)	ICC	b	'77' or '80'	'9F36'	2
Application Usage Control	Indicates issuer's specified restrictions on the geographic usage and services allowed for the application	ICC	b	'70' or '77'	'9F07'	2
Application Version Number	Version number assigned by the payment system for the application	ICC	b	'70' or '77'	'9F08'	2
Application Version Number	Version number assigned by the payment system for the application	Terminal	b	—	'9F09'	2
Authorisation Code	Value generated by the authorisation authority for an approved transaction	Issuer	As defined by the Payment Systems	—	'89'	6

Name	Description	Source	Format	Template	Tag	Length
Authorisation Response Code	Code that defines the disposition of a message	Issuer/ Terminal	an 2	—	'8A'	2
Authorisation Response Cryptogram (ARPC)	Cryptogram generated by the issuer and used by the card to verify that the response came from the issuer.	Issuer	b	—	—	4 or 8
Bank Identifier Code (BIC)	Uniquely identifies a bank as defined in ISO 9362.	ICC	var.	'BF0C' or '73'	'5F54'	8 or 11
Biometric Encryption Key (BEK)	An AES-128 key generated from the Biometric Key Seed, that is used to encrypt/decrypt the BDB constructed on the Biometric Processing Application	Terminal	b	—	—	16
Biometric Header Template (BHT)	A template defined in ISO/IEC 19785-3, that is nested under the BIT.	Card, Terminal	b	'7F60'	'A1'	var.
Biometric Information Template (BIT)	A template defined in ISO/IEC 19785-3, that describes information regarding the biometric format and solution supported in a card.	Card	b	'BF4A', 'BF4B'	'7F60'	var.
Biometric Information Template (BIT)	A template defined in ISO/IEC 19785-3, that describes information regarding the biometric format and solution supported in a terminal.	Terminal	b	—	'7F60'	var.
Biometric Key Seed	A random number generated by the terminal, that is used as the seed to generate the Biometric Encryption Key (BEK) and Biometric MAC Key (BMK)	Terminal	b	—	—	N_{PE} or N_{IC}

Name	Description	Source	Format	Template	Tag	Length
Biometric MAC Key (BMK)	A key generated from the Biometric Key Seed, that is used to ensure the integrity of the BDB	Terminal	b	—	—	32
Biometric Solution ID	A unique identifier assigned by EMVCo that is used to identify a biometric program, regional or global, supported by the card or terminal. The Biometric Solution ID is referred to within ISO/IEC 19785-3 as the "Index".	Terminal, Card	b	'A1' 'BF4E'	'90'	var.
Biometric Subtype	A data element defined in ISO/IEC 19785-3, that describes the subtype of the Biometric Type supported by the card or terminal, as shown in Table 50.	Terminal, Card	b	'A1'	'82'	1
Biometric Terminal Capabilities	A data element that identifies the Biometric CVM capabilities of the terminal	Terminal	b	—	'9F30'	3
Biometric Try Counters Template	A template that contains one or more of the following Biometric Try Counters: • Facial Try Counter • Finger Try Counter • Iris Try Counter • Palm Try Counter • Voice Try Counter	Card	b	—	'BF4C'	var.
Biometric Type	A data element defined in ISO/IEC 19785-3, that describes the type of biometrics supported by the card or terminal among facial, finger, iris, palm and voice, as shown in Table 49.	Terminal, Card	b	'A1' 'BF4E'	'81'	var.

Name	Description	Source	Format	Template	Tag	Length
Biometric Verification Data Template	A template that contains the TLV-coded values for the data to be included in the VERIFY command. The Biometric Verification Data Template contains Biometric Type ('81'), Biometric Solution ID ('90'), Enciphered Biometric Key Seed ('DF50'), Enciphered Biometric Data (tag 'DF51'), and MAC of Enciphered Biometric Data (tag 'DF52').	Terminal	b	—	'BF4E'	var.
Card BIT Group Template	A template in the card that contains one or more Biometric Information Templates (BITs)	Card	b	'70'	'9F31'	var.
Card Risk Management Data Object List 1 (CDOL1)	List of data objects (tag and length) to be passed to the ICC in the first GENERATE AC command	ICC	b	'70' or '77'	'8C'	var. up to 252
Card Risk Management Data Object List 2 (CDOL2)	List of data objects (tag and length) to be passed to the ICC in the second GENERATE AC command	ICC	b	'70' or '77'	'8D'	var. up to 252
Card Status Update (CSU)	Contains data sent to the ICC to indicate whether the issuer approves or declines the transaction, and to initiate actions specified by the issuer. Transmitted to the card in Issuer Authentication Data.	Issuer	b	—	—	4
Cardholder Name	Indicates cardholder name according to ISO/IEC 7813	ICC	ans 2-26	'70' or '77'	'5F20'	2-26

Name	Description	Source	Format	Template	Tag	Length
Cardholder Name Extended	Indicates the whole cardholder name when greater than 26 characters using the same coding convention as in ISO/IEC 7813	ICC	ans 27-45	'70' or '77'	'9F0B'	27-45
Cardholder Verification Method (CVM) List	Identifies a method of verification of the cardholder supported by the application	ICC	b	'70' or '77'	'8E'	10-252
Cardholder Verification Method (CVM) Results	Indicates the results of the last CVM performed	Terminal	b	—	'9F34'	3
Certification Authority Public Key Check Sum	A check value calculated on the concatenation of all parts of the Certification Authority Public Key (RID, Certification Authority Public Key Index, Certification Authority Public Key Modulus, Certification Authority Public Key Exponent) using SHA-1	Terminal	b	—	—	20
Certification Authority Public Key Exponent	Value of the exponent part of the Certification Authority Public Key	Terminal	b	—	—	1 or 3
Certification Authority Public Key Index	Identifies the certification authority's public key in conjunction with the RID	ICC	b	'70' or '77'	'8F'	1
Certification Authority Public Key Index	Identifies the certification authority's public key in conjunction with the RID	Terminal	b	—	'9F22'	1

Name	Description	Source	Format	Template	Tag	Length
Certification Authority Public Key Modulus	Value of the modulus part of the Certification Authority Public Key	Terminal	b	—	—	N_{CA} (up to 248)
Command Template	Identifies the data field of a command message	Terminal	b	—	'83'	var.
Cryptogram Information Data	Indicates the type of cryptogram and the actions to be performed by the terminal	ICC	b	'77' or '80'	'9F27'	1
Data Authentication Code	An issuer assigned value that is retained by the terminal during the verification process of the Signed Static Application Data	ICC	b	—	'9F45'	2
Dedicated File (DF) Name	Identifies the name of the DF as described in ISO/IEC 7816-4	ICC	b	'6F'	'84'	5-16
Default Dynamic Data Authentication Data Object List (DDOL)	DDOL to be used for constructing the INTERNAL AUTHENTICATE command if the DDOL in the card is not present	Terminal	b	—	—	var.
Default Transaction Certificate Data Object List (TDOL)	TDOL to be used for generating the TC Hash Value if the TDOL in the card is not present	Terminal	b	—	—	var.
Directory Definition File (DDF) Name	Identifies the name of a DF associated with a directory	ICC	b	'61'	'9D'	5-16

Name	Description	Source	Format	Template	Tag	Length
Directory Discretionary Template	Issuer discretionary part of the directory according to ISO/IEC 7816-5	ICC	var.	'61'	'73'	var. up to 252
Dynamic Data Authentication Data Object List (DDOL)	List of data objects (tag and length) to be passed to the ICC in the INTERNAL AUTHENTICATE command	ICC	b	'70' or '77'	'9F49'	up to 252
Enciphered Biometric Data	The enciphered data sent in the VERIFY command	Terminal	b	'BF4E'	'DF51'	var.
Enciphered Biometric Key Seed	The Biometric Key Seed enciphered using the public key from the ICC	Terminal	b	'BF4E'	'DF50'	N_{PE} or N_{IC}
Enciphered Personal Identification Number (PIN) Data	Transaction PIN enciphered at the PIN pad for online verification or for offline verification if the PIN pad and IFD are not a single integrated device	Terminal	b	—	—	8
Facial Try Counter	Identifies the number of facial verification tries remaining	Card	b	'BF4C'	'DF50'	1
File Control Information (FCI) Issuer Discretionary Data	Issuer discretionary part of the FCI	ICC	var.	'A5'	'BF0C'	var. up to 222
Finger Try Counter	Identifies the number of finger verification tries remaining	Card	b	'BF4C'	'DF51'	1

Name	Description	Source	Format	Template	Tag	Length
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.	'6F'	'A5'	var.
File Control Information (FCI) Template	Identifies the FCI template according to ISO/IEC 7816-4	ICC	var.	—	'6F'	var. up to 252
ICC Dynamic Number	Time-variant number generated by the ICC, to be captured by the terminal	ICC	b	—	'9F4C'	2-8
Integrated Circuit Card (ICC) PIN Encipherment Public Key Certificate (RSA) or Integrated Circuit Card (ICC) Public Key Certificate for ODE (ECC)	ICC Public Key certified by the issuer for PIN or biometric encipherment (ODE). Format and length are different depending on RSA or ECC.	ICC	b	'70' or '77'	'9F2D'	N_I or $N_{FIELD} + N_{SIG} + N_{HASH} + 17$

Name	Description	Source	Format	Template	Tag	Length
Integrated Circuit Card (ICC) PIN Encipherment Public Key Exponent	ICC PIN Encipherment Public Key Exponent used for PIN encipherment	ICC	b	'70' or '77'	'9F2E'	1 or 3
Integrated Circuit Card (ICC) PIN Encipherment Public Key Remainder	Remaining digits of the ICC PIN Encipherment Public Key Modulus	ICC	b	'70' or '77'	'9F2F'	$N_{PE} - N_I + 42$
Integrated Circuit Card (ICC) Public Key Certificate	ICC Public Key certified by the issuer. Format and length are different depending on RSA or ECC.	ICC	b	'70' or '77'	'9F46'	N_I or $N_{FIELD} + N_{SIG} + N_{HASH} + 17$
Integrated Circuit Card (ICC) Public Key Exponent	ICC Public Key Exponent used for the verification of the Signed Dynamic Application Data	ICC	b	'70' or '77'	'9F47'	1 or 3
Integrated Circuit Card (ICC) Public Key Remainder	Remaining digits of the ICC Public Key Modulus	ICC	b	'70' or '77'	'9F48'	$N_{IC} - N_I + 42$

Name	Description	Source	Format	Template	Tag	Length
Interface Device (IFD) Serial Number	Unique and permanent serial number assigned to the IFD by the manufacturer	Terminal	an 8	—	'9F1E'	8
International Bank Account Number (IBAN)	Uniquely identifies the account of a customer at a financial institution as defined in ISO 13616.	ICC	var.	'BF0C' or '73'	'5F53'	var. up to 34
Iris Try Counter	Identifies the number of iris verification tries remaining	Card	b	'BF4C'	'DF52'	1
Issuer Action Code – Default	Specifies the issuer's conditions that cause a transaction to be rejected if it might have been approved online, but the terminal is unable to process the transaction online	ICC	b	'70' or '77'	'9F0D'	5
Issuer Action Code – Denial	Specifies the issuer's conditions that cause the denial of a transaction without attempt to go online	ICC	b	'70' or '77'	'9F0E'	5
Issuer Action Code – Online	Specifies the issuer's conditions that cause a transaction to be transmitted online	ICC	b	'70' or '77'	'9F0F'	5
Issuer Application Data	Contains proprietary application data for transmission to the issuer in an online transaction. **Note:** For CCD-compliant applications, Annex C, section C7 defines the specific coding of the Issuer Application Data (IAD). To avoid potential conflicts with CCD-compliant applications, it is strongly recommended that the IAD data element in an application that is not CCD-compliant should not use the coding for a CCD-compliant application	ICC	b	'77' or '80'	'9F10'	var. up to 32

Name	Description	Source	Format	Template	Tag	Length
Issuer Authentication Data	Data sent to the ICC for online issuer authentication	Issuer	b	—	'91'	8-16
Issuer Code Table Index	Indicates the code table according to ISO/IEC 8859 for displaying the Application Preferred Name	ICC	n 2	'A5'	'9F11'	1
Issuer Country Code	Indicates the country of the issuer according to ISO 3166	ICC	n 3	'70' or '77'	'5F28'	2
Issuer Country Code (alpha2 format)	Indicates the country of the issuer as defined in ISO 3166 (using a 2 character alphabetic code)	ICC	a 2	'BF0C' or '73'	'5F55'	2
Issuer Country Code (alpha3 format)	Indicates the country of the issuer as defined in ISO 3166 (using a 3 character alphabetic code)	ICC	a 3	'BF0C' or '73'	'5F56'	3
Issuer Identification Number (IIN)	The number that identifies the major industry and the card issuer and that forms the first part of the Primary Account Number (PAN)	ICC	n 6	'BF0C' or '73'	'42'	3
Issuer Identification Number Extended (IINE)	The number that identifies the major industry and the card issuer and that forms the first part (6 or 8 digits) of the Primary Account Number (PAN). While the first 6 digits of the IINE (tag '9F0C') and IIN (tag '42') are the same and there is no need to have both data objects on the card, cards may have both the IIN and IINE data objects present.	ICC	n 6 or 8	'BF0C' or '73'	'9F0C'	var. 3 or 4

Name	Description	Source	Format	Template	Tag	Length
Issuer Public Key Certificate	Issuer public key certified by a certification authority. Format and length are different depending on RSA or ECC.	ICC	b	'70' or '77'	'90'	N_{CA} or $N_{FIELD} + N_{SIG} + 21$
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	'70' or '77'	'9F32'	1 or 3
Issuer Public Key Remainder	Remaining digits of the Issuer Public Key Modulus	ICC	b	'70' or '77'	'92'	$N_I - N_{CA} + 36$
Issuer Script Command	Contains a command for transmission to the ICC	Issuer	b	'71' or '72'	'86'	var. up to 261
Issuer Script Identifier	Identification of the Issuer Script	Issuer	b	'71' or '72'	'9F18'	4
Issuer Script Results	Indicates the result of the terminal script processing	Terminal	b	—	—	var.
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.
Issuer URL	The URL provides the location of the Issuer's Library Server on the Internet.	ICC	ans	'BF0C' or '73'	'5F50'	var.

Name	Description	Source	Format	Template	Tag	Length
Language Preference	1-4 languages stored in order of preference, each represented by 2 alphabetical characters according to ISO 639 **Note:** EMVCo strongly recommends that cards be personalised with data element '5F2D' coded in lowercase, but that terminals accept the data element whether it is coded in upper or lower case.	ICC	an 2	'A5'	'5F2D'	2-8
Last 4 Digits of PAN	The last four digits of the PAN, as defined in the EMV Payment Tokenisation Framework	ICC	n 4	'70' or '77'	'9F25'	2
Last Online Application Transaction Counter (ATC) Register	ATC value of the last transaction that went online	ICC	b	—	'9F13'	2
Log Entry	Provides the SFI of the Transaction Log file and its number of records	ICC	b	'BF0C' or '73'	'9F4D'	2
Log Format	List (in tag and length format) of data objects representing the logged data elements that are passed to the terminal when a transaction log record is read	ICC	b	—	'9F4F'	var.
Lower Consecutive Offline Limit	Issuer-specified preference for the maximum number of consecutive offline transactions for this ICC application allowed in a terminal with online capability	ICC	b	'70' or '77'	'9F14'	1
MAC of Enciphered Biometric Data	An HMAC generated on the Enciphered Biometric Data to ensure integrity	Terminal	b	'BF4E'	'DF52'	8

Name	Description	Source	Format	Template	Tag	Length
Maximum Target Percentage to be used for Biased Random Selection	Value used in terminal risk management for random transaction selection	Terminal	—	—	—	—
Merchant Category Code	Classifies the type of business being done by the merchant, represented according to ISO 8583:1993 for Card Acceptor Business Code	Terminal	n 4	—	'9F15'	2
Merchant Identifier	When concatenated with the Acquirer Identifier, uniquely identifies a given merchant	Terminal	ans 15	—	'9F16'	15
Merchant Name and Location	Indicates the name and location of the merchant	Terminal	ans	—	'9F4E'	var.
Message Type	Indicates whether the batch data capture record is a financial record or advice	Terminal	n 2	—	—	1
Offline BIT Group Template	A template, defined in ISO/IEC 7816-11, that is nested under the Card BIT Group Template, as shown in Table 51, and contains one or more multiple Biometric Information Templates (BITs) for offline biometric verification supported by card	Card	b	'9F31'	'BF4A'	var.

Name	Description	Source	Format	Template	Tag	Length
Online BIT Group Template	A template, defined in ISO/IEC 7816-11, that is nested under the Card BIT Group Template, as shown in Table 51, and contains one or more multiple Biometric Information Templates (BITs) for online biometric verification supported by card	Card	b	'9F31'	'BF4B'	var.
Palm Try Counter	Identifies the number of palm verification tries remaining	Card	b	'BF4C'	'DF53'	1
Payment Account Reference (PAR)	A non-financial reference assigned to each unique PAN and used to link a Payment Account represented by that PAN to affiliated Payment Tokens, as defined in the EMV Tokenisation Framework. The PAR may be assigned in advance of Payment Token issuance.	ICC	an 29 (see section 4.3)	'70' or '77'	'9F24'	29
Personal Identification Number (PIN) Pad Secret Key	Secret key of a symmetric algorithm used by the PIN pad to encipher the PIN and by the card reader to decipher the PIN if the PIN pad and card reader are not integrated	Terminal	—	—	—	—
Personal Identification Number (PIN) Try Counter	Number of PIN tries remaining	ICC	b	—	'9F17'	1
Point-of-Service (POS) Entry Mode	Indicates the method by which the PAN was entered, according to the first two digits of the ISO 8583:1987 POS Entry Mode	Terminal	n 2	—	'9F39'	1

Name	Description	Source	Format	Template	Tag	Length
Preferred Attempts Template[23]	A template that contains the TLV-coded values for the preferred attempts of any BIT of a Biometric Type. It contains one or more of the following: • Preferred Facial Attempts • Preferred Finger Attempts • Preferred Iris Attempts • Preferred Palm Attempts • Preferred Voice Attempts	Card	b		'BF4D'	var.
Preferred Facial Attempts	Number of preferred attempts for any BIT of the facial Biometric Type stored in the card.	Card	b	'BF4D'	'DF50'	1
Preferred Finger Attempts	Number of preferred attempts for any BIT of the finger Biometric Type stored in the card.	Card	b	'BF4D'	'DF51'	1
Preferred Iris Attempts	Number of preferred attempts for any BIT of the iris Biometric Type stored in the card	Card	b	'BF4D'	'DF52'	1
Preferred Palm Attempts	Number of preferred attempts for any BIT of the palm Biometric Type stored in the card.	Card	b	'BF4D'	'DF53'	1
Preferred Voice Attempts	Number of preferred attempts for any BIT of the voice Biometric Type stored in the card.	Card	b	'BF4D'	'DF54'	1

[23]The preferred attempts for each Biometric Type are used to define how many attempts the issuer allows the terminal to verify using a single BIT. It is different from any biometric try limit used within the card to limit how many tries are allowed by the issuer for a Biometric Type.

For example, an issuer may set a finger try limit in the card (associated with the Finger Try Counter) to five, and set the Preferred Finger Attempts to three, allowing three attempts to verify the primary finger before two additional attempts are allowed using a secondary finger.

Name	Description	Source	Format	Template	Tag	Length
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	'A5'	'9F38'	var.
Proprietary Authentication Data	Contains issuer data for transmission to the card in the Issuer Authentication Data of an online transaction.	Issuer	b	—	—	var. up to 8
READ RECORD Response Message Template	Contains the contents of the record read. (Mandatory for SFIs 1-10. Response messages for SFIs 11-30 are outside the scope of EMV, but may use template '70')	ICC	var.	—	'70'	var. up to 252
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'	var.
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'	var.
Service Code	Service code as defined in ISO/IEC 7813 for track 1 and track 2	ICC	n 3	'70' or '77'	'5F30'	2
Short File Identifier (SFI)	Identifies the AEF referenced in commands related to a given ADF or DDF. It is a binary data object having a value in the range 1 to 30 and with the three high order bits set to zero.	ICC	b	'A5'	'88'	1

Name	Description	Source	Format	Template	Tag	Length
Signed Dynamic Application Data	Digital signature on critical application parameters for DDA or CDA	ICC	b	'77' or '80'	'9F4B'	N_{IC}
Signed Static Application Data	Digital signature on critical application parameters for SDA	ICC	b	'70' or '77'	'93'	N_I
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	—	'70' or '77'	'9F4A'	var.
Target Percentage to be Used for Random Selection	Value used in terminal risk management for random transaction selection	Terminal	—	—	—	—
Template Try Counter	Identifies the number of biometric verification tries remaining for a specific BIT	Terminal	b	—	—	1
Terminal Action Code – Default	Specifies the acquirer's conditions that cause a transaction to be rejected if it might have been approved online, but the terminal is unable to process the transaction online	Terminal	b	—	—	5
Terminal Action Code – Denial	Specifies the acquirer's conditions that cause the denial of a transaction without attempt to go online	Terminal	b	—	—	5
Terminal Action Code – Online	Specifies the acquirer's conditions that cause a transaction to be transmitted online	Terminal	b	—	—	5

Name	Description	Source	Format	Template	Tag	Length
Terminal BIT Group Template	A template in the terminal, defined in ISO/IEC 7816-11, that contains one or more Biometric Information Templates (BITs)	Terminal	b	—	—	var.
Terminal Capabilities	Indicates the card data input, CVM, and security capabilities of the terminal for the selected AID	Terminal	b	—	'9F33'	3
Terminal Country Code	Indicates the country of the terminal, represented according to ISO 3166	Terminal	n 3	—	'9F1A'	2
Terminal Floor Limit	Indicates the floor limit in the terminal in conjunction with the AID	Terminal	b	—	'9F1B'	4
Terminal Identification	Designates the unique location of a terminal at a merchant	Terminal	an 8	—	'9F1C'	8
Terminal Risk Management Data	Application-specific value used by the card for risk management purposes	Terminal	b	—	'9F1D'	1-8
Terminal Type	Indicates the environment of the terminal, its communications capability, and its operational control	Terminal	n 2	—	'9F35'	1
Terminal Verification Results	Status of the different functions as seen from the terminal	Terminal	b	—	'95'	5
Threshold Value for Biased Random Selection	Value used in terminal risk management for random transaction selection	Terminal	—	—	—	—

Name	Description	Source	Format	Template	Tag	Length
Token Requestor ID	Uniquely identifies the pairing of the Token Requestor with the Token Domain, as defined in the EMV Payment Tokenisation Framework	ICC	n 11	'70' or '77'	'9F19'	6
Track 1 Discretionary Data	Discretionary part of track 1 according to ISO/IEC 7813	ICC	ans	'70' or '77'	'9F1F'	var.
Track 2 Discretionary Data	Discretionary part of track 2 according to ISO/IEC 7813	ICC	cn	'70' or '77'	'9F20'	var.
Track 2 Equivalent Data	Contains the data elements of track 2 according to ISO/IEC 7813, excluding start sentinel, end sentinel, and Longitudinal Redundancy Check (LRC), as follows: Primary Account Number Field Separator (Hex 'D') Expiration Date (YYMM) Service Code Discretionary Data (defined by individual payment systems) Pad with one Hex 'F' if needed to ensure whole bytes	ICC	b n, var. up to 19 b n 4 n 3 n, var. b	'70' or '77'	'57'	var. up to 19
Transaction Amount	Clearing amount of the transaction, including tips and other adjustments	Terminal	n 12	—	—	6

Name	Description	Source	Format	Template	Tag	Length
Transaction Certificate Data Object List (TDOL)	List of data objects (tag and length) to be used by the terminal in generating the TC Hash Value	ICC	b	'70' or '77'	'97'	var. up to 252
Transaction Certificate (TC) Hash Value	Result of a hash function specified in Book 2, section B3.1	Terminal	b	—	'98'	20
Transaction Currency Code	Indicates the currency code of the transaction according to ISO 4217	Terminal	n 3	—	'5F2A'	2
Transaction Currency Exponent	Indicates the implied position of the decimal point from the right of the transaction amount represented according to ISO 4217	Terminal	n 1	—	'5F36'	1
Transaction Date	Local date that the transaction was authorised	Terminal	n 6 YYMMDD	—	'9A'	3
Transaction Personal Identification Number (PIN) Data	Data entered by the cardholder for the purpose of the PIN verification	Terminal	b	—	'99'	var.
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

Name	Description	Source	Format	Template	Tag	Length
Transaction Reference Currency Conversion	Factor used in the conversion from the Transaction Currency Code to the Transaction Reference Currency Code	Terminal	n 8	—	—	4
Transaction Reference Currency Exponent	Indicates the implied position of the decimal point from the right of the transaction amount, with the Transaction Reference Currency Code represented according to ISO 4217	Terminal	n 1	—	'9F3D'	1
Transaction Sequence Counter	Counter maintained by the terminal that is incremented by one for each transaction	Terminal	n 4-8	—	'9F41'	2-4
Transaction Status Information	Indicates the functions performed in a transaction	Terminal	b	—	'9B'	2
Transaction Time	Local time that the transaction was authorised	Terminal	n 6 HHMMSS	—	'9F21'	3
Transaction Type	Indicates the type of financial transaction, represented by the first two digits of the ISO 8583:1987 Processing Code. The actual values to be used for the Transaction Type data element are defined by the relevant payment system	Terminal	n 2	—	'9C'	1
Unpredictable Number	Value to provide variability and uniqueness to the generation of a cryptogram	Terminal	b	—	'9F37'	4

Name	Description	Source	Format	Template	Tag	Length
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	'70' or '77'	'9F23'	1
Voice Try Counter	Identifies the number of voice verification tries remaining	Card	b	'BF4C'	'DF54'	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 hexadecimal 'F's.
- A data element in format a, an, or ans is left justified and padded with trailing hexadecimal zeroes.

When data is moved from one entity to another (for example, card to terminal), it shall always be passed in order from high order to low order, regardless of how it is internally stored. The same rule applies when concatenating data.

Note: Data that can occur in template '70' or '77' can never occur in both.

A2 Data Elements by Tag

Name	Template	Tag
Issuer Identification Number (IIN)	'BF0C' or '73'	'42'
Application Dedicated File (ADF) Name	'61'	'4F'
Application Label	'61' or 'A5'	'50'
Track 2 Equivalent Data	'70' or '77'	'57'
Application Primary Account Number (PAN)	'70' or '77'	'5A'
Cardholder Name	'70' or '77'	'5F20'
Application Expiration Date	'70' or '77'	'5F24'
Application Effective Date	'70' or '77'	'5F25'
Issuer Country Code	'70' or '77'	'5F28'
Transaction Currency Code	—	'5F2A'
Language Preference	'A5'	'5F2D'
Service Code	'70' or '77'	'5F30'
Application Primary Account Number (PAN) Sequence Number	'70' or '77'	'5F34'
Transaction Currency Exponent	—	'5F36'
Issuer URL	'BF0C' or '73'	'5F50'
International Bank Account Number (IBAN)	'BF0C' or '73'	'5F53'
Bank Identifier Code (BIC)	'BF0C' or '73'	'5F54'
Issuer Country Code (alpha2 format)	'BF0C' or '73'	'5F55'
Issuer Country Code (alpha3 format)	'BF0C' or '73'	'5F56'
Account Type	—	'5F57'
Application Template	'70' or '77'	'61'
File Control Information (FCI) Template	—	'6F'
READ RECORD Response Message Template	—	'70'
Issuer Script Template 1	—	'71'
Issuer Script Template 2	—	'72'
Directory Discretionary Template	'61'	'73'

Table 38: Data Elements Tags

Name	Template	Tag
Response Message Template Format 2	—	'77'
Biometric Information Template (BIT), card	'BF4A' or 'BF4B'	'7F60'
Biometric Information Template (BIT), terminal	–	'7F60'
Response Message Template Format 1	—	'80'
Amount, Authorised (Binary)	—	'81'
Biometric Type	'A1' or 'BF4E'	'81'
Application Interchange Profile	'77' or '80'	'82'
Biometric Subtype	'A1'	'82'
Command Template	—	'83'
Dedicated File (DF) Name	'6F'	'84'
Issuer Script Command	'71' or '72'	'86'
Application Priority Indicator	'61' or 'A5'	'87'
Short File Identifier (SFI)	'A5'	'88'
Authorisation Code	—	'89'
Authorisation Response Code	—	'8A'
Card Risk Management Data Object List 1 (CDOL1)	'70' or '77'	'8C'
Card Risk Management Data Object List 2 (CDOL2)	'70' or '77'	'8D'
Cardholder Verification Method (CVM) List	'70' or '77'	'8E'
Certification Authority Public Key Index	'70' or '77'	'8F'
Issuer Public Key Certificate	'70' or '77'	'90'
Biometric Solution ID	'A1' or 'BF4E'	'90'
Issuer Authentication Data	—	'91'
Issuer Public Key Remainder	'70' or '77'	'92'
Signed Static Application Data	'70' or '77'	'93'
Application File Locator (AFL)	'77' or '80'	'94'
Terminal Verification Results	—	'95'
Transaction Certificate Data Object List (TDOL)	'70' or '77'	'97'
Transaction Certificate (TC) Hash Value	—	'98'
Transaction Personal Identification Number (PIN) Data	—	'99'

Table 38: Data Elements Tags, continued

Name	Template	Tag
Transaction Date	—	'9A'
Transaction Status Information	—	'9B'
Transaction Type	—	'9C'
Directory Definition File (DDF) Name	'61'	'9D'
Acquirer Identifier	—	'9F01'
Amount, Authorised (Numeric)	—	'9F02'
Amount, Other (Numeric)	—	'9F03'
Amount, Other (Binary)	—	'9F04'
Application Discretionary Data	'70' or '77'	'9F05'
Application Identifier (AID) – terminal	—	'9F06'
Application Usage Control	'70' or '77'	'9F07'
Application Version Number	'70' or '77'	'9F08'
Application Version Number	—	'9F09'
Application Selection Registered Proprietary Data (ASRPD)	'73'	'9F0A'
Cardholder Name Extended	'70' or '77'	'9F0B'
Issuer Identification Number Extended (IINE)	'BF0C' or '73'	'9F0C'
Issuer Action Code – Default	'70' or '77'	'9F0D'
Issuer Action Code – Denial	'70' or '77'	'9F0E'
Issuer Action Code – Online	'70' or '77'	'9F0F'
Issuer Application Data	'77' or '80'	'9F10'
Issuer Code Table Index	'A5'	'9F11'
Application Preferred Name	'61' or 'A5'	'9F12'
Last Online Application Transaction Counter (ATC) Register	—	'9F13'
Lower Consecutive Offline Limit	'70' or '77'	'9F14'
Merchant Category Code	—	'9F15'
Merchant Identifier	—	'9F16'
Personal Identification Number (PIN) Try Counter	—	'9F17'
Issuer Script Identifier	'71' or '72'	'9F18'

Table 38: Data Elements Tags, continued

Name	Template	Tag
Token Requestor ID	'70' or '77'	'9F19'
Terminal Country Code	—	'9F1A'
Terminal Floor Limit	—	'9F1B'
Terminal Identification	—	'9F1C'
Terminal Risk Management Data	—	'9F1D'
Interface Device (IFD) Serial Number	—	'9F1E'
Track 1 Discretionary Data	'70' or '77'	'9F1F'
Track 2 Discretionary Data	'70' or '77'	'9F20'
Transaction Time	—	'9F21'
Certification Authority Public Key Index	—	'9F22'
Upper Consecutive Offline Limit	'70' or '77'	'9F23'
Payment Account Reference (PAR)	'70' or '77'	'9F24'
Last 4 Digits of PAN	'70' or '77'	'9F25'
Application Cryptogram	'77' or '80'	'9F26'
Cryptogram Information Data	'77' or '80'	'9F27'
ICC PIN Encipherment Public Key Certificate (RSA) or Integrated Circuit Card (ICC) Public Key Certificate for ODE (ECC)	'70' or '77'	'9F2D'
ICC PIN Encipherment Public Key Exponent	'70' or '77'	'9F2E'
ICC PIN Encipherment Public Key Remainder	'70' or '77'	'9F2F'
Biometric Terminal Capabilities	–	'9F30'
Card BIT Group Template	'70'	'9F31'
Issuer Public Key Exponent	'70' or '77'	'9F32'
Terminal Capabilities	—	'9F33'
Cardholder Verification Method (CVM) Results	—	'9F34'
Terminal Type	—	'9F35'
Application Transaction Counter (ATC)	'77' or '80'	'9F36'
Unpredictable Number	—	'9F37'
Processing Options Data Object List (PDOL)	'A5'	'9F38'

Table 38: Data Elements Tags, continued

Name	Template	Tag
Point-of-Service (POS) Entry Mode	—	'9F39'
Amount, Reference Currency	—	'9F3A'
Application Reference Currency	'70' or '77'	'9F3B'
Transaction Reference Currency Code	—	'9F3C'
Transaction Reference Currency Exponent	—	'9F3D'
Additional Terminal Capabilities	—	'9F40'
Transaction Sequence Counter	—	'9F41'
Application Currency Code	'70' or '77'	'9F42'
Application Reference Currency Exponent	'70' or '77'	'9F43'
Application Currency Exponent	'70' or '77'	'9F44'
Data Authentication Code	—	'9F45'
ICC Public Key Certificate	'70' or '77'	'9F46'
ICC Public Key Exponent	'70' or '77'	'9F47'
ICC Public Key Remainder	'70' or '77'	'9F48'
Dynamic Data Authentication Data Object List (DDOL)	'70' or '77'	'9F49'
Static Data Authentication Tag List	'70' or '77'	'9F4A'
Signed Dynamic Application Data	'77' or '80'	'9F4B'
ICC Dynamic Number	—	'9F4C'
Log Entry	'BF0C' or '73'	'9F4D'
Merchant Name and Location	—	'9F4E'
Log Format	—	'9F4F'
Biometric Header Template (BHT)	'7F60'	'A1'
File Control Information (FCI) Proprietary Template	'6F'	'A5'
File Control Information (FCI) Issuer Discretionary Data	'A5'	'BF0C'
Offline BIT Group Template	'9F31'	'BF4A'
Online BIT Group Template	'9F31'	'BF4B'
Biometric Try Counters Template	–	'BF4C'
Preferred Attempts Template	–	'BF4D'
Biometric Verification Data Template	–	'BF4E'

Table 38: Data Elements Tags, continued

Name	Template	Tag
Facial Try Counter	'BF4C'	'DF50'
Preferred Facial Attempts	'BF4D'	'DF50'
Enciphered Biometric Key Seed	'BF4E'	'DF50'
Finger Try Counter	'BF4C'	'DF51'
Preferred Finger Attempts	'BF4D'	'DF51'
Enciphered Biometric Data	'BF4E'	'DF51'
Iris Try Counter	'BF4C'	'DF52'
Preferred Iris Attempts	'BF4D'	'DF52'
MAC of Enciphered Biometric Data	'BF4E'	'DF52'
Palm Try Counter	'BF4C'	'DF53'
Preferred Palm Attempts	'BF4D'	'DF53'
Voice Try Counter	'BF4C'	'DF54'
Preferred Voice Attempts	'BF4D'	'DF54'

Table 38: Data Elements Tags, continued

Annex B Rules for BER-TLV Data Objects

As defined in ISO/IEC 8825, a BER-TLV data object consists of 2-3 consecutive fields:

- The tag field (T) consists of one or more consecutive bytes. It indicates a class, a type, and a number (see Table 39). The tag field of the data objects described in this specification is coded on one or two bytes.
- The length field (L) consists of one or more consecutive bytes. It indicates the length of the following field. The length field of the data objects described in this specification which are transmitted over the card-terminal interface is coded on one or two bytes.

  Note: Three length bytes may be used if needed for templates '71' and '72' and tag '86' (to express length greater than 255 bytes), as they are not transmitted over the card-terminal interface.
- The value field (V) indicates the value of the data object. If L = '00', the value field is not present.

A BER-TLV data object belongs to one of the following two categories:

- A primitive data object where the value field contains a data element for financial transaction interchange.
- A constructed data object where the value field contains one or more primitive or constructed data objects. The value field of a constructed data object is called a template.

The coding of BER-TLV data objects is defined as follows.

B1 Coding of the Tag Field of BER-TLV Data Objects

Table 39 describes the first byte of the tag field of a BER-TLV data object:

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0							Universal class
0	1							Application class
1	0							Context-specific class
1	1							Private class
		0						Primitive data object
		1						Constructed data object
			1	1	1	1	1	See subsequent bytes
			Any other value <31					Tag number

Table 39: Tag Field Structure (First Byte) BER-TLV

According to ISO/IEC 8825, Table 40 defines the coding rules of the subsequent bytes of a BER-TLV tag when tag numbers ≥ 31 are used (that is, bits b5 – b1 of the first byte equal 11111).

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1								Another byte follows
0								Last tag byte
	Any value > 0							(Part of) tag number

Table 40: Tag Field Structure (Subsequent Bytes) BER-TLV

Before, between, or after TLV-coded data objects, '00' bytes without any meaning may occur (for example, due to erased or modified TLV-coded data objects).

Note: It is strongly recommended that issuers do not use tags beginning with 'FF' for proprietary purposes, as existing terminals may not recognise 'FF' as the beginning of a constructed private class tag.

The tag field of a BER-TLV data object is coded according to the following rules:

- The following application class templates defined in ISO/IEC 7816 apply: '61' and '6F'.
- The following range of application class templates is defined in Part II: '70' to '7F'. The meaning is then specific to the context of an application according to this specification. Tags '78', '79', '7D', and '7E' are defined in ISO/IEC 7816-6 and are not used in this specification.
- The application class data objects defined in ISO/IEC 7816 and described in Part II are used according to the ISO/IEC 7816 definition.
- Context-specific class data objects are defined in the context of this specification or in the context of the template in which they appear.
- The coding of primitive context-specific class data objects in the ranges '80' to '9E' and '9F00' to '9F4F' is reserved for this specification.
- The coding of primitive context-specific class data objects in the range '9F50' to '9F7F' is reserved for the payment systems. Payment system-specific tags are interpreted within the context of the application RID.
- The coding of tag 'BF0C' and constructed context-specific class data objects in the range 'BF20' to 'BF4F' is reserved for this specification.
- The coding of constructed context-specific class data objects in the ranges 'BF10' to 'BF1F' and 'BF50' to 'BF6F' is reserved for the payment systems. Payment system-specific tags are interpreted within the context of the application RID.
- The coding of constructed context-specific class data objects in the ranges 'BF01' to 'BF0B', 'BF0D' to 'BF0F', and 'BF70' to 'BF7F' is left to the discretion of the issuer. Issuer-specific tags are interpreted within the context of the Issuer Identification Number (as defined in ISO/IEC 7812-1). Additionally, to satisfy business requirements such as proprietary domestic processing, multiple issuers may agree on the definition of a private class tag. Such tags may be interpreted in the context of other data such as Issuer Country Code.
- The coding of primitive and constructed private class data objects is left to the discretion of the issuer. Issuer-specific tags are interpreted within the context of the Issuer Identification Number (as defined in ISO/IEC 7812-1). Additionally, to satisfy business requirements such as proprietary domestic processing, multiple issuers may agree on the definition of a private class tag. Such tags may be interpreted in the context of other data such as Issuer Country Code.

B2 Coding of the Length Field of BER-TLV Data Objects

When bit b8 of the most significant byte of the length field is set to 0, the length field consists of only one byte. Bits b7 – b1 code the number of bytes of the value field. The length field is within the range 1 to 127.

When bit b8 of the most significant byte of the length field is set to 1, the subsequent bits b7 – b1 of the most significant byte code the number of subsequent bytes in the length field. The subsequent bytes code an integer representing the number of bytes in the value field. Two bytes are necessary to express up to 255 bytes in the value field.

B3 Coding of the Value Field of Data Objects

A data element is the value field (V) of a primitive BER-TLV data object. A data element is the smallest data field that receives an identifier (a tag).

A primitive data object is structured as illustrated in Figure 18:

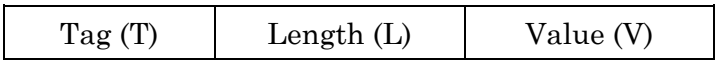


Figure 18: Primitive BER-TLV Data Object (Data Element)

A constructed BER-TLV data object consists of a tag, a length, and a value field composed of one or more BER-TLV data objects. A record in an AEF governed by this specification is a constructed BER-TLV data object. A constructed data object is structured as illustrated in Figure 19:

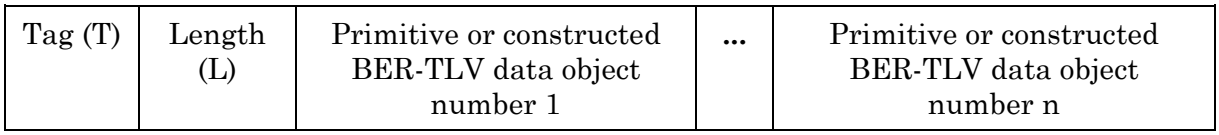


Figure 19: Constructed BER-TLV Data Object

Annex C Coding of Data Elements Used in Transaction Processing

This annex provides the coding for dynamic card data elements and specific data elements used to control the transaction flow in the terminal or to record the status of processing for the transaction. In the tables:

- A '1' means that if that bit has the value 1, the corresponding 'Meaning' applies.
- An 'x' means that the bit does not apply.

Data (bytes or bits) indicated as RFU shall be set to 0.

C1 Application Interchange Profile

AIP Byte 1 (Leftmost)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	x	x	x	x	x	x	x	XDA supported
x	1	x	x	x	x	x	x	SDA supported
x	x	1	x	x	x	x	x	DDA supported
x	x	x	1	x	x	x	x	Cardholder verification is supported
x	x	x	x	1	x	x	x	Terminal risk management is to be performed
x	x	x	x	x	1	x	x	Issuer authentication is supported [24]
x	x	x	x	x	x	0	x	Reserved for use by the EMV Contactless Specifications
x	x	x	x	x	x	x	1	CDA supported

AIP Byte 2 (Rightmost)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	x	x	x	x	x	x	x	Reserved for use by the EMV Contactless Specifications
x	0	x	x	x	x	x	x	Reserved for use by the EMV Contactless Specifications
x	x	0	x	x	x	x	x	Reserved for use by the EMV Contactless Specifications
x	x	x	0	x	x	x	x	RFU
x	x	x	x	0	x	x	x	RFU
x	x	x	x	x	0	x	x	RFU
x	x	x	x	x	x	0	x	RFU
x	x	x	x	x	x	x	0	Reserved for use by the EMV Contactless Specifications

Table 41: Application Interchange Profile

[24] When this bit is set to 1, Issuer Authentication using the EXTERNAL AUTHENTICATE command is supported

C2 Application Usage Control

Application Usage Control Byte 1 (Leftmost)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	x	x	x	x	x	x	x	Valid for domestic cash transactions
x	1	x	x	x	x	x	x	Valid for international cash transactions
x	x	1	x	x	x	x	x	Valid for domestic goods
x	x	x	1	x	x	x	x	Valid for international goods
x	x	x	x	1	x	x	x	Valid for domestic services
x	x	x	x	x	1	x	x	Valid for international services
x	x	x	x	x	x	1	x	Valid at ATMs
x	x	x	x	x	x	x	1	Valid at terminals other than ATMs

Application Usage Control Byte 2 (Rightmost)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	x	x	x	x	x	x	x	Domestic cashback allowed
x	1	x	x	x	x	x	x	International cashback allowed
x	x	0	x	x	x	x	x	RFU
x	x	x	0	x	x	x	x	RFU
x	x	x	x	0	x	x	x	RFU
x	x	x	x	x	0	x	x	RFU
x	x	x	x	x	x	0	x	RFU
x	x	x	x	x	x	x	0	RFU

Table 42: Application Usage Control

C3 Cardholder Verification Rule Format

CV Rule Byte 1 (Leftmost): Cardholder Verification Method (CVM) Codes

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							Fail cardholder verification if this CVM is unsuccessful
	1							Apply succeeding CV Rule if this CVM is unsuccessful
		0	0	0	0	0	0	Fail CVM processing
		0	0	0	0	0	1	Plaintext PIN verification performed by ICC
		0	0	0	0	1	0	Enciphered PIN verified online
		0	0	0	0	1	1	Plaintext PIN verification performed by ICC and signature
		0	0	0	1	0	0	Enciphered PIN verification performed by ICC
		0	0	0	1	0	1	Enciphered PIN verification performed by ICC and signature
		0	0	0	1	1	0	Facial biometric verified offline (by ICC)
		0	0	0	1	1	1	Facial biometric verified online
		0	0	1	0	0	0	Finger biometric verified offline (by ICC)
		0	0	1	0	0	1	Finger biometric verified online
		0	0	1	0	1	0	Palm biometric verified offline (by ICC)
		0	0	1	0	1	1	Palm biometric verified online
		0	0	1	1	0	0	Iris biometric verified offline (by ICC)
		0	0	1	1	0	1	Iris biometric verified online
		0	0	1	1	1	0	Voice biometric verified offline (by ICC)
		0	0	1	1	1	1	Voice biometric verified online

Table 43: CVM Codes

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
		0	1	x	x	x	x	Values in the range 010000-011101 reserved for future use by this specification
		0	1	1	1	1	0	Signature
		0	1	1	1	1	1	No CVM required
		1	0	x	x	x	x	Values in the range 100000-101111 reserved for use by the individual payment systems
		1	1	x	x	x	x	Values in the range 110000-111110 reserved for use by the issuer
		1	1	1	1	1	1	This value is not available for use

Table 43: CVM Codes, continued

CV Rule Byte 2 (Rightmost): Cardholder Verification Method (CVM) Condition Codes

Value	Meaning
'00'	Always
'01'	If unattended cash
'02'	If not unattended cash and not manual cash and not purchase with cashback
'03'	If terminal supports the CVM [25]
'04'	If manual cash
'05'	If purchase with cashback
'06'	If transaction is in the application currency [26] and is under X value (see section 10.5 for a discussion of "X")
'07'	If transaction is in the application currency and is over X value
'08'	If transaction is in the application currency and is under Y value (see section 10.5 for a discussion of "Y")
'09'	If transaction is in the application currency and is over Y value
'0A' - '7F'	RFU
'80' - 'FF'	Reserved for use by individual payment systems

Table 44: CVM Condition Codes

[25] Support for a CVM is described in Book 4 section 6.3.4, second paragraph.

[26] That is, Transaction Currency Code = Application Currency Code.

C4 Issuer Code Table Index

Value	Refers to
'01'	Part 1 of ISO/IEC 8859
'02'	Part 2 of ISO/IEC 8859
'03'	Part 3 of ISO/IEC 8859
'04'	Part 4 of ISO/IEC 8859
'05'	Part 5 of ISO/IEC 8859
'06'	Part 6 of ISO/IEC 8859
'07'	Part 7 of ISO/IEC 8859
'08'	Part 8 of ISO/IEC 8859
'09'	Part 9 of ISO/IEC 8859
'10'	Part 10 of ISO/IEC 8859

Table 45: Issuer Code Table Index

C5 Terminal Verification Results

TVR Byte 1: (Leftmost)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	x	x	x	x	x	x	x	Offline data authentication was not performed
x	1	x	x	x	x	x	x	SDA failed
x	x	1	x	x	x	x	x	ICC data missing
x	x	x	1	x	x	x	x	Card appears on terminal exception file [27]
x	x	x	x	1	x	x	x	DDA failed
x	x	x	x	x	1	x	x	CDA failed
x	x	x	x	x	x	1	x	SDA selected
x	x	x	x	x	x	x	1	XDA selected

TVR Byte 2:

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	x	x	x	x	x	x	x	ICC and terminal have different application versions
x	1	x	x	x	x	x	x	Expired application
x	x	1	x	x	x	x	x	Application not yet effective
x	x	x	1	x	x	x	x	Requested service not allowed for card product
x	x	x	x	1	x	x	x	New card
x	x	x	x	x	0	x	x	RFU
x	x	x	x	x	x	1	x	Biometric performed and successful
x	x	x	x	x	x	x	1	Biometric template format not supported

Table 46: Terminal Verification Results

[27] There is no requirement in this specification for an exception file, but it is recognised that some terminals may have this capability.

TVR Byte 3:

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	x	x	x	x	x	x	x	Cardholder verification was not successful
x	1	x	x	x	x	x	x	Unrecognised CVM
x	x	1	x	x	x	x	x	PIN Try Limit exceeded
x	x	x	1	x	x	x	x	PIN entry required and PIN pad not present or not working
x	x	x	x	1	x	x	x	PIN entry required, PIN pad present, but PIN was not entered
x	x	x	x	x	1	x	x	Online CVM captured
x	x	x	x	x	x	1	x	Biometric required but Biometric capture device not working
x	x	x	x	x	x	x	1	Biometric required, Biometric capture device present, but Biometric Subtype entry was bypassed

TVR Byte 4:

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	x	x	x	x	x	x	x	Transaction exceeds floor limit
x	1	x	x	x	x	x	x	Lower consecutive offline limit exceeded
x	x	1	x	x	x	x	x	Upper consecutive offline limit exceeded
x	x	x	1	x	x	x	x	Transaction selected randomly for online processing
x	x	x	x	1	x	x	x	Merchant forced transaction online
x	x	x	x	x	1	x	x	Biometric Try Limit exceeded
x	x	x	x	x	x	1	x	A selected Biometric Type not supported
x	x	x	x	x	x	x	1	XDA signature verification failed

Table 46: Terminal Verification Results, continued

TVR Byte 5 (Rightmost):

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	x	x	x	x	x	x	x	Default TDOL used
x	1	x	x	x	x	x	x	Issuer authentication failed
x	x	1	x	x	x	x	x	Script processing failed before final GENERATE AC
x	x	x	1	x	x	x	x	Script processing failed after final GENERATE AC
x	x	x	x	0	x	x	x	Reserved for use by the EMV Contactless Specifications
x	x	x	x	x	1	x	x	CA ECC key missing
x	x	x	x	x	x	1	x	ECC key recovery failed
x	x	x	x	x	x	x	0	Reserved for use by the EMV Contactless Specifications

Table 46: Terminal Verification Results, continued

C6 Transaction Status Information

TSI Byte 1 (Leftmost):

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	x	x	x	x	x	x	x	Offline data authentication was performed
x	1	x	x	x	x	x	x	Cardholder verification was performed
x	x	1	x	x	x	x	x	Card risk management was performed
x	x	x	1	x	x	x	x	Issuer authentication was performed
x	x	x	x	1	x	x	x	Terminal risk management was performed
x	x	x	x	x	1	x	x	Script processing was performed
x	x	x	x	x	x	0	x	RFU
x	x	x	x	x	x	x	0	RFU

TSI Byte 2 (Rightmost):

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	x	x	x	x	x	x	x	RFU
x	0	x	x	x	x	x	x	RFU
x	x	0	x	x	x	x	x	RFU
x	x	x	0	x	x	x	x	RFU
x	x	x	x	0	x	x	x	RFU
x	x	x	x	x	0	x	x	RFU
x	x	x	x	x	x	0	x	RFU
x	x	x	x	x	x	x	0	RFU

Table 47: Transaction Status Information

C7 Biometric Information Template (BIT)

The BIT provides descriptive information regarding the biometric formats and solutions supported in a card or in a terminal.

The format of the BIT is defined in ISO/IEC 19785-3 with TLV-encoded patron format, as shown in Table 48.

The Biometric Solution ID is present to ensure that a biometric card certified by a program will only perform biometric verification on the terminal that is certified by the same program, and a terminal certified by a program will only perform biometric CVM on the biometric card that is certified by the same program. The terminals and cards certified by the same program shall contain the same Biometric Solution ID. Such program can be global or regional.

If the Biometric Subtype is not applicable to a Biometric Type or biometric solution, the Biometric Subtype shall be set to all binary zeros.

The biometric reference template stored in the card may consist of a template that is completely in standardised format, completely in proprietary format, or a combination of partially standardised format and partially proprietary format. If the template is partially in standardised format and partially in proprietary format, then the level 2 Biometric Header Template (BHT) 1 and BHT 2, as shown in Table 48, shall be present within the BIT in the card; otherwise the level 2 BHT 1 and BHT 2 shall not be present within the BIT in the card.

If the BIT in the Card BIT Group Template contains level 2 BHT 1 and BHT 2, the terminal compares the terminal BIT tags '87' and '88' against the level 2 BHT 1 tags '87' and '88'.

Note that if the level 2 BHT 2 on the card is needed for complementary matching by the terminal, then this is based on proprietary data. This proprietary data may be stored in the terminal in a proprietary way, but it is not stored in the terminal template. Ideally, BHT 2 should be used to prioritise different solutions, and not to determine whether there is a match or not. However, this is proprietary.

The BIT in a terminal shall not contain the level 2 BHT 1 nor BHT 2. A single BIT in the terminal presents only one format owner and one format type.

<table>
<tr><th>Tag</th><th>L</th><th colspan="3">Value</th><th>Presence</th></tr>
<tr><td>'A1'</td><td>var.</td><td colspan="3">Biometric Header Template (BHT) in compliance with CBEFF. The contents within this template are TLV-encoded and may appear in any order within the template.</td><td>Mandatory</td></tr>
<tr><td></td><td></td><td>Tag</td><td>L</td><td>Value</td><td></td></tr>
<tr><td></td><td></td><td>'80'</td><td>2</td><td>Patron header version (default '0101')</td><td>Mandatory</td></tr>
<tr><td></td><td></td><td>'90'</td><td>var.</td><td>Biometric Solution ID, a unique identifier used for referencing this biometric data set in an application context outside the card</td><td>Mandatory</td></tr>
<tr><td></td><td></td><td>'81'</td><td>1-3</td><td>Biometric Type, as shown in Table 49</td><td>Mandatory</td></tr>
<tr><td></td><td></td><td>'82'</td><td>1</td><td>Biometric Subtype, as shown in Table 50</td><td>Mandatory</td></tr>
<tr><td></td><td></td><td>'83'</td><td>7</td><td>Creation date and time of the reference template (CCYYMMDDhhmmss)</td><td>Optional</td></tr>
<tr><td></td><td></td><td>'84'</td><td>var.</td><td>Creator</td><td>Optional</td></tr>
<tr><td></td><td></td><td>'85'</td><td>8</td><td>Validity period (from CCYYMMDD, to CCYYMMDD)</td><td>Optional</td></tr>
<tr><td></td><td></td><td>'86'</td><td>2</td><td>Identifier of product (PID) that created the reference template, value assigned by IBIA, see www.ibia.org</td><td>Optional</td></tr>
<tr><td></td><td></td><td>'91' or 'B1'</td><td>var.</td><td>Biometric matching algorithm parameters</td><td>Conditional
Only present if it is in the Card BIT Group Template and required by the standard or the specification which defines the format of the BDB[28]</td></tr>
</table>

Table 48: Biometric Information Template (BIT)

Tag	L	Value					Presence
		'87'	2	Format owner, value assigned by IBIA, see www.ibia.org			Conditional Only present if template contains no level 2 BHT 1 and BHT 2
		'88'	2	Format type, specified by format owner			Conditional Only present if template contains no level 2 BHT 1 and BHT 2
		'A1'	var.	BHT 1 (level 2). The contents within this template are TLV-encoded and may appear in any order within the template.			Optional
				Tag	**L**	**Value**	
				'87'	2	Format owner, e.g. format owner identifier of ISO/IEC JTC1/SC37	Mandatory
				'88'	2	Format type, specified by format owner	Mandatory
		'A2'	var.	BHT 2 (level 2). The contents within this template are TLV-encoded and may appear in any order within the template.			Optional

Table 48: Biometric Information Template (BIT), continued

[28] For example, if the format of BDB is compliant with ISO/IEC 19794-2, the biometric matching algorithm parameters, including the maximum number of minutiae and the minutiae ordering scheme, shall be present as defined in ISO/IEC 19794-2.

Tag	L	Value			Presence
		Tag	L	Value	
		'87'	2	Format owner, e.g. a card manufacturer	Mandatory
		'88'	2	Format type, specified by format owner	Mandatory

Table 48: Biometric Information Template (BIT), continued

Note: The data object tags used within the Biometric Header Template (tag 'A1') only have the specified meaning within template 'A1'. For example, tag '90' is the Biometric Solution ID only within template 'A1', and tag '90' is the Issuer Public Key Certificate outside of template 'A1'.

Name of Biometric Type	Value
Facial	'02'
Finger[29]	'08'
Iris	'10'
Palm	'020000'
Voice	'04'

Table 49: Biometric Type

[29] The Biometric Type "Finger" is used for both fingerprint and finger vein. The Format Type (tag '88') contained in BIT (tag '7F60') then identifies whether it represents fingerprint or finger vein.

b8	b7	b6	b5	b4	b3	b2	b1	Biometric Subtype
0	0	0	0	0	0	0	0	No information given
x	x	1	x	x	x	0	1	Right
x	x	1	x	x	x	1	0	Left
x	0	1	0	0	0	x	x	No meaning
x	0	1	0	0	1	x	x	Thumb
x	0	1	0	1	0	x	x	Index finger
x	0	1	0	1	1	x	x	Middle finger
x	0	1	1	0	0	x	x	Ring finger
x	0	1	1	0	1	x	x	Little finger
x	1	1	0	0	1	x	x	Palm
x	1	1	0	1	0	x	x	Back of hand
x	1	1	0	1	1	x	x	Wrist
All other values								RFU

Table 50: Biometric Subtype

C8 BIT Group Template

The BIT Group Template contains one or more BITs. There are two BIT Group Templates defined in this specification: Card BIT Group Template and Terminal BIT Group Template.

The Card BIT Group Template is provided by the card in records that are read by the terminal using the READ RECORD command prior to a biometric verification process.

The sequence of BITs in a Card BIT Group Template identifies the sequence in which different Biometric Subtypes shall be requested for verification.

The Terminal BIT Group Template contains one or more BITs, which are used to identify matching BITs supported by both the card and terminal.

The Card BIT Group Template is structured as shown in Table 51, and the Terminal BIT Group Template is structured as shown in Table 52.

Tag	L	Value			
'9F31'	var.	Card BIT Group Template			
		Tag	**L**	**Value**	
		'BF4A'	var.	Offline BIT Group Template	
				'02'	1	Number of BITs in the group
				'7F60'	var.	BIT 1, see Table 48
				'7F60'	var.	BIT 2, see Table 48
				…	…	…
				'7F60'	var.	BIT n, see Table 48
		'BF4B'	var.	Online BIT Group Template	
				'02'	1	Number of BITs in the group
				'7F60'	var.	BIT 1, see Table 48
				'7F60'	var.	BIT 2, see Table 48
				…	…	…
				'7F60'	var.	BIT n, see Table 48

Table 51: Card BIT Group Template

Tag	L	Value
'02'	1	Number of BITs in the group
'7F60'	var.	BIT 1, see Table 48
'7F60'	var.	BIT 2, see Table 48
…	…	…
'7F60'	var.	BIT n, see Table 48

Table 52: Terminal BIT Group Template

Annex D Transaction Log Information

D1 Purpose

Provide support for accessing a transaction log file by special devices.

D2 Conditions of Execution

This optional function is intended to be executed by special devices.

D3 Sequence of Execution

This function may be executed after Application Selection.

D4 Description

To get the Transaction Log information, the two following data elements are used: Log Entry and Log Format.

Table 53 describes the format of the Log Entry data element (tag '9F4D'):

Byte	Format	Length	Value
1	b	1	SFI containing the cyclic transaction log file
2	b	1	Maximum number of records in the transaction log file

Table 53: Log Entry

Devices that read the transaction log use the Log Entry data element to determine the location (SFI) and the maximum number of transaction log records.

The SFI shall be in the range 11 to 30.

The Transaction Log records shall be accessible using the READ RECORD command as specified in section 6.5.11. The file is a cyclic file as defined in ISO/IEC 7816-4. Record #1 is the most recent transaction. Record #2 is the next prior transaction, etc.

The Transaction Log records shall not be designated in the Application File Locator. Each record is a concatenation of the values identified in the Log Format data element. The records in the file shall not contain the Application Elementary File (AEF) Data Template (tag '70').

The Log Format and the Transaction Log records shall remain accessible when the application is blocked.

To read the transaction log information, the special device uses the following steps:

- Perform Application Selection and retrieve the Log Entry data element located in the FCI Issuer Discretionary Data. If the Log Entry data element is not present, the application does not support the Transaction Log function.
- Issue a GET DATA command to retrieve the Log Format data element.
- Issue READ RECORD commands to read the Transaction Log records.

D5 Example

Note that the following data elements are shown for example purposes only.

A Log Entry data element equal to '0F14' indicates that the transaction log file is located in SFI 15 ('0F') and contains a maximum of 20 records ('14').

A Log Format data element equal to '9A039F21035F2A029F02069F4E149F3602' indicates that the transaction log records have the following content:

Data Content	Tag	Length
Transaction Date	'9A'	3
Transaction Time	'9F21'	3
Transaction Currency Code	'5F2A'	2
Amount, Authorised	'9F02'	6
Merchant Name and Location	'9F4E'	20
Application Transaction Counter	'9F36'	2

Table 54: Example of Log Format

In Table 54, lengths and tags are shown for clarity. They do not appear in the log record which is the concatenation of values (no TLV coding).

Data elements listed in the Log Format may come from the terminal and the card. Terminal data elements such as Merchant Name and Location might have been passed to the card in the PDOL or CDOL data.

Annex E TVR and TSI Bit Settings Following Script Processing

Four possible scenarios can occur when processing a script. These scenarios are described below, together with the expected results in terms of the setting of the appropriate TVR bits, the TSI bit, and the Issuer Script Results.

In the following descriptions:

- "TVR bits" refers to TVR byte 5 bit 6 and bit 5 (depending on whether it is a tag '71' and/or tag '72' script) as defined in Table 46.
- "TSI bit" refers to TSI byte 1 bit 3 as defined in Table 47.

The Issuer Script Results are defined in Book 4 section A5.

E1 Scenarios

Scenario 1

A script is received, it parses correctly, the commands are sent to the card, and the card returns '9000', '62xx', or '63xx' to all commands received.

In this scenario the terminal:

- shall set the TSI bit
- shall not set the TVR bits
- shall set the first byte of the Issuer Script Results to '2x', 'Script processing successful'.

Scenario 2

A script is received, it parses correctly, the commands are sent to the card, but the card does not return '9000', '62xx', or '63xx' to one of the commands received.

In this scenario the terminal:

- shall set the TSI bit
- shall set the appropriate TVR bit(s)
- shall set the first byte of the Issuer Script Results to '1x', 'Script processing failed'
- shall send no further commands from that script to the card, even if they exist.

Scenario 3

A script is received, it does not parse correctly, and so no commands are sent to the card.

In this scenario the terminal:

- shall set the TSI bit
- shall set the appropriate TVR bit(s)
- shall set the first byte of the Issuer Script Results to '00', 'Script not performed'.

Scenario 4

No script is received. In this scenario the terminal shall set neither the TSI bit nor the TVR bit(s).

In this event there will be no Issuer Script Results.

E2 Additional Information

It is possible, but not recommended, that commands may be sent to the card 'on the fly' as a script is parsed. In this event:

- If a parsing error occurs before any commands are sent to the card, the terminal shall set the first byte of the Issuer Script Results to '00' and shall set the appropriate TVR bits and the TSI bit.
- If a parsing error occurs after any command has been sent to the card, the terminal shall set the first byte of the Issuer Script Results to '1x', and shall set the appropriate TVR bits and the TSI bit.

If more than one script is received, the terminal:

- shall set the TSI bit
- shall set the TVR bit(s) (as described in Scenarios 2 and 3) if any error occurs
- shall set the Issuer Script Results as described in Scenarios 1 through 3 for each script on a script-by-script basis
- shall process all Issuer scripts

Annex F Status Words Returned in EXTERNAL AUTHENTICATE

The terminal shall issue an EXTERNAL AUTHENTICATE command to the card only if the card indicates in byte 1 bit 3 of the AIP that it supports issuer authentication using the EXTERNAL AUTHENTICATE command.

The terminal shall issue only one EXTERNAL AUTHENTICATE command to the card during a transaction. As stated in section 10.9, there is a complementary card requirement to this which states that the card shall return status '6985', 'Command Not Supported', to the second and any subsequent EXTERNAL AUTHENTICATE commands received during the transaction.

Table 55 explains various status values the terminal may receive in response to the (first) EXTERNAL AUTHENTICATE command issued to the card, and the action the terminal shall take as a result.

Status	Explanation	Terminal Action
'9000'	Issuer authentication was successful.	The terminal shall continue with the transaction.
'6300' or any other status except '6985' and '9000'	Issuer authentication failed.	The terminal shall set the 'Issuer authentication failed' bit in the TVR to 1, and continue with the transaction.
'6985'	Issuer authentication failed and the card is in an error state (it has indicated in the AIP that it supports EXTERNAL AUTHENTICATE, but in the status returned that it does not).	This condition should never occur; in the event that it does, the behaviour of the terminal is indeterminate and it shall either terminate the transaction OR set the 'Issuer authentication failed' bit in the TVR to 1, and continue with the transaction.

Table 55: Terminal Action after (First) EXTERNAL AUTHENTICATE Response

Annex G Account Type

Value	Account Type
00	Default – unspecified
10	Savings
20	Cheque/debit
30	Credit
All other values RFU	

Table 56: Account Type

Part V

Common Core Definitions

Common Core Definitions

This Part describes an optional extension to this Book, to be used when implementing the Common Core Definitions (CCD).

These Common Core Definitions specify a minimum common set of card application implementation options, card application behaviours, and data element definitions sufficient to accomplish an EMV transaction. Terminals certified to be compliant with the existing EMV Specifications will, without change, accept cards implemented according to the Common Core Definitions, since the Common Core Definitions are supported within the existing EMV requirements.

To be compliant with the Common Core Definitions, an implementation shall implement all the additional requirements in the Common Core Definitions Parts of all affected Books.

Changed and Added Sections

Each section heading below refers to the section in this Book to which the additional requirements apply, or introduces new sections where required. The text defines requirements for a common core implementation, in addition to the requirements already specified in the referenced section of EMV.

Part II – Data Elements and Commands

6 Commands for Financial Transaction

6.2 Response APDU Format

For the following commands used during transaction processing, the body of the response APDU is a constructed data object with tag equal to '77' of which the value field may contain one or more BER-TLV coded data objects.

- GENERATE AC
- GET PROCESSING OPTIONS
- INTERNAL AUTHENTICATE

Tag	Value
'77'	Response Message Template Format 2

Table CCD 1: Body of Response APDU Structure

6.5 Commands

6.5.4 EXTERNAL AUTHENTICATE Command-Response APDUs

6.5.4.1 Definition and Scope

The CCD-compliant application shall support issuer authentication using the second GENERATE AC command. The CCD-compliant application shall indicate that the EXTERNAL AUTHENTICATE command is not supported in EMV applications by setting bit 3 in byte 1 of the AIP to 0.

6.5.5 GENERATE APPLICATION CRYPTOGRAM Command-Response APDUs

6.5.5.1 Definition and Scope

The CCD-compliant application shall support issuer authentication using the second GENERATE AC command.

6.5.5.3 Data Field Sent in the Command Message

CDOL2 shall include tag '8A' (Authorisation Response Code) and tag '91' (Issuer Authentication Data).

6.5.5.4 Data Field Returned in the Response Message

The response message shall be a BER-TLV coded constructed data object introduced by tag '77' and contains only the data shown in Table CCD 2.

Tag	Value
'9F27'	Cryptogram Information Data
'9F36'	Application Transaction Counter
'9F26'	Application Cryptogram
'9F10'	Issuer Application Data

Table CCD 2: Format 2 GENERATE AC Response Message Data Field

The required data elements for the response returned in an envelope as specified for the CDA feature (described in Book 2 section 6.6) are shown in Book 2 Table CCD 1 and Table CCD 2.

The Cryptogram Information Data returned in the GENERATE AC response message is coded according to Table CCD 3:

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0							AAC
0	1							TC
1	0							ARQC
1	1							RFU
		0	0					Payment System-specific cryptogram
				0				No advice required
					0	0	0	No information given

Table CCD 3: Coding of Cryptogram Information Data

6.5.8 GET PROCESSING OPTIONS Command-Response APDUs

6.5.8.2 Command Message

The CCD-compliant application shall not preclude support for PDOL.

6.5.8.4 Data Field Returned in the Response Message

The response message shall be a BER-TLV coded constructed data object introduced by tag '77' and contains only the data shown in Table CCD 4.

Tag	Value
'82'	Application Interchange Profile
'94'	Application File Locator

Table CCD 4: Format 2 GET PROCESSING OPTIONS Response Message Data Field

6.5.9 INTERNAL AUTHENTICATE Command-Response APDUs

6.5.9.4 Data Field Returned in the Response Message

The response message shall be a BER-TLV coded constructed data object introduced by tag '77' and contains only the data shown in Table CCD 5.

Tag	Value
'9F4B'	Signed Dynamic Application Data

Table CCD 5: Format 2 Internal Authenticate Response Message Data Field

6.5.12.2 Command Message

To allow an issuer to use offline plaintext PIN verification as a possible CVM, a CCD-compliant card shall support the VERIFY command with parameter P2 = '80' as defined in Book 3, Table 24.

Part III – Debit and Credit Application Specification

7 Files for Financial Transaction Interchange

7.3 Data Retrievable by GET DATA Command

The ICC shall support the GET DATA command for retrieval of the primitive data object with tag '9F17' (PIN Try Counter).

9 GENERATE AC Command Coding

9.2 Command Data

9.2.2 Transaction Certificate Data

The CCD-compliant application shall not contain a TDOL. The CCD-compliant application shall not request the terminal to generate a TC Hash Value (that is, tag '98' shall not be included in CDOL1 or CDOL2).

The following Section 9.2.3 applies to a CCD-compliant application.

9.2.3 Common Core Definitions Card Verification Results

In response to the GENERATE AC command and as part of the Issuer Application Data, the CCD-compliant application shall return the Card Verification Results (CVR). The CVR includes information for the issuer regarding the results of Card Risk Management processing and application processing. The format of the CVR for a CCD-compliant application is specified in CCD section C9.3.

9.2.3.1 Options Related to Setting/Resetting of Counters and Indicators

The issuer shall have the option of specifying whether a new card is required to set the 'Go Online on Next Transaction Was Set' bit.

The issuer shall have the option of specifying whether the CCD-compliant application requires issuer authentication to be performed for the application to approve (TC) an online transaction.

The issuer shall have the option of specifying whether the CCD-compliant application requires issuer authentication to pass when performed for the application to approve (TC) an online transaction.

If the CCD-compliant application does not require issuer authentication to be performed for the application to approve (TC) an online transaction, the issuer shall have the option of specifying whether the CCD-compliant application requires issuer authentication to pass for resetting all the following non-velocity-checking indicators:

- Issuer Authentication Failed
- Last Online Transaction Not Completed
- Issuer Script Processing Failed
- Go Online on Next Transaction Was Set

If the CCD-compliant application does not require issuer authentication to pass when performed for the application to approve (TC) an online transaction, the issuer shall have the option of specifying whether the CCD-compliant application requires issuer authentication to pass for resetting all the following non-velocity-checking indicators:

- Last Online Transaction Not Completed
- Issuer Script Processing Failed

- Go Online on Next Transaction Was Set

If the CCD-compliant application does not require issuer authentication to be performed or does not require issuer authentication to pass when performed for the application to approve (TC) an online transaction, the issuer shall have the option of specifying whether the CCD-compliant application requires issuer authentication to pass for resetting the velocity-checking offline transaction count(s) and cumulative amount(s).

The issuer shall have the option of indicating whether the application shall use the 'Update Counters' bits in the CSU to update the velocity-checking count(s) and cumulative amount(s) associated with the offline transaction limits referred to in bits b8 – b5 of byte 3 of the CVR, if the 'CSU Created by Proxy for the Issuer' bit in the CSU is set to 1.

If the 'CSU Created by Proxy for the Issuer' bit is set to 1 in the CSU, and if the issuer specifies that 'Update Counters' shall not be used, then the issuer shall have the option of indicating whether the application:

- shall not update the offline counters
- shall set the offline counters to zero
- shall set the offline counters to the upper offline limits
- shall add the transaction to the offline counter(s)

9.2.3.2 Setting and Resetting of Bits in the CVR

The following describes the conditions under which each bit in the CVR of a Common Core Definitions card is set or reset.

Application Cryptogram Type Returned in Second GENERATE AC

In the first GENERATE AC response, these bits shall be set to Second GENERATE AC Not Requested.

In the second GENERATE AC response, these bits shall be set to the value of bits b8 – b7 of the Cryptogram Information Data returned in the response to the second GENERATE AC command of the current transaction (AAC or TC).

Application Cryptogram Type Returned in First GENERATE AC

In both the first and second GENERATE AC response, these bits shall be set to the value of bits b8 – b7 of the Cryptogram Information Data returned in the response to the first GENERATE AC command of the current transaction (AAC, TC, or ARQC).

CDA Performed

In the first GENERATE AC response, this bit shall be set if and only if Signed Dynamic Application Data is returned in the response to the first GENERATE AC command of the current transaction.

In the second GENERATE AC response, this bit shall be set if and only if Signed Dynamic Application Data is returned in the response to the first or second GENERATE AC command (or both) of the current transaction.

Offline DDA Performed

In both the first and second GENERATE AC response, this bit shall be set if and only if Signed Dynamic Application Data is returned in the response to the INTERNAL AUTHENTICATE command of the current transaction.

Issuer Authentication Not Performed

In the second GENERATE AC response, this bit shall be set if and only if the CCD-compliant application did not receive Issuer Authentication Data. This may be the case either if the transaction was unable to go online or if the issuer did not provide Issuer Authentication Data in the response message.

In the first GENERATE AC response, this bit shall be set to the value it had in the most recent second GENERATE AC response sent by the CCD-compliant application.

Issuer Authentication Failed

This bit shall be set in the GENERATE AC response if and only if issuer authentication was performed and failed. In the first GENERATE AC response, it indicates issuer authentication failed in a previous online transaction. In the second GENERATE AC response, it indicates either that issuer authentication failed in the current transaction, or that issuer authentication failed in a previous transaction and the bit was not reset.

Once set, this bit shall remain set until either:

- issuer authentication is successful,
- or all of the following conditions are true:
  - the transaction successfully went online (that is, the Authorisation Response Code does not indicate that the terminal was unable to go online),
  - issuer authentication was not performed,
  - the CCD-compliant application does not require issuer authentication to be performed for the application to approve (TC) an online transaction, and
  - the CCD-compliant application does not require issuer authentication to pass for resetting of non-velocity-checking indicators.

Low Order Nibble of PIN Try Counter

In both the first and second GENERATE AC response, these bits shall be set to the value of the low-order nibble (bits b4 – b1) of the PIN Try Counter.

Offline PIN Verification Performed

In both the first and second GENERATE AC response, this bit shall be set if and only if Offline PIN Verification has been performed (successfully or unsuccessfully) on the current transaction.

Offline PIN Verification Performed and PIN Not Successfully Verified

In both the first and second GENERATE AC response, this bit shall be set if and only if Offline PIN Verification has been performed on the current transaction and the PIN was not successfully verified during processing of the current transaction.

PIN Try Limit Exceeded

In both the first and second GENERATE AC response, this bit shall be set if and only if the PIN Try Counter is zero.

Last Online Transaction Not Completed

This bit shall be set in the first GENERATE AC response if and only if in the previous transaction, the CCD-compliant application requested to go online and the transaction was not completed (that is, the second GENERATE AC command was not received).

Once set, this bit shall remain set until either:

- issuer authentication is successful,
- or all of the following conditions are true:
  - the transaction successfully went online (that is, the Authorisation Response Code does not indicate that the terminal was unable to go online),
  - issuer authentication was not performed,
  - the CCD-compliant application does not require issuer authentication to be performed for the application to approve (TC) an online transaction, and
  - the CCD-compliant application does not require issuer authentication to pass for resetting of non-velocity-checking indicators.
- or all of the following conditions are true:
  - the transaction successfully went online (that is, the Authorisation Response Code does not indicate that the terminal was unable to go online),
  - issuer authentication was performed and failed,
  - the CCD-compliant application does not require issuer authentication to pass when performed for the application to approve (TC) an online transaction, and
  - the CCD-compliant application does not require issuer authentication to pass for resetting of non-velocity-checking indicators.

Lower Offline Transaction Count Limit Exceeded

In both the first and second GENERATE AC response, this bit shall be set if the CCD-compliant application has exceeded an issuer-specified lower limit for the number of transactions approved offline. This bit may represent the condition of multiple counters. At the least, all transactions approved offline whose amounts were not cumulated shall be included in at least one transaction count. This bit may also be set under additional conditions specified by the issuer.

Upper Offline Transaction Count Limit Exceeded

In both the first and second GENERATE AC response, this bit shall be set if the CCD-compliant application has exceeded an issuer-specified upper limit for the number of transactions approved offline. This bit may represent the condition of multiple counters. At the least, all transactions approved offline whose amounts were not cumulated shall be included in at least one transaction count. This bit may also be set under additional conditions specified by the issuer.

Lower Cumulative Offline Amount Limit Exceeded

In both the first and second GENERATE AC response, this bit shall be set if the CCD-compliant application has exceeded an issuer-specified lower limit for cumulative amounts approved offline. This bit may represent the condition of multiple counters. At the least, all domestic transactions approved offline shall be included in at least one cumulative amount. This bit may also be set under additional conditions specified by the issuer.

Upper Cumulative Offline Amount Limit Exceeded

In both the first and second GENERATE AC response, this bit shall be set if the CCD-compliant application has exceeded an issuer-specified upper limit for cumulative amounts approved offline. This bit may represent the condition of multiple counters. At the least, all domestic transactions approved offline shall be included in at least one cumulative amount. This bit may also be set under additional conditions specified by the issuer.

Issuer-discretionary bit 1 – Issuer-discretionary bit 4:

These bits are set in the first and second GENERATE AC response at the discretion of the issuer. The meaning of these bits is defined by the issuer and is outside the scope of this specification.

Number of Successfully Processed Issuer Script Commands Containing Secure Messaging

In the first and second GENERATE AC response, these bits shall be set to the number of commands successfully processed with secure messaging.

Issuer Script Processing Failed

In both the first and second GENERATE AC response, this bit shall be set if and only if processing of a command with secure messaging failed.

Once set, this bit shall remain set until a subsequent GENERATE AC command where either:

- issuer authentication is successful,
- or all of the following conditions are true:
  - the transaction successfully went online (that is, the Authorisation Response Code does not indicate that the terminal was unable to go online),
  - issuer authentication was not performed
  - the CCD-compliant application does not require issuer authentication to be performed for the application to approve (TC) an online transaction, and
  - the CCD-compliant application does not require issuer authentication to pass for resetting of non-velocity-checking indicators.
- or all of the following conditions are true:
  - the transaction successfully went online (that is, the Authorisation Response Code does not indicate that the terminal was unable to go online),
  - issuer authentication was performed and failed,
  - the CCD-compliant application does not require issuer authentication to pass when performed for the application to approve (TC) an online transaction, and
  - the CCD-compliant application does not require issuer authentication to pass for resetting of non-velocity-checking indicators.

Offline Data Authentication Failed on Previous Transaction

In both the first and second GENERATE AC response, this bit shall be set if and only if, in the TVR returned during the previous transaction, any of the following bits was set:

- SDA Failed
- DDA Failed
- CDA Failed

Once set, this bit shall remain set until a subsequent transaction is performed that meets either of the following conditions:

- the previous transaction successfully went online, or
- the previous transaction was approved offline.

If either condition is met the bit is reset in the first GENERATE AC response.

Go Online on Next Transaction Was Set

In both the first and second GENERATE AC response, this bit shall be set if and only if the 'Set Go Online on Next Transaction' bit of the last successfully recovered CSU was set, or it is a new card and the issuer has specified that a new card is required to set the 'Go Online on Next Transaction Was Set' bit.

Once set, this bit shall remain set until a subsequent GENERATE AC command where either:

- all of the following conditions are true:
  - issuer authentication is successful, and
  - the Set Go Online on Next Transaction bit of the CSU is not set.
- or all of the following conditions are true:
  - the transaction successfully went online (that is, the Authorisation Response Code does not indicate that the terminal was unable to go online),
  - issuer authentication was not performed,
  - the CCD-compliant application does not require issuer authentication to be performed for the application to approve (TC) an online transaction, and
  - the CCD-compliant application does not require issuer authentication to pass for resetting of non-velocity-checking indicators.
- or all of the following conditions are true:
  - the transaction successfully went online (that is, the Authorisation Response Code does not indicate that the terminal was unable to go online),
  - issuer authentication was performed and failed,
  - the CCD-compliant application does not require issuer authentication to pass when performed for the application to approve (TC) an online transaction, and
  - the CCD-compliant application does not require issuer authentication for resetting of non-velocity-checking indicators.

Unable to go Online

This bit shall be set in the second GENERATE AC response if and only if the Authorisation Response Code, tag '8A', returned from the terminal indicates the terminal was unable to go online (set to 'Y3' or 'Z3').

9.2.3.3 Mandatory Actions Due to CVR Bit Settings

This section provides a list of mandatory actions that shall be taken by the CCD-compliant application, and issuer-configurable options that shall be supported by the CCD-compliant application.

Issuer Authentication Not Performed

The issuer shall have the option of specifying that if this bit is set, whether the CCD-compliant application shall:

- force transactions at online-capable terminals to go online,
- or allow the transaction to remain offline.

The issuer shall have the option of specifying that if this bit is set, and either the transaction occurs at an offline-only terminal or the terminal is unable to go online, whether the CCD-compliant application shall:

- be allowed to approve (TC) the transaction, or
- decline the transaction.

Issuer Authentication Failed

The issuer shall have the option of specifying that if this bit is set, whether the CCD-compliant application shall:

- force transactions at online-capable terminals to go online,
- or allow the transaction to remain offline.

The issuer shall have the option of specifying that if this bit is set, and either the transaction occurs at an offline-only terminal or the terminal is unable to go online, whether the CCD-compliant application shall:

- be allowed to approve (TC) the transaction, or
- decline the transaction.

PIN Try Limit Exceeded

The issuer shall have the option of specifying that if this bit is set, the CCD-compliant application shall decline the transaction offline.

The issuer shall have the option of specifying that if this bit is set, whether the CCD-compliant application shall:

- force transactions at online-capable terminals to go online,
- or allow the transaction to remain offline.

The issuer shall have the option of specifying that if this bit is set, and either the transaction occurs at an offline-only terminal or the terminal is unable to go online, whether the CCD-compliant application shall:

- be allowed to approve (TC) the transaction, or
- decline the transaction.

The ICC shall not block the application or the card due to this bit being set.

Last Online Transaction Not Completed

If this bit is set, the CCD-compliant application shall force the transaction at online-capable terminals to go online.

The issuer shall have the option of specifying that if this bit is set, and either the transaction occurs at an offline-only terminal or the terminal is unable to go online, whether the CCD-compliant application shall:

- be allowed to approve (TC) the transaction, or
- decline the transaction.

Lower Offline Transaction Count Limit Exceeded

If this bit is set in the first GENERATE AC response, the CCD-compliant application shall force the transaction at online-capable terminals to go online.

Upper Offline Transaction Count Limit Exceeded

If this bit is set and either the transaction occurs at an offline-only terminal or the terminal is unable to go online, the CCD-compliant application shall decline the transaction. However, the issuer shall have the option of allowing the CCD-compliant application to override this decline for a transaction at Terminal Type 26.

Lower Cumulative Offline Amount Limit Exceeded

If this bit is set in the first GENERATE AC response, the CCD-compliant application shall force the transaction at online-capable terminals to go online.

Upper Cumulative Offline Amount Limit Exceeded

If this bit is set and either the transaction occurs at an offline-only terminal or the terminal is unable to go online, the CCD-compliant application shall decline the transaction. However, the issuer shall have the option of allowing the CCD-compliant application to override this decline for a transaction at Terminal Type 26.

Issuer Script Processing Failed

The issuer shall have the option of specifying that if this bit is set, whether the CCD-compliant application shall:

- force transactions at online-capable terminals to go online,
- or allow the transaction to remain offline.

Go Online on Next Transaction Was Set

The issuer shall have the option of specifying that if this bit is set, and either the transaction occurs at an offline-only terminal or the terminal is unable to go online, whether the CCD-compliant application shall:

- be allowed to approve (TC) the transaction, or
- decline the transaction.

9.3 Command Use

The CCD-compliant application shall respond to the first GENERATE AC with any of the following cryptogram types:

- TC
- ARQC
- AAC

The CCD-compliant application shall respond to the second GENERATE AC (if applicable) with either of the following cryptogram types:

- TC
- AAC

10 Functions Used in Transaction Processing

10.5 Cardholder Verification

The CCD-compliant application shall support Cardholder Verification. It shall indicate this by setting the Application Interchange Profile byte 1, bit 5 to 1.

10.5.1 Offline PIN Processing

The CCD-compliant application shall be capable of supporting offline plaintext PIN verification. It is the issuer's option whether or not to use offline plaintext PIN as a cardholder verification method.

10.8 Card Action Analysis

The CCD-compliant application shall not request that the terminal send an advice message to the issuer.

10.8.1 Terminal Messages for an AAC

The CCD-compliant application shall set bits b3 – b1 of the CID to 000 in the GENERATE AC command response.

10.8.2 Advice Messages

The CCD-compliant application shall not request the terminal to send an advice message. Bit b4 of the Cryptogram Information Data shall be set to 0.

10.10 Issuer-to-Card Script Processing

An issuer shall send no more than one issuer script template in an authorisation response message. The script template may contain multiple commands. The script template may be tag '71' or tag '72'.

10.11 Completion

The following Section 10.11.1 applies to a CCD-compliant application.

10.11.1 Additional Completion Actions for a CCD-Compliant Application

10.11.1.1 Actions Taken by CCD-compliant Application After Issuer Authentication is Successful

After issuer authentication is successful, if the 'CSU Created by Proxy for the Issuer' bit in the CSU is set to 1, and if the issuer specifies that 'Update Counters' shall not be used, then the following shall govern the behaviour of velocity-checking counters and cumulative amounts associated with the offline transaction limits referred to in bits b8 – b5 of byte 3 of the CVR:

- If the issuer specifies that the application shall not update the offline counters, no offline counter or cumulative amount is modified.
- If the issuer specifies that the application shall set the offline counters to zero, the application will reset all the offline counters and cumulative amounts to zero. By doing so, the issuer allows the application to accept offline transactions, up to the offline limits.
- If the issuer specifies that the application shall set the offline counters to the upper offline limits, the offline counters and cumulative amounts will be set to their respective upper limits.
- If the issuer specifies that the application shall add the transaction to the offline counter(s), the transaction will be included in the offline counters or cumulative amounts as if it were an offline transaction.

This section describes the actions to be taken by the CCD-compliant application due to the setting of each bit in the CSU after issuer authentication is successful..

Issuer Approves Online Transaction

If 'Issuer Approves Online Transaction' is set and the terminal requests a TC, the application shall approve the transaction by returning a TC.

If 'Issuer Approves Online Transaction' is not set, the application shall decline the transaction by returning an AAC.

Card Block

If 'Card Block' is set, all applications in the ICC shall be permanently disabled, including applications that may be selected implicitly. For all subsequent SELECT commands the card shall return the status bytes 'Function not supported' (SW1-SW2 = '6A81') and perform no other action.

Application Block

If 'Application Block' is set, the currently selected application shall be invalidated. An invalidated application shall return the status bytes 'Selected file invalidated' (SW1-SW2 = '6283') in response to a SELECT command and return only an AAC in response to the GENERATE AC command.

Update PIN Try Counter

If 'Update PIN Try Counter' is set, the application shall update the PIN Try Counter (PTC) with the value contained in bits b4 – b1 of byte 1 of the CSU. If the PIN is blocked, updating the value of the PTC with a non-zero value unblocks the PIN. Updating the value of the PTC with a zero value blocks the PIN.

If 'Update PIN Try Counter' is not set, no update of the PTC shall be performed by the application.

The value contained in bits b4 – b1 of byte 1 of the CSU shall never be interpreted by the application.

Set Go Online on Next Transaction

If 'Set Go Online on Next Transaction' is set, the application shall force subsequent transactions at online-capable terminals to go online (that is, the CCD-compliant application shall return an ARQC in response to the first GENERATE AC command if a TC or an ARQC is requested). The application shall continue to try to go online at online-capable terminals until a subsequent GENERATE AC command where either:

- all of the following conditions are true:
  - issuer authentication is successful, and
  - the Set Go Online on Next Transaction bit of the CSU is not set.
- or all of the following conditions are true:
  - the transaction successfully went online (that is, the Authorisation Response Code does not indicate that the terminal was unable to go online),
  - issuer authentication was not performed,
  - the CCD-compliant application does not require issuer authentication to be performed for the application to approve (TC) an online transaction, and
  - the CCD-compliant application does not require issuer authentication to pass for resetting of non-velocity-checking indicators."
- or all of the following conditions are true:
  - the transaction successfully went online (that is, the Authorisation Response Code does not indicate that the terminal was unable to go online),
  - issuer authentication was performed and failed,
  - the CCD-compliant application does not require issuer authentication to pass when performed for the application to approve (TC) an online transaction, and
  - the CCD-compliant application does not require issuer authentication to pass for resetting of non-velocity-checking indicators.

Update Counters

'Update Counters' (bits b2 – b1 of byte 2 of the CSU) govern the behaviour of velocity-checking counters and cumulative amounts associated with the offline transaction limits referred to in bits b8 – b5 of byte 3 of the CVR if either of the following is true:

- the 'CSU Created by Proxy for the Issuer' bit in the CSU is set to 0
- the issuer specifies that the application shall use the 'Update Counters' bits in the CSU to update the velocity-checking count(s) and cumulative amount(s) regardless of the bit setting for 'CSU Created by Proxy for the Issuer'

If 'Update Counters' is set to 'Do Not Update Offline Counters', no offline counter or cumulative amount shall be modified.

If 'Update Counters' is set to 'Reset Offline Counters to Zero', the application shall reset all the offline counters and cumulative amounts to zero. By doing so, the issuer allows the application to accept offline transactions, up to the offline limits.

If 'Update Counters' is set to 'Set Offline Counters to Upper Offline Limits', the application shall set the offline counters and cumulative amounts to their respective upper limits.

If 'Update Counters' is set to 'Add Transaction to Offline Counters', the application shall include the transaction in the offline counters or cumulative amounts as if it were an offline transaction.

10.11.1.2 Other Completion Actions

After the transaction successfully went online (that is, the Authorisation Response Code does not indicate that the terminal was unable to go online), the CCD-compliant application shall reset to zero the velocity-checking offline transaction count(s) and cumulative offline amount(s) if either of the following are true:

- all of the following conditions are true:
  - the terminal requested a TC,
  - issuer authentication was not performed,
  - the CCD-compliant application does not require issuer authentication to be performed for the application to approve (TC) an online transaction, and
  - the CCD-compliant application does not require issuer authentication to pass for resetting of velocity-checking counters.
- or all of the following conditions are true:
  - the terminal requested a TC,
  - issuer authentication was performed and failed,
  - the CCD-compliant application does not require issuer authentication to pass when performed for the application to approve (TC) an online transaction, and
  - the CCD-compliant application does not require issuer authentication to pass for resetting of velocity-checking counters.

After the transaction successfully went online (that is, the Authorisation Response Code does not indicate that the terminal was unable to go online), the CCD-compliant application shall approve the transaction if all of the following conditions are true:

- the terminal requested a TC, and
- one of the following is true:
  - issuer authentication is successful and the 'Issuer Approves Online Transaction bit' of the recovered CSU is set to 1, or
  - issuer authentication was not performed and the CCD-compliant application does not require issuer authentication to be performed for the application to approve (TC) an online transaction, or
  - issuer authentication was performed and failed and the CCD-compliant application does not require issuer authentication to pass when performed for the application to approve (TC) an online transaction.

After the transaction successfully went online (that is, the Authorisation Response Code does not indicate that the terminal was unable to go online), the CCD-compliant application shall decline the transaction in the second GENERATE AC response if either of the following conditions are true:

- both of the following are true:
  - issuer authentication was not performed, and
  - the CCD-compliant application requires issuer authentication to be performed for the application to approve (TC) an online transaction,
- or both of the following are true:
  - issuer authentication was performed and failed, and
  - the CCD-compliant application requires issuer authentication to pass when performed for the application to approve (TC) an online transaction.

After the transaction did not successfully complete online (that is the Authorisation Response Code indicates that the terminal was unable to go online), the CCD-compliant application shall decide whether to approve or decline the transaction.

Part IV – Annexes

Annex A Data Elements Dictionary

For the data elements shown in Table CCD 6:

- If the source is 'Terminal', the data element shall not be included in any DOL used by a CCD-compliant application.
- If the source is 'ICC', the data element shall not be identified in the AFL of a CCD-compliant application.

Data Element Name	Tag	Source
Amount, Reference Currency	'9F3A'	Terminal
Application Reference Currency	'9F3B'	ICC
Application Reference Currency Exponent	'9F43'	ICC
Default Dynamic Data Authentication Data Object List (DDOL)	—	Terminal
Transaction Certificate (TC) Hash Value	'98'	Terminal

Table CCD 6: Data Elements Not Used by a CCD-Compliant Application

Table CCD 7 lists data elements (in addition to those defined in Annex A) that are defined within the context of the Common Core Definitions.

Name	Description	Source	For-mat	Tem-plate	Tag	Length
Card Verification Results (CVR)	Contains data sent to the issuer indicating exception conditions that occurred during the current and previous transactions. Transmitted to the terminal in Issuer Application Data as specified in Table CCD 9.	ICC	b	—	—	5
Common Core Identifier (CCI)	Data sent to the issuer identifying the format of the Issuer Application Data and the method for calculating the Application Cryptogram. Transmitted to the terminal in Issuer Application Data as specified in Table CCD 9. Contains the following: Format Code (FC) Cryptogram Version (CV)	ICC	b	—	—	1
Derivation Key Index (DKI)	Data sent to the issuer identifying the issuer's derivation key for deriving the card's ICC Master Keys. Transmitted to the terminal in Issuer Application Data as specified in Table CCD 9.	ICC	b	—	—	1
Issuer Discretionary Data	Contains issuer proprietary application data for transmission to the issuer in an online transaction. Transmitted to the terminal in Issuer Application Data as specified in Table CCD 9.	ICC	b	—	—	15

Table CCD 7: Additional Data Elements Defined for CCD

Annex C Coding of Data Elements Used in Transaction Processing

Please add the following sections after section C.8.

C9 Issuer Application Data for a CCD-Compliant Application

The CCD-compliant application shall have an Issuer Application Data (IAD) field of fixed length, 32 bytes long, with the following attributes:

- Byte 1 shall be set to '0F'.
- Byte 2 shall be the Common Core Identifier (CCI).
- Byte 17 shall be set to '0F'.

The CCD-compliant application shall support the selection of different Issuer Application Data if:

- the card requests the Terminal Type and the Additional Terminal Capabilities in PDOL, and
- the values provided by the terminal in the PDOL related data for the Terminal Type and the first two bytes of the Additional Terminal Capabilities are '34' and '0000' respectively.

C9.1 Common Core Identifier

The CCI shall identify the format of the IAD, and the Cryptogram Version (CV). Values in the range '00' to '9F' are reserved to avoid conflict with legacy Cryptogram Version Numbers.

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x	x	x					Common Core IAD Format Code (FC).
1	0	1	0					CCD Version 4.1 IAD Format (= 'A')
				x	x	x	x	Common Core Cryptogram Version (CV)
				0	1	0	1	CCD Version 4.1 Cryptogram Version (= '5' for Triple DES)
				0	1	1	0	CCD Version 4.1 Cryptogram Version (= '6' for AES)

Table CCD 8: Common Core Identifier

Bits b8 - b5 of the CCI shall indicate the Format Code (FC). Values in the range 'A' - 'F' shall indicate a CCD-specified IAD format (all values RFU by EMVCo for CCD).

Bits b4 – b1 of the CCI shall indicate the Cryptogram Version (CV) for the Application Cryptogram. The CV indicates:

- The cryptogram input data and key derivation method the CCD-compliant application uses to generate the Application Cryptogram.
- The cryptogram input data (including CSU coding), key derivation method, and ARPC method the CCD-compliant application expects the issuer to use when generating the Authorisation Response Cryptogram.

Values in the range '4' - 'F' shall indicate a CCD-specified cryptogram algorithm and data set (all values RFU by EMVCo for CCD). Values in the range '0' - '3' shall indicate a proprietary cryptogram algorithm. When using the CV range '0' - '3', applications are not CCD-compliant.

C9.2 Issuer Application Data for Format Code 'A'

The format and coding of the IAD with a Format Code of 'A' shall be as shown in Table CCD 9:

Byte(s)	Contents	Description
1	Length Indicator	Length of EMVCo-defined data in IAD. Set to '0F'.
2	CCI	Common Core Identifier
3	DKI	Derivation Key Index
4-8	CVR	Card Verification Results (see section C9.3)
9-16	Counters	Contents are at the discretion of the Payment System.
17	Length Indicator	Length of Issuer Discretionary Data field in IAD. Set to '0F'.
18-32	Issuer Discretionary Data	Contents are at the discretion of the issuer.

Table CCD 9: Issuer Application Data for Format Code 'A'

C9.3 Card Verification Results

The coding of the CVR for a Common Core IAD Format Code of value 'A' shall be as shown in Table CCD 10.

CVR Byte 1

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x							Application Cryptogram Type Returned in Second GENERATE AC
0	0							AAC
0	1							TC
1	0							Second GENERATE AC Not Requested
1	1							RFU
		x	x					Application Cryptogram Type Returned in First GENERATE AC
		0	0					AAC
		0	1					TC
		1	0					ARQC
		1	1					RFU
				1				CDA Performed
					1			Offline DDA Performed
						1		Issuer Authentication Not Performed
							1	Issuer Authentication Failed

CVR Byte 2

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x	x	x					Low Order Nibble of PIN Try Counter
				1				Offline PIN Verification Performed
					1			Offline PIN Verification Performed and PIN Not Successfully Verified
						1		PIN Try Limit Exceeded
							1	Last Online Transaction Not Completed

Table CCD 10: Card Verification Results for Format Code 'A'

CVR Byte 3

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1								Lower Offline Transaction Count Limit Exceeded
	1							Upper Offline Transaction Count Limit Exceeded
		1						Lower Cumulative Offline Amount Limit Exceeded
			1					Upper Cumulative Offline Amount Limit Exceeded
				1				Issuer-discretionary bit 1
					1			Issuer-discretionary bit 2
						1		Issuer-discretionary bit 3
							1	Issuer-discretionary bit 4

CVR Byte 4

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x	x	x					Number of Successfully Processed Issuer Script Commands Containing Secure Messaging
				1				Issuer Script Processing Failed
					1			Offline Data Authentication Failed on Previous Transaction
						1		Go Online on Next Transaction Was Set
							1	Unable to go Online

CVR Byte 5

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	0	0	0	0	0	RFU

Table CCD 10: Card Verification Results for Format Code 'A', continued

C10 Card Status Update for a CCD-Compliant Application

The Issuer Authentication Data shall include a Card Status Update (CSU) of fixed length, 4 bytes long. The coding of the CSU is shown in Table CCD 11.

CSU Byte 1

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1								Proprietary Authentication Data Included
	0	0	0					RFU
				x	x	x	x	PIN Try Counter

CSU Byte 2

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1								Issuer Approves Online Transaction
	1							Card Block
		1						Application Block
			1					Update PIN Try Counter
				1				Set Go Online on Next Transaction
					1			CSU Created by Proxy for the Issuer
						x	x	Update Counters
						0	0	Do Not Update Offline Counters
						0	1	Set Offline Counters to Upper Offline Limits
						1	0	Reset Offline Counters to Zero
						1	1	Add Transaction to Offline Counter

Note: The 'CSU Created by Proxy for the Issuer' bit shall be set in the CSU if and only if the CSU is generated by a proxy for the Issuer.

CSU Byte 3

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	0	0	0	0	0	RFU

Table CCD 11: Card Status Update for Cryptogram Versions '5' and '6'

CSU Byte 4

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x	x	x	x	x	x	x	Issuer-Discretionary

Table CCD 11: Card Status Update for Cryptogram Versions '5' and '6', continued

The default value for issuer-discretionary data in the CSU is zero.

Annex D Transaction Log Information

If the CCD-compliant application supports transaction logging, it shall be supported using the method specified in Annex D.

Index

A

B

C

D

E

F

G

I

L

M

N

O

P

R

S

T

U

V