Abecs Pinpad

Communication Protocol and Operation

Version: 2.12 (Apr 11th, 2019)

Version History

Version	Date	Author	Comments
2.12 rev01	Apr 11th, 2019	WFM (SETIS)	First English translation. rev01 - May 4th ,2021.

Contents

References

🕮**BibComp**	*Biblioteca Compartilhada para Pinpad - Especificação Detalhada - Version 1.08a (Apr 15th, 2013).*
🕮**EMV#1**	EMV - Integrated Circuit Card Specifications for Payment Systems - Book 1 - Application Independent ICC to Terminal Interface Requirements - Version 4.3 - November 2011.
🕮**EMV#2**	EMV - Integrated Circuit Card Specifications for Payment Systems - Book 2 - Security and Key Management - Version 4.3 - November 2011.
🕮**EMV#3**	EMV - Integrated Circuit Card Specifications for Payment Systems - Book 3 - Application Specification - Version 4.3 - November 2011.
🕮**EMV#4**	EMV - Integrated Circuit Card Specifications for Payment Systems - Book 4 - Cardholder, Attendant, and Acquirer Interface Requirements - Version 4.3 - November 2011.
🕮**EMV#CtlsA**	EMV - Contactless Specifications for Payment Systems - Book A - Architecture and General Requirements - Version 2.6 - February 2016.
🕮**PPMChip**	Master Card PayPass – M/Chip Reader Card Application Interface Specification V3.0.2 – May 2013; and EMV - Contactless Specifications for Payment Systems - Book C-2 - Kernel 2 Specification - Version 2.6 - February 2016.
🕮**VCPS**	VCPS - Visa Contactless Payment Specification - Version 2.2 - January 2016; and EMV - Contactless Specifications for Payment Systems - Book C-3 - Kernel 3 Specification - Version 2.6 - February 2016.
🕮**ExpPay**	Expresspay - American Express - Terminal Specification - Version 3.1 - April 2015; and EMV - Contactless Specifications for Payment Systems - Book C-4 - Kernel 4 Specification - Version 2.6 - February 2016.
🕮**D-PAS**	D-PAS - Discover Contactless - Terminal Application Specification - Version 1.1 - March 2015; and EMV - Contactless Specifications for Payment Systems - Book C-6 - Kernel 6 Specification - Version 2.6 - February 2016.
🕮**Pure**	Gemalto PURE - Contactless reader Specifications for PURE Dual-Interface cards and Mobile PURE - Version 2.1.8 - August 2016.

Definitions

Abecs	Or "*Associação Brasileira das Empresas de Cartões de Crédito e Serviços*" (Brazilian Association of Credit Card and Service Companies).
Acquirer	Company that captures and processes payment card transactions (also referred as "Acquirer Network").
AES	Or "Advanced Encryption Standard", also known as "Rijndael", it is a symmetric encryption algorithm defined by FIPS 197 or ISO / IEC 18033-3. Although his algorithm considers keys of different sizes, this specification specifically uses AES-128 (16-byte key). Being a symmetric algorithm, AES has a reverse function, in this specification denoted as **AES**$^{-1}$.

AID	Or “Application Identifier”, it is a 5 to 16-byte data object that identifies a payment application on an EMV card (Ex: Visa Credit = A0000000031010h).
Bypass	Situation in which the cardholder refuses to enter the PIN, pressing the [OK/ENTER] key on the pinpad with an empty input field.
Card Association	Institution that defines rules and provides interoperability for issuing and accepting payment cards (ex: VISA, MasterCard, etc.).
Cardholder	It refers to the person who uses a card to perform a payment transaction.
CBC	Or “Cipher-block Chaining”, data block encryption method.
Cleartext	Information or data are referred in this specification as “cleartext” when not encrypted (before encryption or after decryption).
Command	Instruction sent from the SPE to the pinpad for it to execute and return a response.
CRC	Or “Cyclic Redundancy Check”, validation code for error detection (see **section 7.2**).
Cryptogram	Block of data encrypted using a symmetric key (**DES**, **TDES**, **AES**) or an asymmetric public key (**RSA**).
CTLS	Not an initialism, this definition was created in this specification to refer to a contactless chip card, to differentiate it from the ICC.
DES	Or “Data Encryption Standard”, symmetric key encryption algorithm defined by the FIPS-46-3 standard. Being a symmetric algorithm, DES has a reverse function, in this specification denoted as $\mathbf{DES^{-1}}$.
Display	Device for displaying text and images on the pinpad, usually a liquid crystal display (LCD).
DUKPT	Or “Derived Unique Key Per Transaction”, encryption method defined by the ANSI X9.24:2009 standard (DUKPT:TDES)
ECB	Or “Electronic Codebook”, data block encryption method.
EMV	Standard for processing **ICC** payment cards, defined in 🕮**EMV#1**, 🕮**EMV#2**, 🕮**EMV#3** and 🕮**EMV#4**.
EMV Kernel	A “EMV Type Approval Level 2” certified software core that is responsible for processing EMV cards (ICC or CTLS) on the pinpad.
Fallback	Contingency process through which an **ICC** is accepted by the **SPE** through its magnetic stripe, usually due to a technical problem with the chip.
ICC	Or “Integrated Circuit Card”, for this specification it refers exclusively to contact chip card, according to ISO-7816.
Issuer	Entity, usually a bank, that issues cards for use in pinpads, whether magnetic, ICC or CTLS.
$\mathbf{K_{MOD}/K_{PUB}/K_{PRV}}$	RSA key managed by the SPE, used in the “Secure Communication” (**section 5.2**) and “Encrypted PAN” (**section 5.3**) modes, composed of a “module” ($\mathbf{K_{MOD}}$), a “public exponent” ($\mathbf{K_{PUB}}$) and a “private exponent”($\mathbf{K_{PRV}}$)
$\mathbf{K_{SEC}}$	AES key created by the pinpad in “Secure Communication” mode (**section 5.2**).
$\mathbf{K_{RAND}}$	Random TDES key used to encode card tracks in “End-to-End Encryption” (**section 5.4**).
KSN	Or “Key Serial Number”, it is the serial number of a key used in **DUKPT** encryption.
MK	Or “Master Key”, TDES encryption key inserted in the pinpad (in this specification referred to as **MK: TDES**).

MK/WK	PIN (or any data) encryption method defined by the ANSI X9.8 standard, which uses a MK and a “Working Key” provided externally.
Nibble	Equivalent to half byte, that is, a set of 4 bits (represents values 0h to Fh).
PAN	Or “Primary Account Number”, that is, the number of a payment card.
PCI	Or “Payment Card Industry Security Standards Council”, the normative council that defines security rules for card payment systems.
PIN	Or “Personal Identification Number”, the cardholder password.
Pinpad	Formally "PIN-pad", it is a secure device (“tamper proof”) that preserves encryption keys (MK/WK or DUKPT) and includes keyboard, display, magnetic card, ICC, SAM, CTLS interfaces and serial communication (RS232, USB, Bluetooth, etc.).
Protocol	Also referred to as “Communication Protocol”, it is a bidirectional data transfer mechanism between the SPE and the pinpad, so that the SPE can send the commands.
RSA	Or “Rivest, Shamir & Adleman”, an asymmetric encryption algorithm defined by the PKCS # 1 standard (RFC 3447). An RSA encryption key is made up of “module”, “public exponent” and “private exponent”.
RFU	Reserved for Future Use.
SAM	Or “Secure Application Module”, refers to a card with a chip (“2FF” format) embedded in the pinpad.
SPE	Portuguese initialism for “Electronic Payment System”, that is, the system that uses the pinpad, which can be, for example, a payment checkout or a self-service machine.
Tag	See “**TLV**”.
TDES	Or “Triple-DES”, a symmetric key encryption algorithm defined by the NIST SP 800-57 and SP 800-78-3 standard (2TDEA - keying option 2). Being a symmetric algorithm, TDES has a reverse function, in this specification denoted as $\mathbf{TDES}^{-1}$.
TLV	Or “Tag, Length and Value”, it is a data encoding method used by the EMV standard (see **section 7.1**).
Track	One of the three possible data blocks recorded on a magnetic card, referred as Track 1, Track 2 and Track 3. These data blocks usually contain the PAN, expiration date and other relevant information. ICC and CLTS may contain the same data blocks in their memory.
$\mathbf{WK_{PAN}}$	TDES encryption key used to encode sensitive information in communication messages (mainly PAN) in the method referred to in this specification as “Encrypted PAN” (see **section 5.3**).
XOR	Or “Exclusive OR”, it is a binary logical operation also represented by the symbol “⊕”.

Note: Terms extracted from the EMV standard are *highlighted* in this document to avoid loss of reference and, thus, facilitate its understanding.

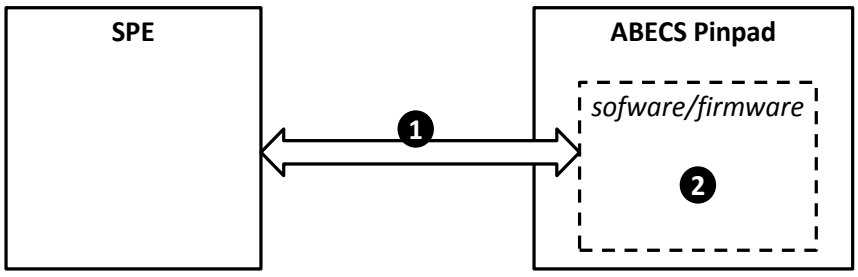
1. Introduction

This document is intended to specify in detail the “Abecs Pinpad”, with the objective of defining an interoperability standard for the use of pinpad type devices in the Brazilian market, mainly encompassing the following functionalities:

- Secure PIN capture;
- Magnetic card reading;
- Processing of EMV chip card (contact or contactless);
- Basic “human-machine interface” operations with the cardholder; and
- Identification and logistics management of the device.

“Abecs Pinpad” refers to a pinpad type device whose software/firmware respects this specification, which does not go into the merit of its hardware.

This specification focuses on two main technical points to guarantee the interoperability of an Abecs Pinpad in different SPEs:



❶ Communication protocol between the SPE and the Abecs Pinpad; and

❷ Internal operation of the pinpad, that is, specification of its software/firmware.

1.1. Target Audience

This specification is intended for the following audiences:

- Acquirer Networks;
- SPE developers; and
- Pinpad providers and their software/firmware developers.

1.2. Versioning

This specification adopts a numerical "**A.BC**" version convention, being:

"C" = Increased when the specification changes only for structural or explanatory improvements, not incurring in functional changes.

"B" = Increased when the specification relates to functional changes in the pinpad, but maintaining full compatibility with the SPE

"A" = Increased when the specification undergoes functional changes that influence both sides: SPE and pinpad.

1.3. Formats used in this document

This document mentions several data in commands and tables, and these data, due to their characteristics, must respect different coding rules.

The representation of a format follows the rule: "**[Format Character][..][Length]**"

[Format Character] = Uppercase letter that defines the format.

[..] = Optional, it indicates that the data is of variable size, from zero to **[Length]** bytes.

[Length] = One to three numeric digits representing the number of bytes used by the information.

Examples:

- The code "W256" indicates a 256-byte information encoded according to the "W" format.
- The code "K..99" indicates information of variable length (from 0 to 99 bytes) encoded according to the "K" format.

The following table details the formats adopted in this document:

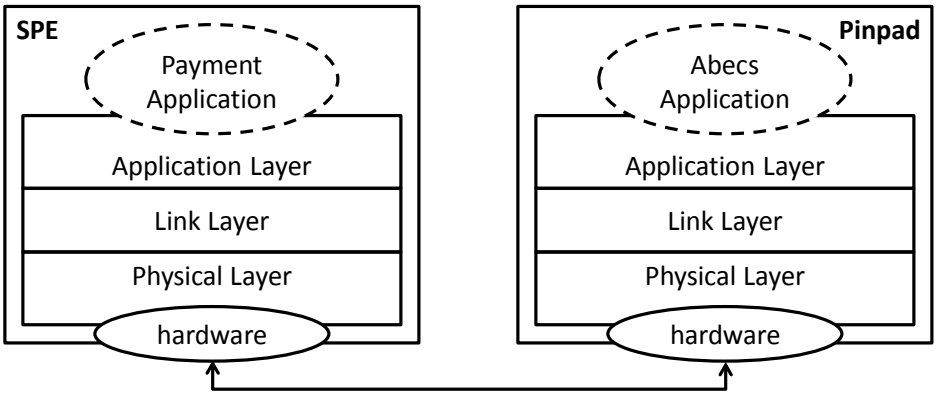
Format	Description
A	Alphanumeric information coded according to the ASCII table, containing bytes from 20h (space) to 7Eh (~). When the information is smaller than the defined field, it should be left aligned with spaces (20h) on the right. Example: If a field of format "A6" contains the information "TEXT", it is encoded as: 54h 45h 58h 54h 20h 20h.

Format	Description
S	Alphanumeric information coded according to the character table defined in **section 7.3.2**, which may contain bytes from 20h (space) to FFh. When the information is smaller than the defined field, it should be left aligned with spaces (20h) on the right. Example: If a field of format “S8” contains the information “Ação”, it is encoded as: 41h E7h E3h 6Fh 20h 20h 20h 20h.
N	Decimal numeric information encoded according to the ASCII table, and can only contain bytes from 30h ("0") to 39h ("9"). When the information is smaller than the defined field, it must be right aligned with zeros (30h) to the left. Example: If a field of format “N8” contains the value 1234, it is encoded as: 30h 30h 30h 30h 31h 32h 33h 34h.
H	Hexadecimal numeric information encoded according to the ASCII table, and may contain only bytes from 30h (“0”) to 39h (“9”), 41h (“A”) to 46h (“F”) and 61h (“a”) to 66h (“f”). When the information is smaller than the defined field, it must be right aligned with zeros (30h) to the left. Each two characters in hexadecimal format represent a byte (value from 00h to FFh), so the **[Length]** must always be an even number. Example: If a field of format “H4” contains the value 3F6Ch, it is encoded as: 33h 46h 36h 43h.
X	Numeric information in binary representation, preceded by the most significant byte. When the information is smaller than the defined field, it must be right aligned with leading zeros. Example: If an “X3” format field contains the value 3000 (BB8h), it is encoded as: 00h 0Bh B8h.
B	Generic information that allows any byte from 00h to FFh.

⚠ **IMPORTANT**: Data of type **“H..???”** are always preceded by a numeric field containing their size information. However, for historical reasons, this value is always divided by two (÷ 2), to represent the number of “original” bytes that generated the hexadecimal encoding.

2. Communication Protocol

This chapter describes the communication protocol between the SPE and the pinpad, considering three levels:



2.1. Physical Layer

The “physical layer” is the lower layer of the protocol that guarantees the transmission and reception of data bytes between the SPE and the pinpad.

Abecs Pinpad essentially considers a “physical layer” of serial communication, regardless of the technology (RS-232, USB, Bluetooth, etc.), with the following configurations when relevant to the medium used:

- Speed: 19.200 bps (bits/second);
- 8 bits/byte;
- No parity; and
- 1 stop bit.

2.2. Link Layer

The Link Layer is intended to define the data communication flow between the SPE and the pinpad, as well as to guarantee the integrity of the information exchanged (hereinafter referred to as "packets").

For the implementation of the Link Layer, the following special bytes (control characters) are used:

Name	Value	Description
«EOT»	04h	Pinpad response when receiving a **«CAN»**.
«ACK»	06h	Sent from the pinpad to the SPE when receiving a valid packet.
«DC3»	13h	Substitution byte, to prevent special bytes from traveling in the body of the packet.
«NAK»	15h	It is returned to the side that sent an invalid packet, requesting its retransmission.
«SYN»	16h	Indicates the start of a packet.
«ETB»	17h	Indicates the end of a packet.
«CAN»	18h	Sent from the SPE to the pinpad to cancel the execution of a command.

2.2.1. Packet format

The data packets exchanged between the parties, regardless of the direction (SPE ↔ pinpad), always have the following format:

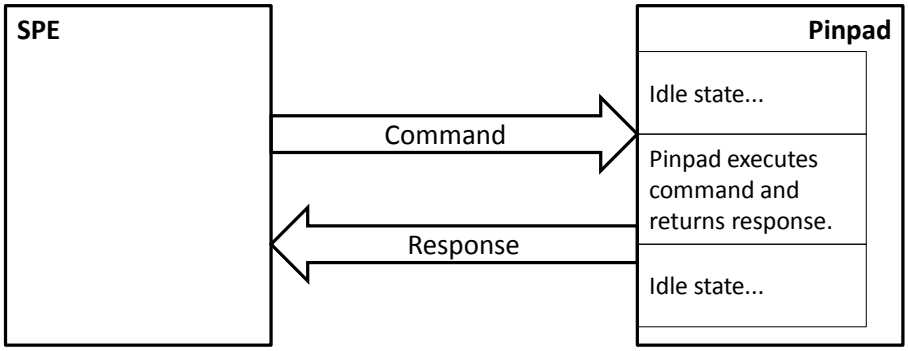
Name	Format	Description
PKTSTART	B1	Byte 16h («SYN») to identify the beginning of the packet.
PKTDATA	???	Packet contents, considering the following substitution rule: ▪ Byte 13h («DC3») is replaced by bytes 13h («DC3») and 33h; ▪ Byte 16h («SYN») is replaced by bytes 13h («DC3») and 36h; and ▪ Byte 17h («ETB») is replaced by bytes 13H («DC3») and 37h. The "original" packet (excluding any substitutions) can have a **maximum of 2049 bytes**.
PKTSTOP	B1	Byte 17h («ETB») to identify the end of the packet.
PKTCRC	X2	CRC-16 of **PKTDATA** and **PKTSTOP**, calculated over the "original" data, before any substitution made using the «DC3» byte (see algorithm in **section 7.2**).

⚠ For compatibility with the legacy base, the SPE can only send a packet to the pinpad with **PKTDATA** greater than 1024 bytes in the case of an "Abecs Command" (see **section 3.1.3**).

2.2.2. Communication flow

The communication flow always starts in the SPE. The pinpad is a “passive” entity, that is, it never sends data to the SPE unless requested.

- A data packet sent by the SPE to the pinpad is called “**command**”; and
- The data packet returned by the pinpad to the SPE is called a “**response**”.

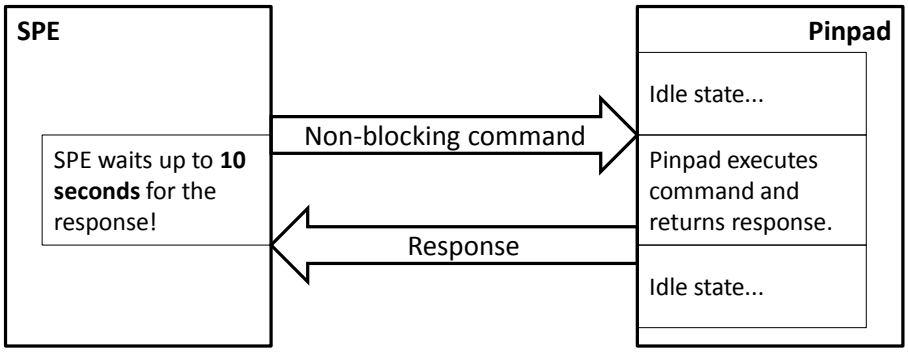


This specification considers two types of commands, “**blocking**” and “**non-blocking**”, as detailed below. To find out the type of a particular command, see its definition in **Chapter 3**.

➲ Non-blocking commands

Commands that do not require interaction with the cardholder are called “non-blocking”.

In this case, the SPE must wait up to 10 seconds for the response, informing a “time-out” error if it is not received within this time.

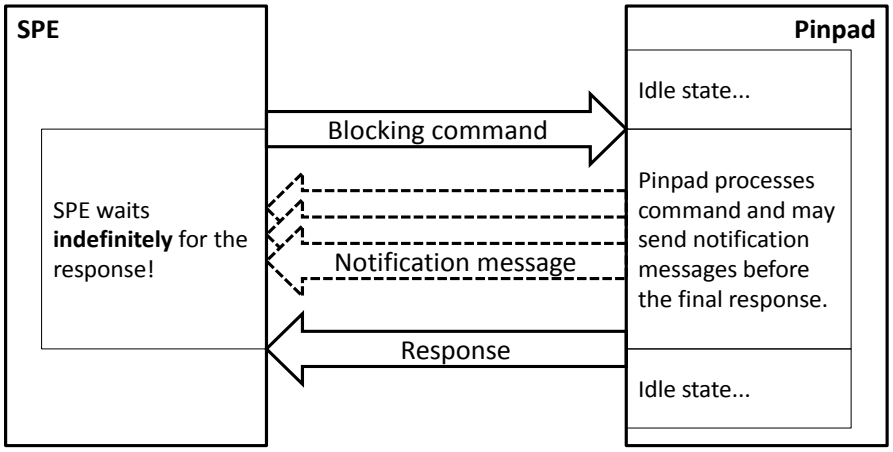


➲ Blocking commands

Commands that require interaction with the cardholder (for example, PIN capture) cause the pinpad to hold the processing indefinitely, being called “blocking”.

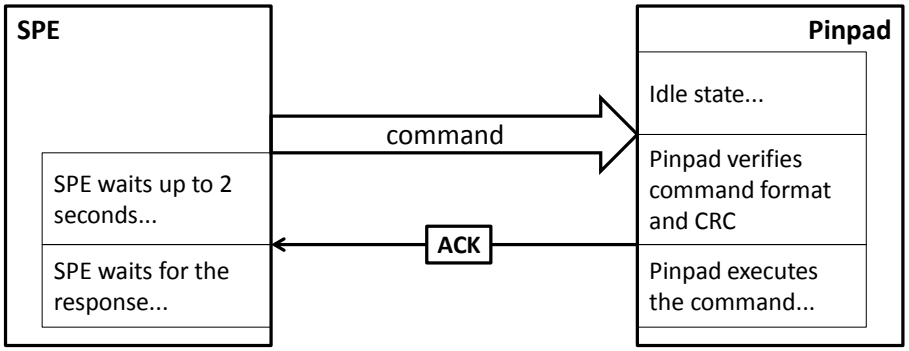
In this case, the SPE must wait indefinitely for a response, never informing a “time-out” error.

This type of command also allows the pinpad to return intermediate responses called “notification messages” to the SPE (see **section 2.3.3**).

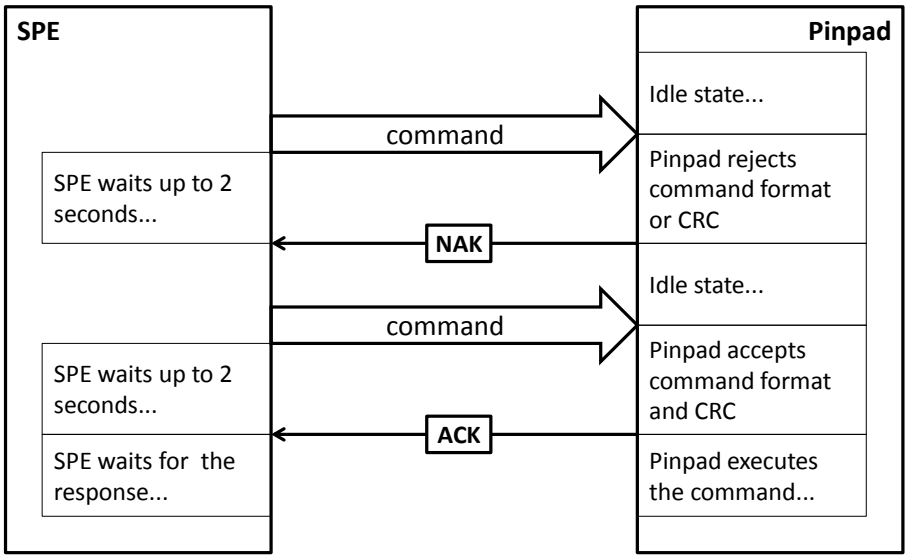


2.2.2.1. Command sending by SPE

The SPE sends a command packet to the pinpad according to the format described in **section 2.2.1**.



Upon receiving the command, the pinpad verifies the CRC and sends an **«ACK»** (06h) if the data is correct. If the values do not match, or the format of the packet is invalid, the pinpad sends a **«NAK»** (15h) and discards the packet.



The SPE must wait for an **«ACK»** or a **«NAK»** for 2 seconds after sending the command. Failure to receive any of these bytes aborts communication.

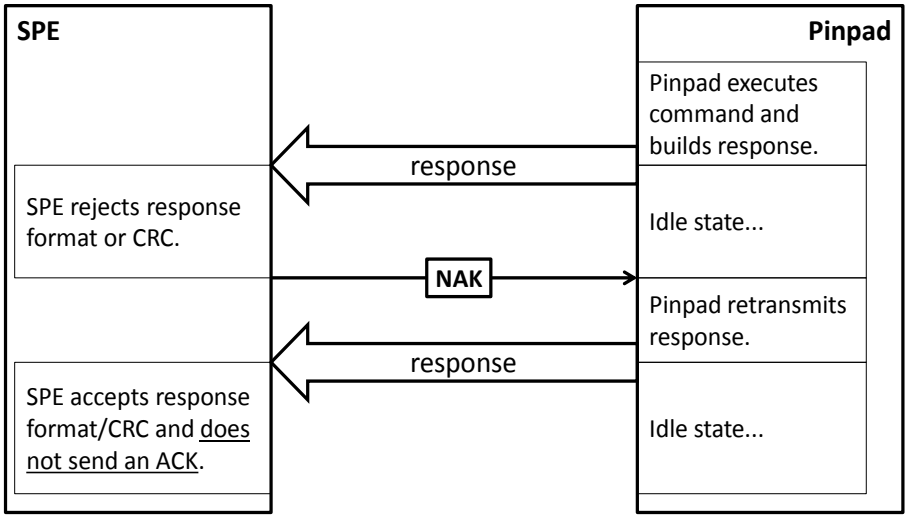
Upon receiving a **«NAK»**, the SPE must retransmit the command. The SPE must attempt to send the command up to 3 times, aborting after the 3rd **«NAK»** received.

➲ Examples:

SPE sends command to the pinpad, but it does not receive the CRC correctly.		
SPE ⇒	16 4F 50 4E 17 **00 00**	•OPN•••
The pinpad does not recognize the command as valid and returns **«NAK»**.		
⇐ PP	15	•
SPE resends the command, which is now received with the correct CRC.		
SPE ⇒	16 4F 50 4E 17 A8 A9	•OPN•¨©
The pinpad returns **«ACK»** and accepts the command.		
⇐ PP	06	•

2.2.2.2. Response return from the pinpad

When processing a command, the pinpad returns one or more response packets to the SPE (in the case of notification messages), according to the format described in **section 2.2.1**.



Upon receiving a response from the pinpad, the SPE must check the CRC of the received packet and send a **«NAK»** in case of error, returning to wait for the response. This process must be repeated up to 3 times.

If the received packet is intact, nothing should be sent.

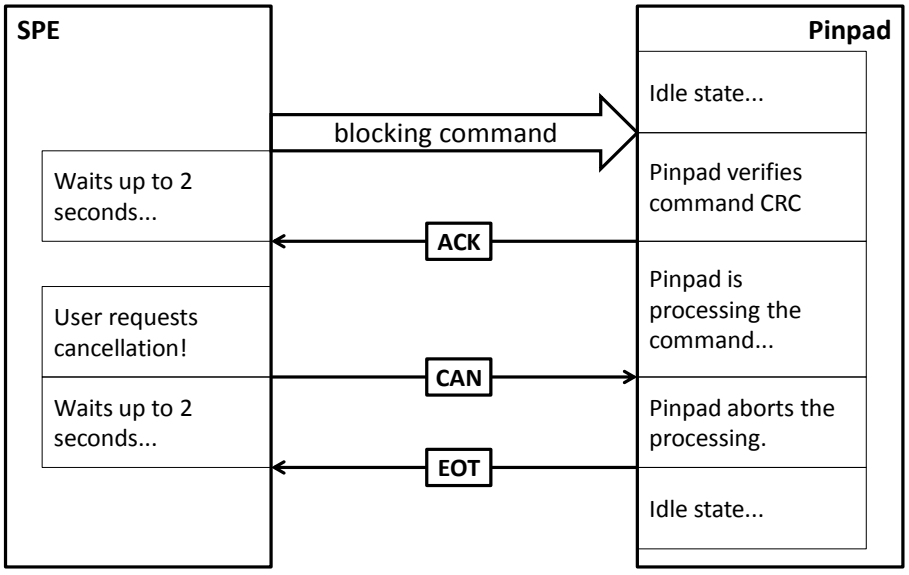
➲ Examples:

The SPE sends a command to the pinpad.		
SPE ⇒	16 44 53 50 30 33 32 20 20 20 20 4F 50 45 52 41 C7 C3 4F 20 20 20 20 20 20 20 46 49 4E 41 4C 49 5A 41 44 41 20 20 20 17 52 13	•DSP032••••OPERA ÇÃO•••••••FINALI ZADA••••R•
The pinpad returns **«ACK»** and accepts the command.		
⇐ PP	06	•
The pinpad returns the response, but the SPE does not receive a valid CRC.		
⇐ PP	16 44 53 50 30 30 30 17 **FF FF**	•DSP000•ÿÿ
The SPE does not recognize the response and sends a **«NAK»**, requesting its retransmission.		
SPE ⇒	15	•
The pinpad returns the response again, and it is now received with a valid CRC.		
⇐ PP	16 44 53 50 30 30 30 17 39 63	•DSP000•9c
The SPE accepts the response.		

2.2.2.3. Canceling a “blocking” command

In the case of “blocking” commands, the SPE must wait for an answer indefinitely. However, this type of command can be aborted at any time by the SPE by sending a **«CAN»** byte.

Upon receiving the **«CAN»** byte, the pinpad aborts the operation in progress, returns an **«EOT»** byte and returns to the idle state, in order to wait for a new command. In fact, the pinpad always responds **«EOT»** to a **«CAN»**, regardless of its status.



The SPE must wait for the **«EOT»** for 2 seconds, in order to obtain confirmation of the cancellation. If this byte is not received, the SPE must try to send the **«CAN»** up to 3 times.

During this time, the SPE must ignore any other bytes it may receive, as, coincidentally, there may be a response from the pinpad or a notification message being returned at the time of cancellation.

⚠ It is important that the SPE always initiates the communication flow with the pinpad by sending a **«CAN»**, in order to abort any blocking command that may be in process.

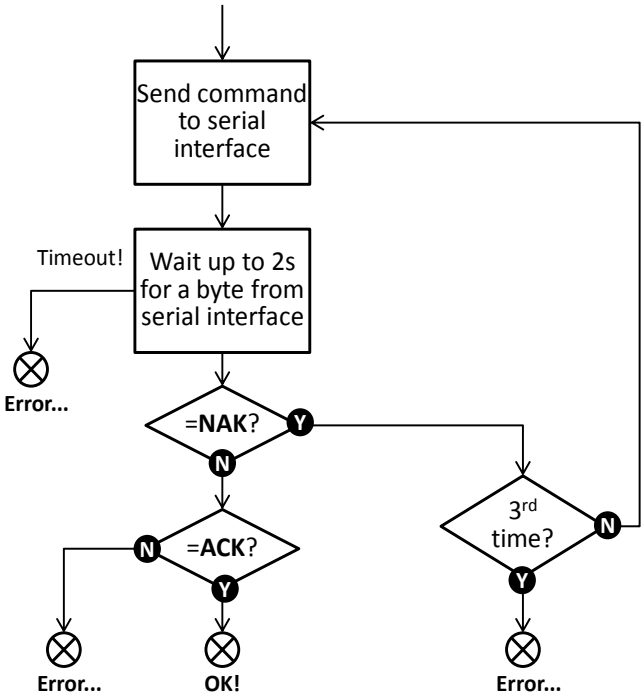
➲ Examples:

The SPE sends a blocking command to the pinpad.		
SPE ⇒	16 47 43 44 30 31 36 00 0C 00 01 3C 00 0E 00 01 0A 00 0B 00 02 00 09 17 C1 42	•GCD016••••<•••• ••••••••ÁB
The pinpad returns **«ACK»** and accepts the command.		
⇐ PP	06	•
After a delay, the SPE decides to abort the command by sending a **«CAN»**.		
SPE ⇒	18	•
The pinpad aborts the execution immediately and returns **«EOT»**.		
⇐ PP	04	•

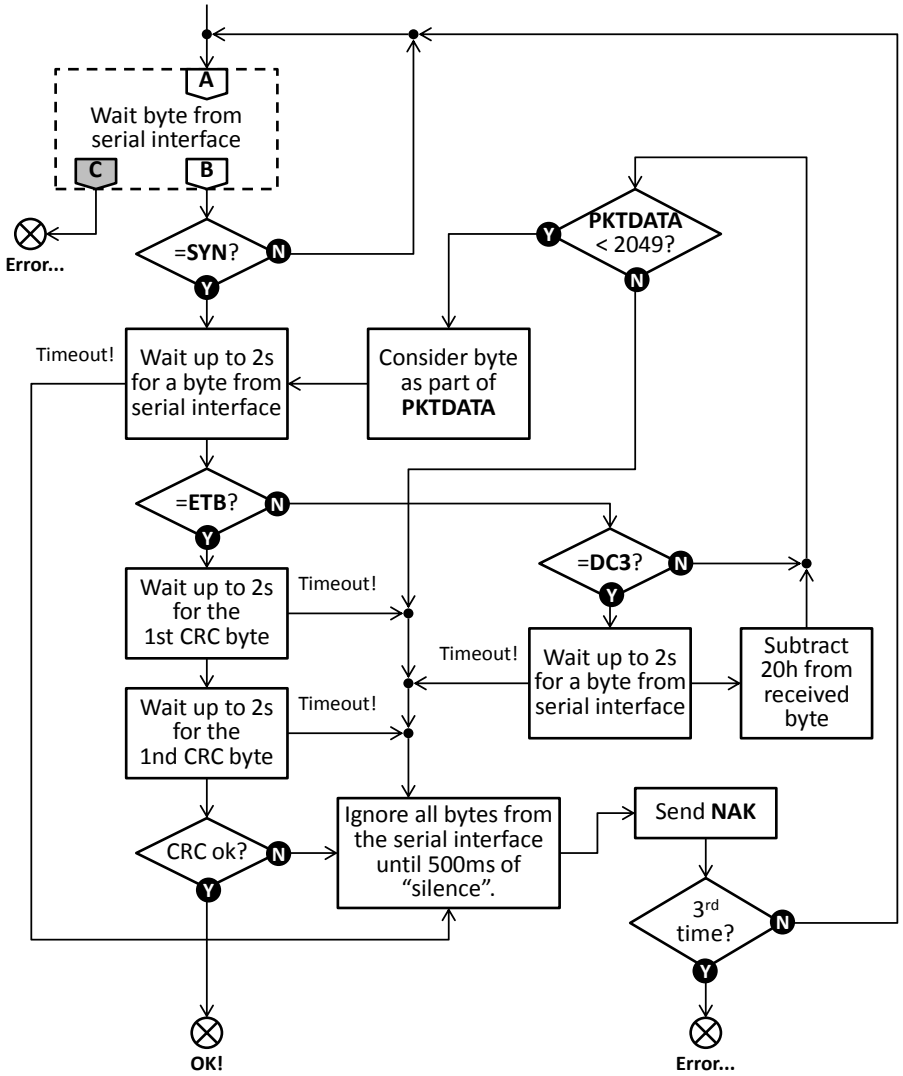
2.2.3. Processing flows in the SPE

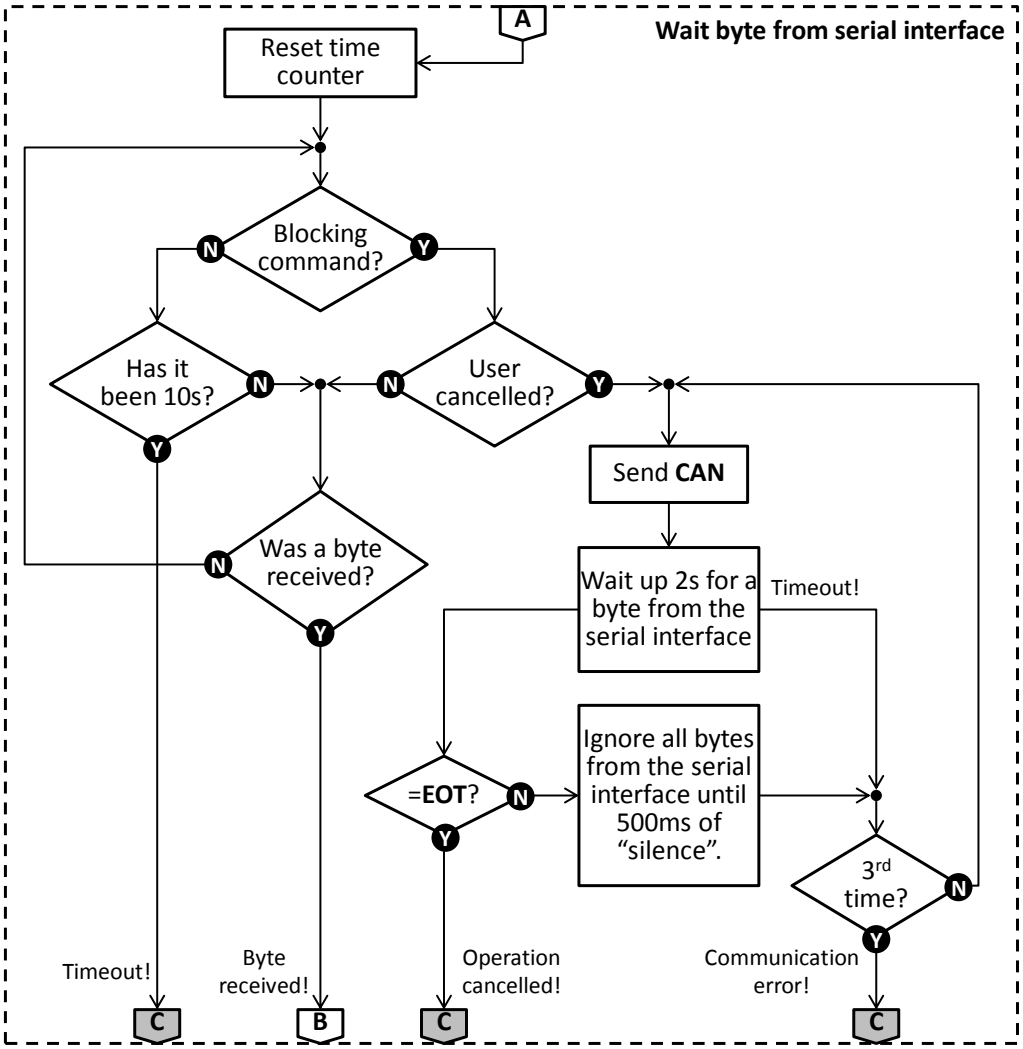
This section describes the internal processing flows in the SPE for the correct treatment of the Link Layer.

➲ Command sending



➲ Response receiving




Wait byte from serial interface
A
Reset time counter
Blocking command?
N
Y
Has it been 10s?
N
Y
User cancelled?
N
Y
Send CAN
Was a byte received?
N
Y
Wait up 2s for a byte from the serial interface
Timeout!
=EOT?
N
Y
Ignore all bytes from the serial interface until 500ms of “silence”.
3rd time?
N
Y
Timeout!
Byte received!
Operation cancelled!
Communication error!
C
B
C
C

2.3. Application Layer

The “Application Layer” defines the format of the data that travels in **PKTDATA** through the Link Layer, being that it depends on the direction of the packet (SPE ↔ pinpad).

⚠ If **PKTDATA** starts with the **«DC2»** byte (12h), it is encrypted according to the “Secure Communication” method described in **section 5.2**. In this case, the formats described in this section refer to the **CLRDATA** field.

2.3.1. Command format

All commands sent by the SPE to the pinpad must respect the format described below.

A command may or may not contain data blocks (parameters) of up to 999 bytes each, always preceded by the size information.

Field Id.	Format	Description
CMD_ID	A3	Command code (described in **Chapter 3**).
CMD_LEN1	N3	Length of the following data (from “000” to “999” bytes)
CMD_BLK1	B..999	First parameter block
CMD_LEN2	N3	Length of the following data (from “000” to “999” bytes)
CMD_BLK2	B..999	Second parameter block
...	...	...
CMD_LENn	N3	Length of the following data (from “000” to “999” bytes)
CMD_BLKn	B..999	Last parameter block

2.3.2. Response format

The responses returned by the pinpad to the SPE must respect the formats described below.

➲ Successful execution

A response to the successful execution of a command may (or may not) contain data blocks of up to 999 bytes each, always preceded by the length information.

Field Id.	Format	Description
RSP_ID	A3	Response code (same as **CMD_ID**)
RSP_STAT	N3	**“000**” value, meaning success.
RSP_LEN1	N3	Length of the following data (from “000” to “999” bytes)

Field Id.	Format	Description
RSP_BLK1	B..999	First response data block
RSP_LEN2	N3	Length of the following data (from “000” to “999” bytes)
RSP_BLK2	B..999	Second response data block
...	...	...
RSP_LENn	N3	Length of the following data (from “000” to “999” bytes)
RSP_BLKn	B..999	Last response data block

➲ Execution error

In the event of an error in the execution of a recognized command, the pinpad always returns the following 6-byte response.

Field Id.	Format	Description
RSP_ID	A3	Response code (same as **CMD_ID**)
RSP_STAT	N3	Processing status (≠ “000”), as defined in **section 3.1**.

⚠ If **RSP_STAT** ≠ “000”, the response shall never contain data!

2.3.3. Notification messages

During the execution of “blocking” commands, the pinpad may send notification messages to the SPE, so that it can show them to the user.

Notification messages have the following format:

Field Id.	Format	Description
RSP_ID	A3	**“NTM”**
RSP_STAT	N3	“000” (always success)
RSP_LEN1	N3	“000” to “032”
NTM_MSG	S32	Message to be presented to the SPE user, formatted so that it can be displayed in 2 lines of 16 characters.

2.3.4. Exceptions

If a command is not recognized as valid by the pinpad, it cannot return a coherent answer (**RSP_ID** with the same value as **CMD_ID**). In this case, the following response is used:

Field Id.	Format	Description
RSP_ID	A3	**"ERR"**
RSP_STAT	N3	↳ST_NOSEC = "Secure Communication" not stablished (see **section 5.2**); ↳ST_ERRPKTSEC = **PKTDATA** codification error in case of encrypted packet (see **section 5.2**); or ↳ST_INVCALL = **CMD_ID** is not recognized by the pinpad.

3. Commands

This chapter details the commands processed by the pinpad at the Application Layer, according to the format presented in **section 2.3**.

⚠ All formats and examples described in this chapter disregard the Link Layer, as well as the "Secure Communication" mode, given that the commands and responses operate above these layers.

3.1. Preliminary Information

3.1.1. Return Codes

As shown in **section 2.3**, the pinpad response packets must contain a "processing result" information (**RSP_STAT**) indicating success or, in the event of failure, the reason. The values accepted by this specification are described in the following table:

Name	Value	Description
↳ST_OK	000	Command executed successfully.
↳ST_NOSEC	003	Attempted to use "Secure Communication" when it has not been established.
↳ST_F1	004	Function #1 key pressed.
↳ST_F2	005	Function #2 key pressed.
↳ST_F3	006	Function #3 key pressed.
↳ST_F4	007	Function #4 key pressed.
↳ST_BACKSP	008	Clear (backspace) key pressed.

Name	Value	Description
↳ST_ERRPKTSEC	009	Error decoding data received via “Secure Communication”; or Cleartext command received with “Secure Communication” established.
↳ST_INVCALL	010	Invalid call to a command (previous operations are necessary) or unknown command (in case of an “**ERR**” response).
↳ST_INVPARM	011	An invalid parameter was passed to the command.
↳ST_TIMEOUT	012	The maximum time stipulated for the operation has been exhausted.
↳ST_CANCEL	013	Operation canceled by the cardholder.
↳ST_MANDAT	019	A mandatory parameter was not sent by the SPE.
↳ST_TABVERDIF	020	EMV Tables version differs from the expected.
↳ST_TABERR	021	Error when trying to write tables (lack of space, for example).
↳ST_INTERR	040	Internal pinpad error (unexpected situation that does not correspond to the other error codes described here).
↳ST_MCDATAERR	041	Magnetic card reading error.
↳ST_ERRKEY	042	MK / DUKPT referenced is not present in the pinpad.
↳ST_NOCARD	043	There is no ICC present in the coupler or CTLS detected by the antenna.
↳ST_PINBUSY	044	Pinpad cannot process PIN capture temporarily due to security constrains (such as when the capture limit is reached within a time interval).
↳ST_RSPOVRFL	045	Response data exceeds the maximum allowed size.
↳ST_ERRCRYPT	046	Generic cryptographic validation error.
↳ST_DUMBCARD	060	ICC inserted, but not responding (“mute”).
↳ST_ERRCARD	061	Communication error between the pinpad and the ICC or CTLS.
↳ST_CARDINVALIDAT	067	ICC is invalidated.
↳ST_CARDPROBLEMS	068	ICC with problems. This status is valid for many situations in which the ICC does not behave as expected and the transaction must be terminated.
↳ST_CARDINVDATA	069	The ICC behaves correctly but has invalid or inconsistent data.
↳ST_CARDAPPNAV	070	ICC with no matching application.
↳ST_CARDAPPNAUT	071	The application selected in the ICC cannot be used in this situation.
↳ST_ERRFALLBACK	076	High level error in the ICC that allows fallback to magnetic stripe.
↳ST_INVAMOUNT	077	Invalid amount for the transaction.

Name	Value	Description
↳ST_ERRMAXAID	078	Number of candidate AIDs exceeds the processing capacity of the EMV kernel.
↳ST_CARDBLOCKED	079	Card is blocked.
↳ST_CTLSMULTIPLE	080	More than one CTLS was presented to the reader simultaneously.
↳ST_CTLSCOMMERR	081	Communication error between the pinpad (antenna) and the CTLS.
↳ST_CTLSINVALIDAT	082	CTLS is invalidated.
↳ST_CTLSPROBLEMS	083	CTLS with problems. This status is valid for many situations in which the CTLS does not behave as expected and the transaction must be terminated.
↳ST_CTLSAPPNAV	084	CTLS with no matching application.
↳ST_CTLSAPPNAUT	085	The application selected in the CTLS cannot be used in this situation.
↳ST_CTLSEXTCVM	086	Cardholder must perform a validation on his device (mobile phone, for example) and then re-present it to the pinpad.
↳ST_CTLSIFCHG	087	CTLS processing resulted in “change interface” (request ICC or magnetic card).
↳ST_MFNFOUND	100	Media file not found.
↳ST_MFERRFMT	101	Media file format error.
↳ST_MFERR	102	Media file loading error.

⚠ In the detail sections of the commands in this specification, we seek to list only the relevant return codes for the command being described. Most commands support the ↳ST_OK, ↳ST_INVPARM, ↳ST_MANDAT and ↳ST_INTERR return codes and these are omitted to simplify the document.

3.1.2. Obsolete Commands

Some commands described here are considered **obsolete**, that is, they will be removed in future versions of this specification.

⚠ The SPE **shall not use an obsolete command** for a pinpad that is known to follow this specification. To recognize an Abecs Pinpad, it shall use the **“OPN”** command described in **section 3.2.2**.

⚠ The pinpad **shall implement an obsolete command** while it is described in this specification, in order to maintain compatibility with legacy systems.

Commands defined as obsolete are individually identified throughout this chapter.

3.1.3. Abecs Commands

All new commands of this specification (not included in 🕮**BibComp**) are called "**Abecs Commands**" and follow a flexible format, in which the parameters and response data are coded in a standardized way, always preceded by identification and length, similar to TLV coding described in **section 7.1**, but in a proprietary and simplified way. This allows total flexibility in any future evolution of the commands.

For the "Abecs Commands", data packets traveling between the SPE and the pinpad can have up to 2044 bytes. For the other commands of this specification, the limit is 1024 bytes.

3.1.3.1. Command format

Commands sent from the SPE to the pinpad follow the format below:

Field Id.		Format	Description
CMD_ID		A3	Command code.
CMD_LEN1		N3	Length of the following data.
CMD_BLK1	**CMD_PARID**	X2	Parameter identification (**SPE_xxxx**).
	CMD_PARLEN	X2	Parameter length, up to 995 (03E3h).
	CMD_PAR	???	Parameter data.
	…	…	…
	CMD_PARID	X2	Parameter identification (**SPE_xxxx**).
	CMD_PARLEN	X2	Parameter length, up to 995 (03E3h).
	CMD_PAR	???	Parameter data.
CMD_LEN2		N3	Length of the following data.
CMD_BLK2	**CMD_PARID**	X2	Parameter identification (**SPE_xxxx**).
	CMD_PARLEN	X2	Parameter length, up to 995 (03E3h).
	CMD_PAR	???	Parameter data.
	…	…	…

➲ Composition rules

- The SPE can send the parameters in any order, not necessarily the same as shown in the description of the commands in this chapter.
- The SPE can divide the parameters into one or more blocks (**CMD_BLKn**), given that the **CMD_LENn** field allows a maximum of only 999 bytes.
- The parameters sent to the pinpad can be mandatory or optional, as required by the command. The pinpad will simply ignore parameters there are unknown or unnecessary for the command being processed.

➲ Presentation

The following convention is adopted to simplify the specification of Abecs Commands in this chapter:

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= "**XXX**").
SPE_xxx	(*)	Input parameter.
...	...	...
SPE_xxx	(*)	Input parameter.

(*) Presence definition:

M = Parameter is mandatory for the command. If not sent by the SPE, the pinpad will return ↳ST_MANDAT.

MD = Parameter is mandatory depending on the situation (it may be a complement to another parameter, for example). If its presence is required but it is absent, the pinpad will return ↳ST_MANDAT.

O = Parameter is optional and the SPE will only send it if desired for processing the command. If the information is required for the processing, the pinpad will use a default value predefined in this specification.

➲ List of parameters

CMD_PARID	Value	Format	Description
SPE_IDLIST	0001h	B..128 (n×X2,n≤64)	List of return data identifiers (up to 64).
SPE_MTHDPIN	0002h	N1	Method to be used for PIN encryption: **"1"** = MK/WK:TDES:PIN; and **"3"** = DUKPT:TDES:PIN (see **section 5.1.1**).
SPE_MTHDDAT	0003h	N2	Method to be used for data encryption: **"10"** = MK/WK:TDES:DAT (**ECB** block encryption); **"11"** = MK/WK:TDES:DAT (**CBC** block encryption); **"50"** = DUKPT:TDES:DAT#3 (**ECB** block encryption**,** see **section 5.1.1**); and **"51"** = DUKPT:TDES:DAT#3 (**CBC** block encryption**,** see **section 5.1.1**).
SPE_TAGLIST	0004h	B..128	List of tags referring to the EMV objects required by the SPE.
SPE_EMVDATA	0005h	B..512	EMV objects sent to the pinpad (in TLV format - see **section 7.1**).

CMD_PARID	Value	Format	Description
SPE_CEXOPT	0006h	A6	"**CEX**" command options. "**0**xxxxx" = Ignore keys; "**1**xxxxx" = Verify key pressing. "x**0**xxxx" = Ignore magnetic card; "x**1**xxxx" = Verify magnetic card swiping. "xx**0**xxx" = Ignore ICC; "xx**1**xxx" = Verify ICC insertion; "xx**2**xxx" = Verify ICC removal. "xxx**0**xx" = Ignore CTLS (do not activate antenna); "xxx**1**xx" = Activate antenna and verify CTLS presence. "xxxx00" = RFU.
SPE_TRACKS	0007h	N4	Identification of track data to be returned by the pinpad in **"GTK"** command.
SPE_OPNDIG	0008h	N1	Number of **numeric** digits (**even** number) to be preserved as cleartext at the beginning of encrypted tracks (accepted values: "0", "2", "4", "6", "8").
SPE_KEYIDX	0009h	N2	DUPKT or MK slot index ("00" to "99")
SPE_WKENC	000Ah	B16	Working Key encrypted by MK:TDES.
SPE_MSGIDX	000Bh	X2	Index to the message to be presented.
SPE_TIMEOUT	000Ch	X1	Wait time for a cardholder action (in seconds - up to 255). **IMPORTANT**: This parameter reflects the cardholder inactivity time and not the maximum command execution time.
SPE_MINDIG	000Dh	X1	Minimum number of digits to be captured on the pinpad (from 0 to 32).
SPE_MAXDIG	000Eh	X1	Maximum number of digits to be captured on the pinpad (from 0 to 32).
SPE_DATAIN	000Fh	B..995	Generic data to be sent to the pinpad.
SPE_ACQREF	0010h	N2	Acquirer identifier for searching the AID Tables (de "01" a "99").
SPE_APPTYPE	0011h	N..20	Application type identifiers for searching the AID Tables (from "01" to "98"). This field supports 1 to 10 different identifiers.
SPE_AIDLIST	0012h	A..512	Specific list of records in the AID Tables to be used in the transaction processing, which can include up to 128 entries in the "AARR" format, as follows: "AA" = Identifier of the Acquirer responsible for the table (from "01" to "99"); and "RR" = Index of the record in the table (from "01" to "ZZ").

CMD_PARID	Value	Format	Description
SPE_AMOUNT	0013h	N12	Transaction amount *(Amount, authorized)*, in cents.
SPE_CASHBACK	0014h	N12	Cashback amount *(Amount, other)* , in cents.
SPE_TRNDATE	0015h	N6	Transaction date (“AAMMDD”)
SPE_TRNTIME	0016h	N6	Transaction time (“HHMMSS”)
SPE_GCXOPT	0017h	N5	**“GCX”** command options: “**0**xxxx” = Wait for magnetic card or ICC; or “**1**xxxx” = Wait for magnetic card; ICC or CTLS; “x**0**xxx” = Show transaction amount on the card waiting prompt, if not zero. “x**1**xxx” = Do not show transaction amount. “xx000” = RFU.
SPE_GOXOPT	0018h	N5	**“GOX”** command options: “**1**xxxx” = PAN is in the Exception List (only for ICC EMV). “x**1**xxx” = Transaction shall not be offline approved (only for ICC EMV). “xx**1**xx” = Do not allow PIN bypass. “xxx00” = RFU.
SPE_FCXOPT	0019h	N4	**“FCX”** command options: “**0**xxx” = Transaction approved by the Acquirer. “**1**xxx” = Transaction declined by the Acquirer. “**2**xxx” = Unable to go online (or invalid response from the Acquirer). “x000” = RFU.
SPE_TRMPAR	001Ah	B10	*Terminal Risk Management* parameters to be used on **“GOX”**: ▪ *Terminal Floor Limit* (“X4” format, in cents); ▪ *Target Percentage to be used for Biased Random Selection* (“X1” format); ▪ *Threshold Value for Biased Random Selection* (“X4” format, in cents); and ▪ *Maximum Target Percentage to be used for Biased Random Selection* (“X1” format).
SPE_DSPMSG	001Bh	S..128	Display message in free format, may have line break characters (0Dh). When formatting this message, the SPE shall respect the pinpad display capabilities (see **PP_DSPTXTSZ**).
SPE_ARC	001Ch	A2	*Authorization Response Code* (approval/declination code returned by the Acquirer).
SPE_IVCBC	001Dh	B8	“IV” (Initialization Vector) to be used in CBC block cryptography

CMD_PARID	Value	Format	Description
SPE_MFNAME	001Eh	A8	Media file name (only numeric characters and letters, without spaces or symbols). The file name is not case sensitive, that is, the names “ImgAlt01” and “IMGALT01” represent the same file.
SPE_MFINFO	001Fh	B10	Information about the media file: X4 = Size (de 0 a 4294967295 bytes). B2 = CRC of the file. B1 =Type (01h = **PNG**, 02h = **JPG**, 03h = **GIF**, other values = RFU); and B3 = RFU (000000h).
SPE_MNUOPT	0020h	S..24	Text with a menu option.
SPE_TRNTYPE	0021h	B1	Transaction type to be performed: 00h = Payment; 01h = Cash; 09h = Payment with cashback; 20h = Refund; 30h = Balance inquiry; or Other values according to ISO 8583:1987.
SPE_TRNCURR	0022h	N3	Currency code to be used in the transaction (ex.: Real = “986”, Dollar = “840”).
SPE_PANMASK	0023h	N4	PAN masking definition in “LLRR” format: “LL” = Number of open digits on the left; and “RR” = Number of open digits on the right.
SPE_PBKMOD	0024h	B256	RSA public key modulus (2048 bits).
SPE_PBKEXP	0025h	B..3	RSA public key exponent.

3.1.3.2. Response format

The responses returned to the SPE by the pinpad follow the format below:

Field Id.	Format	Description
RSP_ID	A3	Command code.
RSP_STAT	N3	Processing result, as defined in **section 3.1**.
RSP_LEN1	N3	Length of the following data (**RSP_BLK1**).

Field Id.		Format	Description
RSP_BLK1	**RSP_DATID**	X2	Response data field identifier (**PP_xxxx**).
	RSP_DATLEN	X2	Length of the response data field, up to 995 (03E3h).
	RSP_DAT	???	Response data field.
	…	…	…
	RSP_DATID	X2	Response data field identifier (**PP_xxxx**).
	RSP_DATLEN	X2	Length of the response data field, up to 995 (03E3h).
	RSP_DAT	???	Response data field.
RSP_LEN2		N3	Length of the following data (**RSP_BLK2**).
RSP_BLK2	**RSP_DATID**	X2	Response data field identifier (**PP_xxxx**).
	RSP_DATLEN	X2	Length of the response data field, up to 995 (03E3h).
	RSP_DAT	???	Response data field.
	…	…	…

➲ Composition rules

- The pinpad may return data fields in any order, not necessarily the same as shown in the description of the commands in this chapter.
- The pinpad may divide the response data into one or more blocks (**RSP_BLKn**), since the **RSP_LENn** size discriminator allows a maximum of only 999 bytes.
- Response data returned by the pinpad can be mandatory or optional, according to the command specification. The SPE shall ignore any unknown or unnecessary response data field.

➲ Presentation

The following convention is adopted to simplify the specification of Abecs Commands in this chapter:

Field Id.	Presence	Description / Remark
CMD_ID	M	Response code (= "**XXX**").
RSP_STAT	M	Only the relevant return codes are presented, in order to complement **section 3.1**.
PP_xxx	(*)	Response data field.
…	…	…
PP_xxx	(*)	Response data field.

(*) Presence definition:

M = Data field is mandatory for the command. If it is not returned by the pinpad, the SPE shall end the operation with a fatal error.

MD = Data field is mandatory depending on the situation (it can be a complement to another data, for example). If its presence is required but it is absent, the SPE shall end the operation with a fatal error.

MR = Data is mandatory if required by the SPE in the command. The SPE shall criticize or not its presence according to the situation.

O = Data is optional as a processing result and the SPE should not criticize its absence.

➲ List of return data fields

RSP_DATID	Value	Format	Description
PP_SERNUM (†)	8001h	A..32	Pinpad serial number (free format, it depends on the manufacturer).
PP_PARTNBR (†)	8002h	A..32	Pinpad part number (free format, it depends on the manufacturer).
PP_MODEL (†)	8003h	A..20	Model / hardware version, in the format: "**xx...xx;m...m**", where: ▪ "xx...xx" is the device name; and ▪ "m...m" is the memory capacity ("512KB", "1MB", "2MB", ...).
PP_MNNAME (†)	8004h	A..20	Name of the manufacturer (free format).
PP_CAPAB (†)	8005h	A10	Pinpad capabilities: "**0**xxxxxxxxx" = Does not support CTLS; "**1**xxxxxxxxx" = Supports CTLS. "x**0**xxxxxxxx" = Display is not graphic; "x**1**xxxxxxxx" = Monochromatic graphic display; "x**2**xxxxxxxx" = Color graphic display. "xx00000000" = RFU.
PP_SOVER (†)	8006h	A..20	Basic software or operating system version (free format).
PP_SPECVER (†)	8007h	A4	Specification version, in "V.VV" format (in this case, fixed "**2.12**")
PP_MANVERS (†)	8008h	A16	"Manager" application version, in the format "VVV.VV YYMMDD".
PP_APPVERS (†)	8009h	A16	"Abecs" application version, in the format "VVV.VV YYMMDD".
PP_GENVERS (†)	800Ah	A16	"Extension" application version, in the format "VVV.VV YYMMDD".
PP_KRNLVER (†)	8010h	A..20	ICC EMV kernel version.
PP_CTLSVER (†)	8011h	A..20	CTLS EMV kernel version (general or entry point).
PP_MCTLSVER (†)	8012h	A..20	CTLS EMV kernel version - MasterCard PayPass.
PP_VCTLSVER (†)	8013h	A..20	CTLS EMV kernel version - VISA PayWave.
PP_AECTLSVER (†)	8014h	A..20	CTLS EMV kernel version - American Express.

RSP_DATID	Value	Format	Description
PP_DPCTLSVER (†)	8015h	A..20	CTLS EMV kernel version - Discover.
PP_PUREVER (†)	8016h	A..20	CTLS EMV kernel version - Pure.
PP_DSPTXTSZ (†)	8020h	N4	Maximum number of rows and columns of the display for showing messages in text mode (“RRCC” format).
PP_DSPGRSZ (†)	8021h	N8	Maximum number of rows and columns of the graphic display for image presentation (“RRRRCCCC” format, in pixels).
PP_MFSUP (†)	8022h	A..20	Supported media file types: “**1**xxx ...” = Supports PNG format; “x**1**xx ...” = Supports JPG format. “xx**1**x ...” = Supports GIF format.
---	8030h	---	Reserved.
---	8031h	---	Reserved.
PP_MKTDESP (†)	8032h	A100	100 characters representing the **MK:TDES:PIN** key map contained in the pinpad, with each character corresponding to a position (from “00” to “99”), indicating: “0” = Key absent (not loaded); “1” = Key present (loaded); and “2” = Slot not supported.
PP_MKTDESD (†)	8033h	A100	Same for **MK:TDES:DAT** slots.
---	8034h	---	Reserved.
PP_DKPTTDESP (†)	8035h	A100	Same for **DUKPT:TDES:PIN** slots.
PP_DKPTTDESD (†)	8036h	A100	Same for **DUKPT:TDES:DAT** slots.
PP_EVENT	8040h	A2	Event detected by the pinpad in the “**CEX**” command: “00” = [OK/ENTER] key pressed; “02” = [↑] key pressed; “03” = [↓] key pressed; “04” = [F1] key pressed; “05” = [F2] key pressed; “06” = [F3] key pressed; “07” = [F4] key pressed; “08” = [CLEAR] key pressed; “13” = [CANCEL] key pressed; “90” = A magnetic card was swiped; “91” = ICC removed (or already absent); “92” = ICC inserted (or already present); “93” = CTLS not detected in 2 (two) minutes; and “94” = CTLS detected.
PP_TRK1INC	8041h	A..60	Card Track 1, incomplete (see **section 5.4.1**)
PP_TRK2INC	8042h	A..30	Card Track 2, incomplete (see **section 5.4.1**)

RSP_DATID	Value	Format	Description
PP_TRK3INC	8043h	A..30	Card Track 3, incomplete (see **section 5.4.1**)
PP_TRACK1	8044h	B..88	Card Track 1 (complete), in cleartext or encrypted (see **section 5.4.2.1**). **Note**: Although Track 1 is represented in ASCII, this field follows the "B" format in case the data is encrypted.
PP_TRACK2	8045h	B..28	Card Track 1 (complete), in cleartext or encrypted (see **section 5.4.2.2**). Each Track 2 symbol occupies a nibble, according to the following code: 0h (0000) → "0" Ah (1010) → ":" Dh (1101) → "=" ... Bh (1011) → ";" Eh (1110) → ">" 9h (1001) → "9" Ch (1100) → "<" Fh (1110) → "?" Data are left aligned, with trailing Fh ("?") if necessary.
PP_TRACK3	8046h	B..60	Card Track 1 (complete), in cleartext or encrypted (same format as **PP_TRACK2**).
PP_TRK1KSN	8047h	B10	KSN of DUKPT used for Track 1 encryption.
PP_TRK2KSN	8048h	B10	KSN of DUKPT used for Track 2 encryption.
PP_TRK3KSN	8049h	B10	KSN of DUKPT used for Track 3 encryption.
PP_ENCPAN	804Ah	B..16	Card PAN, in cleartext of encrypted (see **section 5.4.2.2**). Each digit of the PAN occupies a nibble. Data is left aligned with trailing Fh, if necessary. Example: A PAN "9781234789432" is encoded as: 97h 81h 23h 47h 89h 43h 2Fh.
PP_ENCPANKSN	804Bh	B10	KSN of DUKPT used for PAN encryption.
PP_KSN	804Ch	B10	KSN of DUKPT used for PIN or data encryption.
PP_VALUE	804Dh	A..32	Value captured by the pinpad.
PP_DATAOUT	804Eh	B..256	Generic data returned by the pinpad.
PP_CARDTYPE	804Fh	N2	"**GCX**" response: Card type. **"00"** = Magnetic; **"03"** = ICC EMV; **"05"** = CTLS magstripe mode; or **"06"** = CTLS EMV.
PP_ICCSTAT	8050h	N1	"**GCX**" response: Status of the previous ICC processing.
PP_AIDTABINFO	8051h	A..120	"**GCX**" response: Information from the AID Table, which may contain up to 20 concatenated "A6" records.
PP_PAN	8052h	N..19	PAN of the processed card.
PP_PANSEQNO	8053h	N2	*Application PAN Sequence Number* of the processed card.
PP_EMVDATA	8054h	B..512	List of EMV objects returned by the pinpad (in TLV format - see **section 7.1**).

RSP_DATID	Value	Format	Description
PP_CHNAME	8055h	A..26	Cardholder name of the processed card.
PP_GOXRES	8056h	N6	"**GOX**" response: EMV processing status. "**0**xxxxx" = Transaction offline approved; "**1**xxxxx" = Transaction declined; or "**2**xxxxx" = Transaction requires online approval. "x**1**xxxx" = Signature on paper. "xx**1**xxx" = Successful offline PIN verification. "xx**2**xxx" = PIN captured for online verification. "xxx**1**xx" = Cardholder verification performed on the mobile device (smartphone, for example) "xxxx00" = RFU.
PP_PINBLK	8057h	B8	Encrypted PIN.
PP_FCXRES	8058h	N3	"**FCX**" response: EMV processing status. "**0**xx" = Transaction approved; or "**1**xx" = Transaction declined. "x00" = RFU.
PP_ISRESULTS	8059h	B..50	*Issuer Script Results* (multiple of 5 - up to 10 results).
PP_BIGRAND	805Ah	B900	900 random bytes generated by the pinpad (used for testing only).
PP_LABEL	805Bh	S..16	Label of the application being processed (ICC EMV or CTLS).
PP_ISSCNTRY	805Ch	N3	*Issuer Country Code* of the processed card.
PP_CARDEXP	805Dh	N6	*Application Expiration Date* of the processed card, in the "YYMMDD" format.
PP_MFNAME	805Eh	A8	Name of a media file loaded on the pinpad, always in uppercase.
PP_DEVTYPE	8060h	N2	Device type used in the transaction: "00" = Card; "01" = Mobile device (i.e. smartphone); "02" = Keyring; "03" = Watch; "04" = Mobile tag; "05" = Bracelet; "06" = Mobile device case/sleeve; "10" = Tablet or e-reader; Other values = Future use.
----------	8061h	B16	Reserved.
PP_TLRMEM (†)	8062h	X4	Amount of available memory (in bytes) for loading EMV Table records using the "**TLR**" command.
PP_ENCKRAND	8063h	B256	Random key (K_{RAND}) encrypted by an RSA public key in PKCS # 1 format.

RSP_DATID	Value	Format	Description
---------	9000h to 9063h	----	Reserved range.
PP_KSNTDESPnn	9100h to 9163h	B10	DUKPT:TDES:PIN KSN, slot index #nn (from 00 to 99) **IMPORTANT**: Pay attention to hexadecimal values (**PP_KSNTDESP14** = 910Eh)!!
PP_KSNTDESDnn	9200h to 9263h	B10	DUKPT:TDES:DAT KSN, slot index #nn (from 00 to 99) **IMPORTANT**: Pay attention to hexadecimal values (**PP_KSNTDESD79** = 924Fh)!!
PP_TABVERnn	9300h to 9363h	A10	EMV Tables version correspondent to the Acquirer #nn (00 to 99). Index #00 corresponds to the "general" version for all Acquirers.

(†) See "**GIX**" command (**section 3.2.4**).

3.2. Control Commands

This section details the following commands related to general pinpad control:

CMD_ID	Meaning	Obsolete	Blocking	Abecs
"OPN"	Open Pinpad	☑	☐	☐
"OPN"	Open Pinpad (Secure)	☐	☐	☐
"GIN"	Get Information	☑	☐	☐
"GIX"	Get Information - Extended	☐	☐	☑
"DWK"	Define WK_{PAN}	☑	☐	☐
"CLO"	Close Pinpad	☐	☐	☐
"CLX"	Close Pinpad - Extended	☐	☐	☑

3.2.1. “OPN” command (classic)

☑ **Obsolete**
☐ **Blocking**
☐ **Abecs**

This command allocates hardware and software resources necessary for the pinpad operation.

A successful calling of this command is a prerequisite for all others described in this specification.

⚠ This command format is **obsolete**. The SPE must use the format described in **section 3.2.2**.

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= “**OPN**”).

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= “**OPN**”).
RSP_STAT	N3	See **section 3.1.1**.

➲ Examples

SPE requests “opening” of the pinpad.		
SPE ⇒	4F 50 4E	OPN
Operation is successful.		
⇐ PP	4F 50 4E 30 30 30	OPN000

3.2.2. "OPN" command (secure)

☐ **Obsolete**
☐ **Blocking**
☐ **Abecs**

This command performs the same functions as the **"OPN" (classic)** and also establishes the "Secure Communication" key between the SPE and the pinpad (see **section 5.2**).

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= "**OPN**").
CMD_LEN1	N3	Length of the following data.
OPN_OPMODE	N1	Operation mode (fixed **"0"**).
OPN_MODLEN	N3	Number of bytes represented in **OPN_MOD** (length ÷ 2), fixed "256" (see explanation in **section 5.2**).
OPN_MOD	H512	Modulus of the RSA key created by the SPE ($\mathbf{K_{MOD}}$).
OPN_EXPLEN	N1	Number of bytes represented in **OPN_EXP** (length ÷ 2).
OPN_EXP	H..6	Public exponent of the RSA key created by the SPE ($\mathbf{K_{PUB}}$).

➲ Response (Abecs Pinpad)

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**OPN**").
RSP_STAT	N3	See **section 3.1.1**.
RSP_LEN1	N3	Length of the following data.
OPN_CRKSLEN	N3	Number of bytes represented in **OPN_CRKSEC** (length ÷ 2), fixed "256".
OPN_CRKSEC	H512	Cryptogram (**CRKSEC**) generated using the provided public key, containing $\mathbf{K_{SEC}}$ (see format in **section 5.2.1**).

➲ Response (obsolete pinpad)

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**OPN**").
RSP_STAT	N3	See **section 3.1.1**.

⚠ If the pinpad returns this response format, it means that it does not yet follow this specification. In this specific case, there is no "Secure Communication" and, for compatibility reasons, the SPE shall not use Abecs Commands.

➲ Examples

SPE requests **K_{SEC}** key, providing a 256-byte RSA key module and a public exponent of value 13 (0Dh).

SPE ⇒	4F 50 4E 35 31 39 30 32 35 36 41 38 32 41 36 36 30 42 33 43 34 39 32 32 36 45 46 43 44 41 42 41 37 46 43 36 38 30 36 36 42 38 33 44 32 33 44 30 35 36 30 45 44 41 33 41 31 32 42 36 33 45 39 31 33 32 46 32 39 39 46 42 46 33 34 30 41 35 41 45 42 43 34 43 44 35 44 43 31 46 31 34 38 37 33 46 38 33 41 38 30 42 41 39 41 38 38 44 33 46 45 41 42 42 41 42 34 31 44 46 46 43 31 39 34 34 42 42 42 41 41 38 39 46 32 36 41 46 39 43 43 32 38 46 46 33 31 43 34 39 37 45 42 39 31 44 38 32 46 38 36 31 33 45 37 34 36 33 43 34 37 35 32 39 46 42 44 31 39 32 35 46 44 33 33 32 36 41 38 44 43 30 32 37 37 30 34 44 41 36 38 38 36 30 45 36 38 42 44 30 41 31 43 45 41 38 44 45 36 45 43 37 35 36 30 34 43 44 33 44 39 41 36 41 46 33 38 38 32 32 44 45 34 35 41 41 41 30 43 39 46 42 46 32 42 44 34 37 38 33 42 30 46 39 41 38 31 46 36 33 35 30 43 30 31 38 38 31 35 36 46 39 30 38 46 41 42 31 46 35 35 39 43 46 43 45 31 46 39 31 41 33 39 33 34 33 31 45 38 42 46 32 43 44 37 38 43 30 34 42 44 35 33 30 44 42 34 34 31 30 39 31 43 44 46 46 42 34 30 30 44 41 43 30 38 42 31 34 35 30 44 42 36 35 43 30 30 45 32 44 34 41 46 34 45 39 41 38 35 41 31 41 31 39 42 36 31 46 35 35 30 46 30 43 32 38 39 42 31 34 42 44 36 33 44 46 38 41 31 35 33 39 41 38 43 46 36 32 39 46 39 38 46 38 38 45 41 39 34 34 44 39 30 35 36 36 37 35 30 30 30 46 39 35 42 46 44 30 46 45 46 43 35 36 46 39 44 39 44 36 36 45 32 37 30 31 42 44 42 44 37 31 39 33 33 31 39 31 41 45 39 39 32 38 46 35 44 36 32 33 46 45 38 42 39 39 45 43 43 37 37 37 34 34 34 46 46 41 41 38 33 44 45 34 35 36 46 35 43 38 44 33 43 38 33 45 43 35 31 31 41 46 31 30 44	OPN5190256A82A66 0B3C49226EFCDABA 7FC68066B83D23D0 560EDA3A12B63E91 32F299FBF340A5AE BC4CD5DC1F14873F 83A80BA9A88D3FEA BBAB41DFFC1944BB BAA89F26AF9CC28F F31C497EB91D82F8 613E7463C47529FB D1925FD3326A8DC0 27704DA68860E68B D0A1CEA8DE6EC756 04CD3D9A6AF38822 DE45AAA0C9FBF2BD 4783B0F9A81F6350 C0188156F908FAB1 F559CFCE1F91A393 431E8BF2CD78C04B D530DB441091CDFF B400DAC08B1450DB 65C00E2D4AF4E9A8 5A1A19B61F550F0C 289B14BD63DF8A15 39A8CF629F98F88E A944D9056675000F 95BFD0FEFC56F9D9 D66E2701BDBD7193 3191AE9928F5D623 FE8B99ECC777444F FAA83DE456F5C8D3 C83EC511AF10D

Pinpad generates a random K_{SEC} (DB3B4D015432AB3223555A1F81759A94) and returns the cryptogram generated by the public key.

⇐ **PP**	4F 50 4E 30 30 30 35 31 35 32 35 36 34 45 35 38 30 35 45 35 41 43 46 33 42 45 34 41 33 46 44 32 37 33 30 30 45 36 38 32 44 44 42 30 32 38 44 43 34 33 32 32 33 44 36 44 32 45 35 39 44 42 31 32 42 43 42 35 32 44 32 33 38 44 31 38 37 35 43 46 31 39 41 36 39 46 45 34 30 35 32 42 37 46 45 44 30 31 36 30 41 44 46 33 30 30 36 44 38 44 36 36 31 35 36 41 41 31 41 30 41 35 35 45 32 46 31 41 30 34 35 33 32 32 46 45 44 35 39 34 35 42 32 46 34 41 37 41 36 45 36 36 43 38 44 32 46 41 39 37 34 37 39 44 33 31 42 31 30 36 46 45 43 31 41 35 39 33 37 30 31 38 34 41 43 36 33 37 33 42 31 30 35 33 44 41 39 42 45 37 44 43 30 31 42 32 41 41 31 38 43 32 30 38 45 31 43 30 37 37 39 43 30 43 43 44 37 44 34 34 39 36 45 35 33 32 36 45 39 38 41 45 37 34 34 43 43 43 35 38 43 41 37 42 34 36 33 30 44 39 36 44 44 33 37 46 42 37 42 37 39 44 36 46 42 41 37 39 33 30 31 38 43 39 32 43 36 31 35 31 36 39 33 39 43 43 41 31 32 44 31 39 32 34 31 34 36 31 36 30 35 44 35 38 39 30 38 32 42 42 35 45 44 37 39 45 35 41 45 37 32 30 43 39 44 43 43 30 37 32 35 30 46 45 45 35 32 37 44 31 38 41 44 38 41 42 33 37 34 39 45 32 45 45 30 44 34 38 44 39 42 43 32 45 30 41 45 44 37 35 41 44 37 34 39 45 31 31 41 33 37 39 43 33 37 42 36 38 34 30 31 30 34 38 41 44 37 39 44 45 32 35 34 45 30 42 34 35 45 31 34 33 45 42 44 30 37 39 37 43 38 33 46 37 41 44 38 38 44 32 35 35 46 31 39 31 35 33 43 38 30 42 31 35 39 42 45 41 34 46 35 45 36 30 34 45 46 41 39 38 44 30 39 31 39 33 46 42 39 42 45 34 46 45 39 32 32 42 43 31 44 31 42 46 44 39 37 39 31 45 37 37 36 34 43 36 32 35 41 45 33 45 38 35 42 44 43 43 38 39 30 33 42 44	OPN0005152564E58 05E5ACF3BE4A3FD2 7300E682DDB028DC 43223D6D2E59DB12 BCB52D238D1875CF 19A69FE4052B7FED 0160ADF3006D8D66 156AA1A0A55E2F1A 045322FED5945B2F 4A7A6E66C8D2FA97 479D31B106FEC1A5 9370184AC6373B10 53DA9BE7DC01B2AA 18C208E1C0779C0C CD7D4496E5326E98 AE744CCC58CA7B46 30D96DD37FB7B79D 6FBA793018C92C61 516939CCA12D1924 1461605D589082BB 5ED79E5AE720C9DC C07250FEE527D18A D8AB3749E2EE0D48 D9BC2E0AED75AD74 9E11A379C37B6840 1048AD79DE254E0B 45E143EBD0797C83 F7AD88D255F19153 C80B159BEA4F5E60 4EFA98D09193FB9B E4FE922BC1D1BFD9 791E7764C625AE3E 85BDCC8903BD

For validation purposes, this example considers the following value for the private exponent:

K_{PRV} =

```
40 AD D8 7A 79 A5 F9 8D 26 2C BD E2 60 0A 00 1F
79 FA 15 0D 68 2C 8C 7D 59 C9 4B 89 BF C5 12 22
7B 53 6A 97 31 3E 8F BD 2F 47 B5 F7 8F 66 F2 7B
E7 8E BC BE 55 8F 7D 88 58 7C E5 BD F2 15 D3 CD
63 AD 4B 0E BC 1C 44 6E 95 32 5F 87 DC F1 B0 37
DE 4B 39 77 FD 38 8C 4E 77 C0 5D 99 03 CF 18 AA
9B 6C 5D 28 DB C5 A3 69 3E 4C AA EE 27 8D D8 EE
0E E5 97 41 CC 06 8C 9C 74 98 70 2F 32 A6 87 67
6B A0 D1 02 AD F1 70 45 5D E2 6B 71 6E 0A C1 CA
13 93 71 D0 B5 27 5F 0B 93 F7 07 9F 2F 9C F0 1D
21 D6 C0 D4 1E 21 2E 20 FE 40 C1 E3 AF AF 73 47
3F 5B 7C 16 79 01 A9 5B 49 44 80 4E DC D6 8D 4C
A4 E2 C5 D3 3C BF 88 AC 42 71 2C ED 32 47 9A 03
6B 48 9F 38 23 D8 B8 63 FA 9C EB 9E 5A 4C ED AB
AD 25 19 11 D4 F9 20 D1 5D 72 B5 47 A0 AD 21 27
6E 9C FD 79 F8 7B 83 0C 32 B7 65 05 68 D8 EB D5
```


Using the RSA key with the K_{PRV} defined above, the following data block is obtained when "opening" **CRKSEC**:

CRKSEC cleartext =

```
00 02 FA 6D BD 58 30 43 21 4C A1 BA EA EA 54 F2
DB 72 2E 7F 96 41 89 7D C7 57 DB 31 6C 79 88 07
C1 27 AA 16 88 6D 4E 31 0A CC 97 1B 0B 2D 1F 22
60 DD B1 E7 15 17 AC 33 5F FB CD B3 16 C7 98 80
7B 78 BE 8B 96 BE 37 97 A0 3C BD 23 C8 7A 92 CD
26 BD C7 37 E3 8C 39 4C 96 D9 70 96 75 B1 FA 7C
49 2E E2 23 B7 1D BD 63 6E 87 FE A8 C0 46 F4 9C
F9 B4 45 FA 57 FA 6D BD 58 30 43 21 4C A1 BA EA
EA 54 F2 DB 72 2E 7F 96 41 89 7D C7 57 DB 31 6C
79 88 07 C1 27 AA 16 88 6D 4E 31 0A CC 97 1B 0B
2D 1F 22 60 DD B1 E7 15 17 AC 33 5F FB CB 78 BE
8B 96 BE 37 97 A0 3C BD 23 C8 7A 92 CD 26 BD C7
37 E3 8C 39 4C 96 D9 70 96 75 B1 FA 7C 49 2E E2
23 B7 1D BD 63 6E 87 FE A8 C0 46 F4 9C F9 B4 45
FA 57 6E 87 FE A8 C0 46 F4 9C F9 B4 45 FA 57 00
DB 3B 4D 01 54 32 AB 32 23 55 5A 1F 81 75 9A 94
```

K_{SEC} =

```
DB 3B 4D 01
54 32 AB 32
23 55 5A 1F
81 75 9A 94
```

3.2.3. "GIN" command

☑ **Obsolete**
☐ **Blocking**
☐ **Abecs**

This command obtains general information about the pinpad and its software (or firmware). If the information does not exist or does not apply for the pinpad model, it returns a blank field (spaces).

⚠ This command is **obsolete**. The SPE must use the "**GIX**" command for this functionality.

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= "**GIN**").
CMD_LEN1	N3	Length of the following data (fixed "002").
GIN_ACQIDX	N2	Acquirer Network index. If not "**00**", it requests information about the software/firmware responsible for processing transactions for the Acquirer Network number **GIN_ACQIDX**. If "**00**", it requests general information about the pinpad.

➲ Response (for GIN_ACQIDX = "00")

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**GIN**").
RSP_STAT	N3	See **section 3.1.1**.
RSP_LEN1	N3	Length of the following data (fixed "100").
GIN_MNAME	A20	Name of the pinpad manufacturer.
GIN_MODEL	A19	Model / hardware version, in a "**xx...xx;m...m**" format, where: ▪ "xx...xx" is the device name; and ▪ "m...m" is the memory capacity ("512KB", "1MB", "2MB", ...).
GIN_CTLSSUP	A1	If the pinpad supports CTLS, this field must contain the letter "**C**", otherwise a blank space.
GIN_SOVER	A20	Basic software or operating system version (free format).
GIN_SPECVER	A4	Specification version, in "V.VV" format (in this case, fixed "**2.12**")
GIN_MANVER	A16	"Manager" application version, in the format "VVV.VV YYMMDD".
GIN_SERNUM	A20	Pinpad serial number (free format, it depends on the manufacturer).

➲ Response (for GIN_ACQIDX = “02”)

Field Id.	Format	Description
RSP_ID	A3	Response code (= “**GIN**”).
RSP_STAT	N3	See **section 3.1.1**.
RSP_LEN1	N3	Length of the following data (fixed “042”).
GIN_ACQNAM	A8	Acquirer name (fixed “**Abecs**”)
GIN_KRNLVER	A12	ICC EMV kernel version.
GIN_APPVERS	A13	“Abecs” application version, in the format “VVV.VV YYMMDD”.
GIN_SPECVER	A4	Specification version, in “V.VV” format (in this case, fixed “**2.12**”)
GIN_RFU1	A3	RFU (blank spaces)
GIN_RFU2	N2	RFU (fixed “00”)

➲ Response (for GIN_ACQIDX = “03”)

Field Id.	Format	Description
RSP_ID	A3	Response code (= “**GIN**”).
RSP_STAT	N3	See **section 3.1.1**.
RSP_LEN1	N3	Length of the following data (fixed “042”).
GIN_ACQNAM	A6	Acquirer name (fixed “**Abecs**”)
GIN_KRNLVER	A4	ICC EMV kernel version.
GIN_CTLSVER	A4	CTLS EMV kernel version (general or entry point).
GIN_MCTLSVER	A3	CTLS EMV kernel version - MasterCard PayPass.
GIN_VCTLSVER	A3	CTLS EMV kernel version - VISA PayWave.
GIN_APPVERS	A13	“Abecs” application version, in the format “VVV.VV YYMMDD”.
GIN_SPECVER	A4	Specification version, in “V.VV” format (in this case, fixed “**2.12**”)
GIN_RFU3	A2	RFU (blank spaces)
GIN_DUKPT	A1	Identifies the presence of DUKPT:TDES:PIN in slot “01”: “**T**” = Key present; or “ ” (blank space) = Key absent.
GIN_RFU2	N2	RFU (fixed “00”)

➲ Response (for other GIN_ACQIDX)

Field Id.	Format	Description
RSP_ID	A3	Response code (= “**GIN**”).

Field Id.	Format	Description
RSP_STAT	N3	See **section 3.1.1**.
RSP_LEN1	N3	Length of the following data (fixed “042”).
GIN_ACQNAME	A20	Acquirer name (fixed “**Abecs**”)
GIN_APPVERS	A13	“Abecs” application version, in the format “VVV.VV YYMMDD”.
GIN_SPECVER	A4	Specification version, in “V.VV” format (in this case, fixed “**2.12**”)
GIN_RFU1	A3	RFU (blank spaces)
GIN_RFU2	N2	RFU (fixed “00”)

➲ Examples

SPE requests pinpad information for GIN_ACQIDX = “00”.

SPE ⇒	47 49 4E 30 30 32 30 30	GIN00200

Operation is successful.

⇐ PP	47 49 4E 30 30 30 31 30 30 43 59 47 4E 55 53 20 20 20 20 20 20 20 20 20 20 20 20 20 20 50 50 20 58 2D 31 3B 31 30 4D 42 20 20 20 20 20 20 20 20 43 38 30 36 35 58 41 30 37 37 58 30 30 36 30 58 20 20 20 20 20 32 2E 30 30 30 30 31 2E 30 33 20 31 33 30 37 31 35 20 20 20 30 30 31 31 30 31 30 31 30 33 30 30 30 30 30 20 20 20 20 20	GIN000100CYGNUS• ••••••••••••PP• X-1;10MB•••••••• C8065XA077X0060X •••••2.00001.03• 130715•••0011010 10300000•••••

SPE requests pinpad information for GIN_ACQIDX = “02”.

SPE ⇒	47 49 4E 30 30 32 30 32	GIN00202

Operation is successful.

⇐ PP	47 49 4E 30 30 30 30 34 32 41 62 65 63 73 20 20 20 56 31 2E 30 39 20 20 20 20 20 20 20 30 30 31 2E 30 33 20 31 33 30 37 31 35 32 2E 30 30 20 20 20 30 30	GIN000042Abecs•• •V1.09••••••001 .03•1307152.00•• •00

3.2.4. “GIX” command

☐ **Obsolete**
☐ **Blocking**
☑ **Abecs**

This command obtains general information about the pinpad and its software (or firmware), as well as the cryptographic keys loaded on it. If the information does not exist or does not apply for the pinpad model, it is simply not returned.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**GIX**”).
SPE_IDLIST	O	List of identifiers of data to be returned by the pinpad, which may include any of the identifiers listed in the response below. If this field is not provided, the pinpad will consider all objects identified with “(†)” in the table of **section 3.1.3.2**.

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= “**GIX**”).
RSP_STAT	M	Relevant return codes (see **section 3.1.1**): ↳ST_RSPOVRFL..... Response length exceeds the maximum allowed by the protocol.
PP_SERNUM(†)	MR	
PP_PARTNBR(†)	O	If information supported by the pinpad.
PP_MODEL(†)	MR	
PP_MNNAME(†)	MR	
PP_CAPAB(†)	MR	
PP_SOVER(†)	MR	
PP_SPECVER(†)	MR	
PP_MANVERS(†)	MR	
PP_APPVERS(†)	MR	
PP_GENVERS(†)	MR	
PP_KRNLVER(†)	MR	
PP_CTLSVER(†)	MR	
PP_MCTLSVER(†)	MR	
PP_VCTLSVER(†)	MR	
PP_AECTLSVER(†)	MR	

Field Id.	Presence	Description / Remark
PP_DPCTLSVER(†)	MR	
PP_PUREVER(†)	MR	
PP_DSPTXTSZ(†)	MR	
PP_DSPGRSZ(†)	O	Only if the pinpad has a graphic display.
PP_MFSUP(†)	O	Only if the pinpad supports the "**DSI**" command.
PP_MKTDESP(†)	MR	
PP_MKTDESD(†)	MR	
PP_DKPTTDESP(†)	MR	
PP_DKPTTDESD(†)	MR	
PP_TLRMEM(†)	MR	
PP_KSNTDESPnn	O	Only if the pinpad has a DUKPT:TDES:PIN key loaded in slot #nn.
PP_KSNTDESDnn	O	Only if the pinpad has a DUKPT:TDES:DAT key loaded in slot #nn.
PP_TABVERnn	O	Value according to rules defined for the "**GTS**" command (see **section 3.5.1**).
PP_BIGRAND	MR	Used only for protocol tests.

➲ Examples

SPE requests PP_SERNUM, PP_MNNAME, PP_DKPTTDESP, PP_KSNTDESP01 and PP_KSNTDESP14.

SPE ⇒	47 49 58 30 31 34 00 01 00 0A 80 01 80 04 80 34 91 01 91 0E	GIX014....€.€.€4 ,.,.

Pinpad returns the information but does not return the KSN of DUKPT:TDES:PIN #14, as this key is not loaded.

⇐ PP	47 49 58 30 30 30 31 35 31 80 01 00 0C 39 39 31 32 37 34 33 36 36 31 35 35 80 04 00 0D 48 45 4D 49 53 50 48 45 52 45 53 20 20 80 34 00 64 30 31 31 31 30 30 31 31 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 91 01 00 0A FF FF F9 13 25 00 43 20 04 43	GIX000151€...991 274366155€...HEM ISPHERES••€4.d01 1100110000000000 0000000222222222 2222222222222222 2222222222222222 2222222222222222 2222222222222222 22'...ÿÿù.%.C .C

SPE sends the command without parameters**.**

SPE ⇒	47 49 58	GIX

Pinpad returns PP_SERNUM, PP_PARTNBR, PP_MODEL, PP_MNNAME, PP_CAPAB, PP_SOVER, PP_SPECVER, PP_MANVERS, PP_APPVERS, PP_GENVERS, PP_KRNLVER, PP_CTLSVER, PP_MCTLSVER, PP_VCTLSVER, PP_AECTLSVER, PP_DPCTLSVER, PP_PUREVER, PP_DSPTXTSZ, PP_MKTDESP, PP_MKTDESD, PP_DKPTTDESP, PP_DKPTTDESD and PP_TLRMEM.

The fields PP_DSPGRSZ and PP_MFSUP are not supported by this pinpad.

⇐ PP

```
47 49 58 30 30 30 36 35 37 80 01 00 08 30 31 39  GIX000657€...019
32 37 36 31 32 80 02 00 0D 50 50 32 30 30 30 2F  27612€...PP2000/
42 54 2F 55 53 42 80 03 00 0A 50 50 32 30 30 30  BT/USB€...PP2000
3B 38 4D 42 80 04 00 0C 53 55 42 44 49 56 49 53  ;8MB€...SUBDIVIS
49 4F 4E 53 80 05 00 0A 31 30 30 30 30 30 30 30  IONS€...10000000
30 30 80 06 00 07 53 4F 3A 33 2E 34 35 80 07 00  00€...SO:3.45€..
04 32 2E 30 30 80 08 00 10 30 30 32 2E 31 35 20  .2.00€...002.15•
31 34 30 37 32 33 20 20 20 80 09 00 10 30 30 32  140723•••€...002
2E 30 38 20 31 34 30 32 32 38 20 20 20 80 0A 00  .08•140228•••€..
10 30 30 32 2E 31 37 20 31 34 30 36 30 36 20 20  .002.17•140606••
20 80 10 00 06 33 2E 30 30 36 41 80 11 00 04 31  •€...3.006A€...1
2E 35 34 80 12 00 06 4D 43 31 2E 30 38 80 13 00  .54€...MC1.08€..
06 56 53 31 2E 30 32 80 14 00 06 41 45 31 2E 30  .VS1.02€...AE1.0
35 80 15 00 07 44 43 31 2E 30 30 42 80 16 00 08  5€...DC1.00B€...
50 55 52 45 56 34 30 32 80 20 00 04 30 34 32 30  PUREV402€•..0420
80 32 00 64 30 30 31 30 30 30 30 30 30 30 30 31  €2.d001000000001
30 30 30 30 30 30 31 30 30 30 30 31 30 30 32 32  0000001000010022
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32  2222222222222222
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32  2222222222222222
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32  2222222222222222
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32  2222222222222222
32 32 32 32 32 32 32 32 80 33 00 64 30 30 31 30  22222222€3.d0010
30 30 30 30 30 30 30 30 30 30 30 30 30 30 31 30  0000000000000010
30 30 30 30 30 30 32 32 32 32 32 32 32 32 32 32  0000002222222222
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32  2222222222222222
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32  2222222222222222
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32  2222222222222222
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32  2222222222222222
80 35 00 64 30 31 31 31 30 31 30 30 30 31 31 31  €5.d011101000111
31 30 30 31 31 30 30 31 30 30 30 30 30 31 30 30  1001100100000100
30 30 30 30 30 30 30 30 32 32 32 32 32 32 32 32  0000000022222222
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32  2222222222222222
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32  2222222222222222
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32  2222222222222222
32 32 32 32 32 32 32 32 80 36 00 64 30 30 31 30  22222222€6.d0010
30 30 30 30 30 31 30 31 30 30 30 31 30 30 30 31  0000010100010001
30 30 30 30 30 31 30 30 30 30 30 30 30 31 30 30  0000010000000100
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32  2222222222222222
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32  2222222222222222
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32  2222222222222222
32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32  2222222222222222
80 62 00 04 00 01 86 A0                          €b....†
```

3.2.5. "DWK" command

☑ **Obsolete**
☐ **Blocking**
☐ **Abecs**

This command allows the SPE to enable the "Encrypted PAN" mode, preventing the card number from appearing in cleartext through the pinpad serial protocol, according to the process detailed in **section 5.3**.

This command establishes the key (**WK_{PAN}**) to be used in the process and can be called at any time after successful execution of "**OPN**". From that moment on, the pinpad starts to work in "Encrypted PAN" mode, until the **"CLO/CLX"** command is called.

⚠ The "Encrypted PAN" mode is **obsolete** and has been replaced by the "Secure Communication" method, described in **section 5.2**. It should only be used by the SPE if it identifies that the pinpad does not yet comply with this specification.

⚠ The "Encrypted PAN" mode is not accepted by the pinpad if the SPE is already using the "Secure Communication" method described in **section 5.2**.

➲ Command (Mode 1)

Field Id.	Format	Description
CMD_ID	A3	Command code (= "**DWK**").
CMD_LEN1	N3	Length of the following data (fixed "036").
DWK_MODE	N1	Mode: **"1"** = Extern **WK_{PAN}** encrypted by a MK.
DWK_METHOD	N1	Encryption mode: **"1"** = MK/WK:TDES:DAT
DWK_MKIDX	N2	Slot index of the MK to be used.
DWK_WKPAN	H32	**WK_{PAN}** encrypted by the MK.

➲ Response (Mode 1)

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**DWK**").
RSP_STAT	N3	Relevant return codes (see **section 3.1.1**): ↳ST_ERRKEY.....MK not present in the pinpad. ↳ST_INVPARM .....Slot index (**DWK_MKIDX**) is outside the range supported by the pinpad. ↳ST_INVCALL .....Pinpad is using "Secure Communication" mode.

➲ Command (Mode 2)

Field Id.	Format	Description
CMD_ID	A3	Command code (= "**DWK**").
CMD_LEN1	N3	Length of the following data (fixed "263").
DWK_MODE	N1	Mode: **"2"** = Random **WK_{PAN}** (TDES) generated by the pinpad.
DWK_RSAMOD	H256	RSA public key modulus created by the SPE (**K_{MOD}** - fixed 128 bytes / 1024 bits). IMPORTANT: The first byte of the modulus must be bigger than 54h, due to the data block format (see section 5.3.3).
DWK_RSAEXP	H6	RSA public key exponent created by the SPE (**K_{PUB}** - typically "000003" or "010001").

➲ Response (Mode 2)

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**DWK**").
RSP_STAT	N3	Relevant return codes (see **section 3.1.1**): ↳ST_INVCALL .....Pinpad is using "Secure Communication" mode.
RSP_LEN	N3	Length of the following data (fixed "256").
DWK_CRYPT	H256	RSA cryptogram containing the random **WK_{PAN}**, according to the definition in **section 5.3**.

➲ Examples

SPE initiates the "Encrypted PAN" mode 2, providing an RSA public key.

SPE ⇒	44 57 4B 32 36 33 32 43 30 45 34 45 36 41 41 44 39 44 43 38 31 45 32 45 42 46 38 41 43 31 32 36 45 37 45 45 45 36 35 36 38 30 38 39 38 42 42 41 43 33 30 30 36 33 44 43 44 35 34 33 44 37 30 35 30 34 30 45 39 31 36 44 39 33 45 45 33 31 36 42 39 45 43 34 39 32 42 37 39 36 46 31 37 32 31 34 32 35 46 30 46 30 32 38 38 33 34 32 35 31 41 41 44 35 31 43 45 42 31 37 38 33 33 30 38 45 43 37 44 35 30 37 32 44 38 34 38 31 33 42 44 41 35 39 42 33 31 36 31 43 42 34 38 37 39 34 36 34 45 42 35 41 46 37 31 39 36 39 38 36 35 46 44 33 34 37 34 35 41 37 31 31 44 31 44 41 33 44 44 42 34 44 32 39 44 32 39 44 30 34 32 32 43 36 45 31 37 43 32 35 46 31 37 43 30 42 35 42 33 39 45 36 38 38 43 34 44 30 36 31 32 33 44 44 42 35 46 35 35 38 45 46 30 33 31 36 42 33 46 37 34 34 43 37 30 37 31 46 32 39 37 39 31 30 31 30 30 30 31	DWK2632C0E4E6AAD 9DC81E2EBF8AC126 E7EEE65680898BBA C30063DCD543D705 040E916D93EE316B 9EC492B796F17214 25F0F028834251AA D51CEB1783308EC7 D5072D84813BDA59 B3161CB4879464EB 5AF71969865FD347 45A711D1DA3DDB4D 29D29D0422C6E17C 25F17C0B5B39E688 C4D06123DDB5F558 EF0316B3F744C707 1F29791010001

Pinpad generates a random **WK_{PAN}** (2A525553482A43524F4E49434C45532A) and returns it encrypted using the provided public key.

⇐ **PP**	44 57 4B 30 30 30 32 35 36 42 37 45 30 42 37 38 41 39 34 42 30 32 42 34 38 30 32 32 38 43 39 33 44 35 42 39 31 31 42 41 33 38 33 37 35 33 38 41 45 38 41 42 45 46 44 46 38 38 41 41 46 30 42 46 36 46 33 34 38 35 34 39 31 30 30 41 38 34 30 45 35 38 30 41 41 46 36 35 31 46 33 35 34 44 33 39 31 39 32 43 30 30 38 33 36 44 39 30 35 32 32 46 44 34 35 32 38 39 46 32 35 42 43 43 33 41 31 30 45 41 43 35 35 35 32 31 46 35 35 30 37 34 41 37 38 37 34 34 39 42 38 34 42 43 36 44 42 32 31 39 32 39 44 37 34 33 32 45 38 33 36 45 44 41 30 39 46 46 41 41 32 30 42 33 39 43 45 44 36 38 37 42 37 35 37 39 45 36 31 46 30 44 30 35 39 45 35 32 33 42 38 41 35 42 41 43 36 31 45 46 39 41 30 46 41 32 39 37 32 38 30 41 32 31 41 41 38 44 34 34 35 42 32 42 45 35 42 45 34 34 35 44 41 38 39 30 41 43 36 42 41 37 39 30 30	DWK000256B7E0B78 A94B02B480228C93 D5B911BA3837538A E8ABEFDF88AAF0BF 6F348549100A840E 580AAF651F354D39 192C00836D90522F D45289F25BCC3A10 EAC55521F55074A7 87449B84BC6DB219 29D7432E836EDA09 FFAA20B39CED687B 7579E61F0D059E52 3B8A5BAC61EF9A0F A297280A21AA8D44 5B2BE5BE445DA890 AC6BA7900

For validation purposes, this example considers the following value for the private exponent:

K_{PRV} =

```
65 3C BD C3 95 AC 21 8F 53 81 A3 ED D8 88 4D DE
73 07 70 01 AF 91 54 F5 42 BA 9F B4 3E AA 92 AB
27 41 D6 35 AB 46 D3 F0 39 3F 90 C8 27 E9 74 1B
44 18 FA 10 52 3E C9 58 63 59 85 A9 78 EB AC 19
E4 25 CE 7F 6B 78 66 7E 9C C1 85 C8 1A 0B F2 FF
A7 4A CC 33 FF A3 6F DB 95 66 80 12 FF 32 4E BD
58 04 60 C3 2D 76 61 8B E8 16 98 61 F5 33 2B 83
5C FC 31 1F 7C C5 41 65 87 0D 78 9D 6B 72 68 F1
```


3.2.6. “CLO” command

☐ **Obsolete**
☐ **Blocking**
☐ **Abecs**

This command releases the hardware and software resources allocated by the pinpad and finalizes “Secure Communication” or “Encrypted PAN” processes.

It is recommended that the SPE use this command at the end of a transaction processing.

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= “**CLO**”).
CMD_LEN1	N3	Length of the following data (fixed “032”).
CLO_MSG	S32	32-character message to be displayed on the pinpad display after executing the command, already formatted for 2 rows and 16 columns.

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= “**CLO**”).
RSP_STAT	N3	See **section 3.1.1**.

➲ Examples

SPE requests the “closing” of the pinpad, leaving the message “FORCE TEN @STORE” / “THANK YOU!” on the display.

SPE ⇒	43 4C 4F 30 33 32 46 4F 52 43 45 20 54 45 4E 20 40 53 54 4F 52 45 20 20 20 54 48 41 4E 4B 20 59 4F 55 21 20 20 20	CLO032FORCE•TEN• @STORE•••THANK•Y OU!•••

Operation is successful.

⇐ PP	43 4C 4F 30 30 30	CLO000

3.2.7. “CLX” command

☐ **Obsolete**
☐ **Blocking**
☑ **Abecs**

This command is equivalent to “**CLO**”, however it uses a free format message (allowing the use of all the equipment's display resources) or allows the presentation of a media file (if supported).

This command always returns immediately (it is not blocking), even if the media file informed contains animation (or video), which will be presented while the pinpad does not receive a new command.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**CLX**”).
SPE_DSPMSG	O	Message to be left on the pinpad display after executing the command.
SPE_MFNAME	O	Name of the media file to be presented after executing the command.

NOTES:

⇒ If no parameters are provided, the display is simply erased.

⇒ **SPE_MFNAME** has priority over **SPE_DSPMSG**, that is, if **SPE_MFNAME** is provided and the reported media file exists, **SPE_DSPMSG** is ignored.

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= “**CLX**”).
RSP_STAT	M	See **section 3.1.1**.

➲ Examples

SPE requests the “closing” of the pinpad, leaving a three-line message on the display.

SPE ⇒	43 4C 58 30 34 31 00 1B 00 25 50 52 45 53 54 4F 20 53 48 4F 50 0D 54 48 41 4E 4B 20 59 4F 55 0D 41 4E 44 20 43 4F 4D 45 20 41 47 41 49 4E 21	CLX041...%PRESTO •SHOP.THANK•YOU. AND•COME•AGAIN!

Operation is successful.

⇐ PP	43 4C 58 30 30 30	CLX000

3.3. Basic Commands

We call here “basic commands” those intended for simple access to peripherals and pinpad resources.

The following commands are covered in this section:

CMD_ID	Meaning	Obsolete	Blocking	Abecs
“CEX”	Check Event - Extended	☐	☑	☑
“CHP”	Chip Direct Access	☐	☑	☐
“CKE”	Check Event	☑	☑	☐
“DEX”	Display Message - Extended	☐	☐	☐
“DSP”	Display Message	☐	☐	☐
“EBX”	Encrypt Buffer - Extended	☐	☐	☑
“ENB”	Encrypt Buffer	☑	☐	☐
“GCD”	Get Clear Data	☐	☑	☑
“GDU”	Get DUKPT Serial Number	☑	☐	☐
“GKY”	Get Key	☑	☑	☐
“GPN”	Get Encrypted PIN	☐	☑	☐
“GTK”	Get Tracks	☐	☐	☑
“MNU”	Prompt Menu	☐	☑	☑
“RMC”	Remove Card	☐	☑	☐

3.3.1. “CEX” command

☐ **Obsolete**
☑ **Blocking**
☑ **Abecs**

This command waits for a certain event to occur on the pinpad. The following events can be checked:

• Pressing a (non-numeric) key;

• Swiping a magnetic card;

• Inserting/removing an ICC; and

• Presenting a CTLS.

In the case of magnetic card swiping, the tracks are returned incomplete, according to the security process described in **section 5.4**. To obtain the complete tracks (in cleartext or encrypted), one shall use the **“GTK”** command.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**CEX**”).
SPE_CEXOPT	M	Event to be checked by the pinpad: “**0**xxxxx” = Ignore keys; “**1**xxxxx” = Verify key pressing. “x**0**xxxx” = Ignore magnetic card; “x**1**xxxx” = Verify magnetic card swiping. “xx**0**xxx” = Ignore ICC; “xx**1**xxx” = Verify ICC insertion; “xx**2**xxx” = Verify ICC removal. “xxx**0**xx” = Ignore CTLS (do not activate antenna); “xxx**1**xx” = Activate antenna and verify CTLS presence. “xxxx00” = RFU.
SPE_TIMEOUT	O	Maximum time to wait for an event.
SPE_PANMASK	O	Definitions for PAN masking in the response fields **PP_TRK1INC**, **PP_TRK2INC** and **PP_TRK3INC**. If absent, there is no masking.

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= “**CEX**”).
RSP_STAT	M	See **section 3.1.1**.

Field Id.	Presence	Description / Remark
PP_EVENT	M	Event identification: “00” = [OK/ENTER] key pressed; “02” = [↑] key pressed; “03” = [↓] key pressed; “04” = [F1] key pressed; “05” = [F2] key pressed; “06” = [F3] key pressed; “07” = [F4] key pressed; “08” = [CLEAR] key pressed; “13” = [CANCEL] key pressed; “90” = A magnetic card was swiped; “91” = ICC removed (or already absent); “92” = ICC inserted (or already present); “93” = CTLS not detected in 2 (two) minutes; and “94” = CTLS detected.
PP_TRK1INC	O	Incomplete Track 1, if read from the magnetic card. PAN may be masked according to **SPE_PANMASK**.
PP_TRK2INC	O	Incomplete Track 2, if read from the magnetic card. PAN may be masked according to **SPE_PANMASK**.
PP_TRK3INC	O	Incomplete Track 3, if read from the magnetic card. PAN may be masked according to **SPE_PANMASK**.

⚠ If a magnetic card has been swiped (**PP_EVENT** = “90”), but no track could be read, **RSP_STAT** = ↳ST_OK and **PP_TRK1INC**, **PP_TRK2INC** and **PP_TRK3INC** fields will not be returned. This differs from the **“CKE”** command behavior, which returns RSP_STAT = ↳ST_MCDATAERR in this case.

➲ Examples

SPE requests only the magnetic card swiping event.

SPE ⇒	43 45 58 30 31 30 00 06 00 06 30 31 30 30 30 30	CEX010....010000

Pinpad reports card swipe, but only Track 2 is read.

⇐ PP	43 45 58 30 30 30 30 33 34 80 40 00 02 39 30 80 42 00 18 34 33 31 33 30 33 32 39 32 39 38 33 30 30 31 31 3D 31 35 30 38 36 30 31	CEX000034€@..90€ B..4313032929830 011=1508601

3.3.2. “CHP” command

☐ **Obsolete**
☑ **Blocking**
☐ **Abecs**

This command allows direct access to an ICC (main coupler or SAM) as well as a CTLS.

Additionally, this command makes it possible to capture a PIN for direct verification on the card, regardless of the technology (ICC, SAM or CTLS).

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= “**CHP**”).
CMD_LEN1	N3	Length of the following data.
CHP_SLOT	N1	Identification of the card/coupler to be used: **“0”** = ICC in the main coupler; **“1”** = SAM in socket #1; ... **“8”** = SAM in socket #8; and **“9”** = CTLS.
CHP_OPER	N1	Operation to be performed: **“0”** = Power off card (for CTLS, deactivate antenna); **“1”** = Power on card (for CTLS, first activate antenna and then the card); **“2”** = Exchange command with the card; and **“3”** = Verify PIN directly on the card.
CHP_CMDLEN	N3	Number of bytes represented in **CHP_CMD** (length ÷ 2). This field is "000" when **CHP_OPER** = "0" or "1".
CHP_CMD	H..520	Command to be sent to the card. If **CHP_OPER** = “2”, the following format are accepted: CLA INS P1 P2 CLA INS P1 P2 Le CLA INS P1 P2 Lc XX XX ... XX CLA INS P1 P2 Lc XX XX ... XX Le If **CHP_OPER** = “3”, only the first four bytes of the command to be sent to the card (CLA INS P1 P2) shall be provided, as the rest is automatically assembled according to the pinblock format (**CHP_PINFMT**).
CHP_PINFMT	N1	Pinblock format (only if **CHP_OPER** = “3”): **“0”** = 0Th PPh PPh ... FFh (8 bytes, 4 to 12-digit PIN); **“1”** = 2Th PPh PPh ... FFh (8 bytes, 4 to 12-digit PIN); **“2”** = PPh PPh PPh ... FFh (8 bytes, 4 to 12-digit PIN); and **“9”** = Sequence of ASCII numeric digits (variable length).
CHP_PINMSG	S32	Message to be presented at the time of the PIN capture (only if **CHP_OPER** = “3”), 2 rows by 16 columns.

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**CHP**").
RSP_STAT	N3	Relevant return codes (see **section 3.1.1**): ↳ST_CANCEL .....Cardholder pressed [CANCEL]. ↳ST_NOCARD.....No card present in coupler/antenna. ↳ST_DUMBCARD .....ICC inserted but does not answer ("mute") (it does not apply to CTLS). ↳ST_ERRCARD.....Communication error between the pinpad and the card. ↳ST_TIMEOUT.....PIN capture timeout (**CHP_OPER** = "3").
RSP_LEN1	N3	Length of the following data.
CHP_RSPLEN	N3	Number of bytes represented in **CHP_RSP** (length ÷ 2).
CHP_RSP	H..514	Card response: If **CHP_OPER** = "0", not present (**CHP_RSPLEN** is always "000"). If **CHP_OPER** = "1", it is the card complete ATR. If **CHP_OPER** = "2" or "3", it is the response to the command, followed by SW1 and SW2 bytes.

➲ Notes

- The pinpad **will not** resolve internally the return statuses 61xxh and 6Cxxh of T = 0 cards, thus the SPE must be prepared to deal with these two cases externally.
- The SPE shall always disable the antenna when it finishes processing a CTLS.

➲ Examples

SPE requests the activation of the ICC in the main coupler.

SPE ⇒	43 48 50 30 30 35 30 31 30 30 30	CHP00501000

Operation is successful and the pinpad returns the card's ATR (3B29008072A4456400FF0010).

⇐ PP	43 48 50 30 30 30 30 32 37 30 31 32 33 42 32 39 30 30 38 30 37 32 41 34 34 35 36 34 30 30 46 46 30 30 31 30	CHP0000270123B29008072A4456400FF0010

SPE sends the selection command (SELECT) using a MasterCard AID.

SPE ⇒	43 48 50 30 32 39 30 32 30 31 32 30 30 41 34 30 34 30 30 30 37 41 30 30 30 30 30 30 30 30 34 31 30 31 30	CHP0290201200A4040007A0000000041010

Operation is successful, and the card returns status bytes 6132h.

⇐ PP	43 48 50 30 30 30 30 30 37 30 30 32 36 31 33 32	CHP0000070026132

Since the card returned 61xxh (T = 0 protocol), the SPE sends a GET RESPONSE command to the card.		
SPE ⇒	43 48 50 30 31 35 30 32 30 30 35 30 30 43 30 30 30 30 30 33 32	CHP0150200500C00 00032
Operation is successful, with the card returning the response to the SELECT command.		
⇐ PP	43 48 50 30 30 30 31 30 37 30 35 32 36 46 33 30 38 34 30 37 41 30 30 30 30 30 30 30 30 34 31 30 31 30 41 35 32 35 35 30 30 41 34 44 36 31 37 33 37 34 36 35 37 32 34 33 36 31 37 32 36 34 38 37 30 31 30 31 35 46 32 44 30 36 37 30 37 34 36 35 36 45 36 35 37 33 39 46 31 31 30 31 30 31 39 46 31 32 30 36 34 33 37 32 36 35 36 34 36 39 37 34 39 30 30 30	CHP0001070526F30 8407A00000000410 10A525500A4D6173 7465724361726487 01015F2D06707465 6E65739F1101019F 1206437265646974 9000

SPE requests PIN verification directly on the card (format "1").		
SPE ⇒	43 48 50 30 34 36 30 33 30 30 34 30 30 32 30 30 30 30 30 31 41 4D 4F 55 4E 54 3A 24 39 2C 39 39 39 2C 39 39 45 4E 54 45 52 20 59 4F 55 52 20 50 49 4E 20 20	CHP0460300400200 0001AMOUNT:$9,99 9,99ENTER•YOUR•P IN••
Card returns 6A86h.		
⇐ PP	43 48 50 30 30 30 30 30 37 30 30 32 36 41 38 36	CHP0000070026A86

3.3.3. "CKE" command

☑ **Obsolete**
☑ **Blocking**
☐ **Abecs**

This command waits for a certain event to occur on the pinpad. The following events can be checked:

- Pressing a (non-numeric) key;
- Swiping a magnetic card;
- Inserting/removing an ICC; and
- Presenting a CTLS.

⚠ This command is **obsolete**, the SPE shall use "**CEX**" instead.

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= "**CKE**")
CMD_LEN1	N3	Length of the following data ("003" or "004")
CKE_KEY	N1	Controls a keystroke event. "**0**" = Ignore keys. "**1**" = Check key press.
CKE_MAG	N1	Controls magnetic card swipe event. "**0**" = Ignore magnetic card. "**1**" = Check the card swipe.
CKE_ICC	N1	Controls ICC insertion/removal event. "**0**" = Ignores ICC. "**1**" = Checks ICC insertion. "**2**" = Checks ICC removal.
CKE_CTLS (optional!)	N1	Controls CTLS presentation event. "**0**" = Does not activate the antenna. "**1**" = Activates the antenna and checks for the presence of a CTLS.

➲ Response (for key pressing)

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**CKE**")
RSP_STAT	N3	See **section 3.1.1.**
RSP_LEN1	N3	Length of the following data (fixed "003")
CKE_EVENT	N1	Event identification: **"0"**

Field Id.	Format	Description
CKE_KEYCODE	N2	Pressed key code: "00" = [OK/ENTER] "04" = [F1] "05" = [F2] "06" = [F3] "07" = [F4] "08" = [CLEAR] "13" = [CANCEL]

➲ Response (for magnetic card)

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**CKE**")
RSP_STAT	N3	Relevant return codes (see **section 3.1.1**): ↳ST_MCDATAERR .....Magnetic card event detected, but there was a reading error (no track could be read).
RSP_LEN1	N3	Length of the following data (fixed "225").
CKE_EVENT	N1	Event identification: **"1"**
CKE_TRK1LEN	N2	Length of Track 1.
CKE_TRK1	A76	Track 1 (without the sentinels and with the format byte - first alphanumeric character), left aligned with trailing spaces.
CKE_TRK2LEN	N2	Length of Track 2.
CKE_TRK2	A37	Track 2 (without the sentinels), left aligned with trailing spaces.
CKE_TRK3LEN	N3	Length of Track 3.
CKE_TRK3	A104	Track 3 (without the sentinels), left aligned with trailing spaces.

⚠ If the pinpad is in "Encrypted PAN" mode (see **section 5.3**), the PANs on the tracks return encoded using the **WK_{PAN}** key.

⚠ If the pinpad is in "Encrypted PAN" mode, **CKE_TRK3LEN** is not filled, as Track 2 can reach up to 40 characters (see explanation in **section 5.3**)!!

➲ Response (for ICC)

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**CKE**")
RSP_STAT	N3	See **section 3.1.1.**
RSP_LEN1	N3	Length of the following data (fixed "002")
CKE_EVENT	N1	Event identification: **"2"**
CKE_ICCSTAT	N1	**"0"** = ICC absent; or **"1"** = ICC present.

➲ Response (for CTLS)

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**CKE**")
RSP_STAT	N3	See **section 3.1.1.**
RSP_LEN1	N3	Length of the following data (fixed "002")
CKE_EVENT	N1	Event identification: **"3"**
CKE_CTLSSTAT	N1	**"0"** = CTLS was not detected in 2 (two) minutes. **"1"** = CTLS was detected.

➲ Examples

SPE asks the pinpad to wait for any of the four possible events.

SPE ⇒	43 4B 45 30 30 34 31 31 31 31	CKE0041111

A magnetic card is swiped on the pinpad, which returns its tracks 1 and 2.

⇐ PP	43 4B 45 30 30 30 32 32 35 31 37 34 42 35 31 34 38 36 38 32 32 32 32 32 32 32 32 37 37 5E 41 4C 45 58 20 4C 49 46 45 53 4F 4E 20 20 20 20 20 20 20 20 20 20 20 20 5E 32 31 31 32 32 30 31 39 38 37 36 30 30 30 30 30 30 30 30 30 30 30 34 34 39 37 30 30 30 30 30 20 20 33 37 35 31 34 38 36 38 32 32 32 32 32 32 32 32 37 37 3D 31 35 30 36 32 30 31 30 30 30 30 39 38 37 36 34 34 39 37 30 30 30 30	CKE000225174B514 8682222222277^AL EX•LIFESON•••••• ••••••^211220198 7600000000000449 700000••37514868 2222222277=15062 0100009876449700 00

SPE asks the pinpad to wait only for the keystroke event (also not sending the optional **CKE_CTLS** field).

SPE ⇒	43 4B 45 30 30 33 31 30 30	CKE003100

The F1 key is pressed on the pinpad.

⇐ PP	43 4B 45 30 30 30 30 30 33 30 30 34	CKE000003004

3.3.4. “DEX” command

☐ **Obsolete**
☐ **Blocking**
☐ **Abecs**

This command sends a message to the pinpad display in free format, allowing a better use of the equipment's display capabilities.

The display is erased before the message is presented, so previous messages are not kept.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	A3	Command code (= “**DEX**”).
CMD_LEN1	N3	Length of the following data.
DEX_MSGLEN	N3	Length of **DEX_MSG**.
DEX_MSG	S..160	Message to be presented, which may exceptionally contain the **CR (0Dh)** control character for line break.

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	A3	Response code (= “**DEX**”).
RSP_STAT	N3	See **section 3.1.1**.

➲ Examples

SPE sends a four-line message for presentation on the display.

SPE ⇒	44 45 58 30 34 31 30 33 38 46 72 65 65 7A 65 20 74 68 69 73 20 6D 6F 6D 65 6E 74 0D 41 20 6C 69 74 74 6C 65 0D 62 69 74 20 6C 6F 6E 67 65 72	DEX041038Freeze• this•moment.A•li ttle•bit•longer

Operation is successful.

⇐ PP	44 45 58 30 30 30	DEX000

3.3.5. “DSP” command

☐ **Obsolete**
☐ **Blocking**
☐ **Abecs**

This command sends a message to the pinpad display in a fixed format of 2 rows and 16 columns.

The display is erased before the message is presented, so previous messages are not kept.

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= “**DSP**”)
CMD_LEN1	N3	Length of the following data (fixed “032”)
DSP_MSG	S32	32-character message to be presented on the display, already formatted for 2 rows and 16 columns.

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= “**DSP**”)
RSP_STAT	N3	See **section 3.1.1.**

➲ Examples

SPE sends the message "OPERATION ERROR" / "CODE: 2112/76", correctly formatted for presentation on the display in 2 rows and 16 columns.

SPE ⇒	44 53 50 30 33 32 4F 50 45 52 41 54 49 4F 4E 20 45 52 52 4F 52 20 43 4F 44 45 3A 20 32 31 31 32 2F 37 36 20 20 20	DSP032OPERATION• ERROR•CODE:•2112 /76•••

Operation is successful.

⇐ PP	44 53 50 30 30 30	DSP000

3.3.6. “EBX” command

☐ **Obsolete**
☐ **Blocking**
☑ **Abecs**

This command encrypts a generic data block (up to 256 bytes) using a “data” key (MK/WK or DUKPT), in ECB or CBC mode.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**EBX**”).
SPE_DATAIN	M	Data block to be encrypted, with a multiple of 8 (eight) size, maximum of 256 bytes.
SPE_MTHDDAT	M	Identification of the encryption mode to be used: **“10”** = MK/WK:TDES:DAT (**ECB** block encryption); **“11”** = MK/WK:TDES:DAT (**CBC** block encryption); **“50”** = DUKPT:TDES:DAT#3 (**ECB** block encryption**,** see **section 5.1.1**); and **“51”** = DUKPT:TDES:DAT#3 (**CBC** block encryption**,** see **section 5.1.1**).
SPE_KEYIDX	M	Slot index of the key to be used (MK:DAT or DUKPT:DAT).
SPE_WKENC	MD	Working Key (encrypted by the MK) to be used for encryption. This field is mandatory only if **SPE_MTHDDAT** = **“0x”** or **“1x”**.
SPE_IVCBC	O	“IV” (Initialization Vector) to be used for encryption, if **SPE_MTHDDAT = “x1”** (CBC mode). If absent, the pinpad will consider the “IV” to be zero.

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= “**EBX**”).
RSP_STAT	M	Relevant return codes (see **section 3.1.1**): ↳ST_ERRKEY ..... MK/DUKPT not present in the pinpad. ↳ST_INVPARM..... Slot index (**SPE_KEYIDX)** is outside the range supported by the pinpad. ↳ST_INVPARM..... **SPE_DATAIN** length is not multiple of 8 or is bigger than 256.
PP_DATAOUT	M	Encrypted data (same size as **SPE_DATAIN**).
PP_KSN	MD	KSN (Key Serial Number) of the key used, in case of DUKPT method.

➲ Examples

SPE requests the encryption of a 24-byte block, containing the ASCII message “DATA TO BE ENCRYPTED”, using the DUKPT:TDES index “07”, with variant #5 and CBC mode.

SPE ⇒	45 42 58 30 34 30 00 0F 00 18 44 41 54 41 20 54 4F 20 42 45 20 45 4E 43 52 59 50 54 45 44 20 20 20 20 00 03 00 02 37 31 00 09 00 02 30 37	EBX040....DATA•T O•BE•ENCRYPTED•• ••....71....07

Pinpad returns encrypted data, accompanied by the KSN.

⇐ PP	45 42 58 30 30 30 30 34 32 80 4E 00 18 0F 77 0C 3A 6C AF CA 69 5D 00 50 14 41 82 7B A5 2C 21 81 48 C3 5C 94 D1 80 4C 00 0A FF FF F1 23 45 00 88 80 06 C3	EBX000042€N...w. :l¯Êi].P.A‚{¥,!. HÃ\”Ñ€L..ÿÿñ#E.ˆ €.Ã

3.3.7. "ENB" command

☑ **Obsolete**
☐ **Blocking**
☐ **Abecs**

This command encrypts any 8-byte data block using the MK/WK method, with a "data" MK.

⚠ This command is **obsolete**, the SPE shall use **"EBX"** instead.

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= "**ENB**").
CMD_LEN1	N3	Length of the following data (fixed "051").
ENB_METHOD	N1	Encryption method: **"1"** = MK/WK:TDES:DAT
ENB_MKIDX	N2	Slot index of the MK to be used.
ENB_WKENC	H32	Working Key encrypted by the MK.
ENB_INPUT	H16	8-byte data to be encrypted. In "Encrypted PAN" mode, this data is always encrypted using reverse TDES with the **WK_{PAN}** key (see **section 5.3**), regardless of its content.

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**ENB**").
RSP_STAT	N3	Relevant return data (see **section 3.1.1**): ↳ST_ERRKEY.....MK/DUKPT not present in the pinpad. ↳ST_INVPARM .....Slot index (**ENB_MKIDX**) is outside the range supported by the pinpad.
RSP_LEN1	N3	Length of the following data (fixed "016").
ENB_OUTPUT	H16	Encrypted data.

➲ Examples

SPE requests encryption of data 4C45455045415254 using MK:TDES:DAT index "14".

SPE ⇒	45 4E 42 30 35 31 31 31 34 46 45 34 42 31 33 36 34 34 36 33 32 39 46 45 36 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 34 43 34 35 34 35 35 30 34 35 34 31 35 32 35 34	ENB051114FE4B136 446329FE60000000 0000000004C45455 045415254

Operation is successful.		
⇐ **PP**	45 4E 42 30 30 30 30 31 36 46 43 31 43 37 41 41 43 38 35 32 45 35 44 39 46	ENB000016FC1C7AA C852E5D9F

3.3.8. “GCD” command

☐ **Obsolete**
☑ **Blocking**
☑ **Abecs**

This command allows the SPE to capture cleartext data using the pinpad keyboard. In order to comply with PCI security requirements, the prompt message shall be selected among those available in a fixed table defined by this specification.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**GCD**”).
SPE_MSGIDX	M	Index of the message to be presented, according to the fixed table below.
SPE_MINDIG	O	Minimum number of digits to be captured. If absent, an empty entry is allowed.
SPE_MAXDIG	O	Maximum number of digits to be captured. If absent, the value 32 is assumed. If present, it must be greater than or equal to SPE_MINDIG.
SPE_TIMEOUT	O	Maximum waiting time for a cardholder action, in seconds. If absent, this command never returns ↳ST_TIMEOUT.

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= “**GCD**”).
RSP_STAT	M	Relevant return codes (see **section 3.1.1**): ↳ST_CANCEL..... Cardholder pressed [CANCEL]. ↳ST_TIMEOUT ..... **SPE_TIMEOUT** time has expired.
PP_VALUE	M	Value entered by the cardholder.

➲ Message table

SPE_MSGIDX	Message	SPE_MSGIDX	Message
0001h	DIGITE O DDD	**001Ch**	ANO DO NASCIMENTO (AAAA)
0002h	REDIGITE O DDD	**001Dh**	DIGITE IDENTIFICAÇÃO
0003h	DIGITE O TELEFONE	**001Eh**	CÓDIGO DE FIDELIDADE
0004h	REDIGITE O TELEFONE	**001Fh**	NÚMERO DA MESA
0005h	DIGITE DDD+TELEFONE	**0020h**	QUANTIDADE DE PESSOAS
0006h	REDIGITE DDD+TELEFONE	**0021h**	DIGITE QUANTIDADE

SPE_MSGIDX	Message	SPE_MSGIDX	Message
0007h	DIGITE O CPF	**0022h**	NÚMERO DA BOMBA
0008h	REDIGITE O CPF	**0023h**	NÚMERO DA VAGA
0009h	DIGITE O RG	**0024h**	NÚMERO DO GUICHÊ/CAIXA
000Ah	REDIGITE O RG	**0025h**	CÓDIGO DO VENDEDOR
000Bh	DIGITE OS 4 ÚLTIMOS DÍGITOS	**0026h**	CÓDIGO DO GARÇOM
000Ch	DIGITE CÓDIGO DE SEGURANÇA	**0027h**	NOTA DO ATENDIMENTO
000Dh	DIGITE O CNPJ	**0028h**	NÚMERO DA NOTA FISCAL
000Eh	REDIGITE O CNPJ	**0029h**	NÚMERO DA COMANDA
000Fh	DIGITE A DATA (DDMMAAAA)	**002Ah**	PLACA DO VEÍCULO
0010h	DIGITE A DATA (DDMMAA)	**002Bh**	DIGITE QUILOMETRAGEM
0011h	DIGITE A DATA (DDMM)	**002Ch**	QUILOMETRAGEM INICIAL
0012h	DIGITE O DIA (DD)	**002Dh**	QUILOMETRAGEM FINAL
0013h	DIGITE O MÊS (MM)	**002Eh**	DIGITE PORCENTAGEM
0014h	DIGITE O ANO (AA)	**002Fh**	PESQUISA DE SATISFAÇÃO (0 a 10)
0015h	DIGITE O ANO (AAAA)	**0030h**	AVALIE ATENDIMENTO (0 a 10)
0016h	DATA DE NASCIMENTO (DDMMAAAA)	**0031h**	DIGITE O TOKEN
0017h	DATA DE NASCIMENTO (DDMMAA)	**0032h**	DIGITE NÚMERO DO CARTÃO
0018h	DATA DE NASCIMENTO (DDMM)	**0033h**	NÚMERO DE PARCELAS
0019h	DIA DO NASCIMENTO (DD)	**0034h**	CÓDIGO DO PLANO
001Ah	MÊS DO NASCIMENTO (MM)	**0035h**	CÓDIGO DO PRODUTO
001Bh	ANO DO NASCIMENTO (AA)		

➲ Examples

SPE requests the cardholder's RG (identification number), with a maximum of 10 digits, with a maximum idle time of 1 minute (60 sec).

SPE ⇒	47 43 44 30 31 36 00 0C 00 01 3C 00 0E 00 01 0A 00 0B 00 02 00 09	GCD016....<..... .....

Pinpad successfully returns a 9-digit entered data.

⇐ PP	47 43 44 30 30 30 30 31 33 80 4D 00 09 31 36 39 39 33 37 38 32 33	GCD000013€M..169 937823

3.3.9. "GDU" command

☑ **Obsolete**
☐ **Blocking**
☐ **Abecs**

This command retrieves the current KSN (Key Serial Number) from a DUKPT:TDES:PIN slot.

⚠ This command is **obsolete**, the SPE shall use **"GIX"** instead.

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= "**GDU**").
CMD_LEN1	N3	Length of the following data (fixed "003").
GDU_METHOD	N1	Encryption mode: **"3"** = DUKPT:TDES
GDU_IDX	N2	DUKPT:TDES:PIN slot index.

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**GDU**").
RSP_STAT	N3	Relevant return codes (see **section 3.1.1**): ↳ST_ERRKEY.....DUKPT not present in the pinpad. ↳ST_INVPARM .....Slot index (**GDU_IDX**) is outside the range supported by the pinpad.
RSP_LEN1	N3	Length of the following data (fixed "020").
GDU_KSN	H20	KSN (Key Serial Number).

➲ Examples

SPE requests the current KSN of DUKPT:TDES:PIN slot index "12"		
SPE ⇒	47 44 55 30 30 33 33 31 32	GDU003312
Operation is successful (KSN = FFFFF102910025800001).		
⇐ PP	47 44 55 30 30 30 30 32 30 46 46 46 46 46 31 30 32 39 31 30 30 32 35 38 30 30 30 30 31	GDU000020FFFFF10 2910025800001

3.3.10. “GKY” command

☑ **Obsolete**
☑ **Blocking**
☐ **Abecs**

This command is used to wait for a keystroke on the pinpad, returning the key code. For security reasons, this command does not return numeric keys, so these keys are simply ignored.

⚠ This command is **obsolete**, the SPE shall use **“CEX”** with **SPE_CEXOPT** = “100000” instead.

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= “**GKY**”).

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= “**GKY**”).
RSP_STAT	N3	Relevant return codes (see **section 3.1.1**): ↳ST_OK .....[OK/ENTER] key pressed ↳ST_CANCEL .....[CANCEL] key pressed. ↳ST_BACKSP .....[CLEAR] key pressed. ↳ST_F1 a ↳ST_F4 .....Function key pressed.

➲ Examples

SPE requests key pressing.		
SPE ⇒	47 4B 59	GKY
Operation is successful ([CANCEL] key pressed).		
⇐ PP	47 4B 59 30 31 33	GKY013

3.3.11. “GPN” command

☐ **Obsolete**
☑ **Blocking**
☐ **Abecs**

This command captures the cardholder’s PIN and returns a block of data encrypted according to the MK/WK:TDES or DUKPT:TDES.

The pinpad always clears the display in the end, whether the processing is successful or unsuccessful.

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= “**GPN**”).
CMD_LEN1	N3	Length of the following data.
GPN_METHOD	N1	Encryption method: **“1”** = MK/WK:TDES:PIN **“3”** = DUKPT:TDES:PIN
GPN_KEYIDX	N2	Slot index of the key to be used (MK:PIN or DUKPT:PIN).
GPN_WKENC	H32	Working Key encrypted by the MK. If **GPN_METHOD** = “3”, the pinpad ignores this field.
GPN_PANLEN	N2	PAN length (de “02” a “19”). If “End-to-End” encryption is being used (see **section 5.4**) and the “**GTK**” command has not yet been executed, an “empty” PAN (size “00”) may be provided for the pinpad to consider the PAN already stored in its memory.
GPN_PAN	A19	PAN, left aligned with trailing spaces. If the pinpad is in “Encrypted PAN” mode, this field shall be encrypted using **reverse** TDES with the $\mathbf{WK_{PAN}}$ (see **section 5.3**).
GPN_ENTRIES	N1	Number of entries to be captured (fixed **“1”**).
GPN_MIN1	N2	Minimum capture length (≥ “04”).
GPN_MAX1	N2	Maximum capture length (≥ **GPN_MIN1**).
GPN_MSG1	S32	Message to be prompted for PIN capture (2 rows by 16 columns)

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= “**GPN**”).

Field Id.	Format	Description
RSP_STAT	N3	Relevant return codes (see **section 3.1.1**): ↳ST_CANCEL .....Cardholder pressed [CANCEL]. ↳ST_TIMEOUT.....PIN capture timeout. ↳ST_ERRKEY.....MK/DUKPT not present in the pinpad. ↳ST_INVPARM .....Slot index (**GPN_KEYIDX**) is outside the range supported by the pinpad. ↳ST_INVPARM .....**GPN_MIN1** is lower than “04”. ↳ST_INVPARM .....**GPN_ENTRIES** is not “1”. ↳ST_INVCALL .....PAN is unknown by the pinpad.
RSP_LEN1	N3	Length of the following data (fixed “036”).
GPN_PINBLK	H16	Encrypted PIN
GPN_KSN	H20	KSN (Key Serial Number) of the key used, in case of DUKPT method. For MK / WK, this field is returned with zeros.

➲ Examples

SPE requests a PIN capture using the MK/WK:TDES method, with MK slot index “08”.

SPE ⇒	47 50 4E 30 39 33 31 30 38 34 31 33 35 45 41 35 38 42 41 31 33 45 32 36 32 46 34 34 43 35 39 45 44 37 38 39 39 41 41 33 43 31 36 34 34 34 34 33 33 33 33 32 32 32 32 31 31 31 31 20 20 20 31 30 34 31 32 52 24 20 20 20 20 20 20 20 20 20 33 34 2C 35 36 44 49 47 49 54 45 20 53 55 41 20 53 45 4E 48 41	GPN0931084135EA5 8BA13E262F44C59E D7899AA3C1644443 33322221111•••10 412R$•••••••••34 ,56DIGITE•SUA•SE NHA

Operation is successful.

⇐ PP	47 50 4E 30 30 30 30 33 36 42 42 36 42 45 32 38 46 44 46 33 35 32 32 45 39 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30	GPN000036BB6BE28 FDF3522E90000000 0000000000000

3.3.12. “GTK” command

☐ **Obsolete**
☐ **Blocking**
☑ **Abecs**

This command allows the SPE to obtain the complete tracks of the card read on **“CEX”** or **“GCX”** (in which case it can also return the PAN). Data may be returned in cleartext or encrypted as defined in **section 5.4**.

⚠ The “**GTK**” command can only be used once after “**CEX”** or **“GCX”**.

⚠ For encrypted tracks, on shall use the parameters defined in the specifications of the Acquirer Network that will process the transaction.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**GTK**”).
SPE_TRACKS	O	Identification of which track information shall be returned: “**1**xxx” = PAN is required, if available[1]; “**0**xxx” = PAN is not required. “x**1**xx” = Track 1 is required, if available; “x**0**xx” = Track 1 is not required. “xx**1**x” = Track 2 is required, if available; “xx**0**x” = Track 2 is not required. “xxx**1**” = Track 3 is required, if available; “xxx**0**” = Track 3 is not required. If this field is missing, all information known to the pinpad will be returned.
SPE_MTHDDAT	O	Identification of the encryption mode to be used: **“10”** = MK/WK:TDES:DAT (**ECB** block encryption); **“11”** = MK/WK:TDES:DAT (**CBC** block encryption); **“50”** = DUKPT:TDES:DAT#3 (**ECB** block encryption**,** see **section 5.1.1**); and **“51”** = DUKPT:TDES:DAT#3 (**CBC** block encryption**,** see **section 5.1.1**). “**90**” = TDES with random key (**ECB** block encryption). “**91**” = TDES with random key (**CBC** block encryption). If this field is missing, the tracks are returned in cleartext.
SPE_IVCBC	O	“IV” (Initialization Vector) to be used for encryption, if **SPE_MTHDDAT = “x1**” (CBC mode). If absent, the pinpad will consider the “IV” to be zero.

1 It is understood that the data is “available” when it is successfully read from the magnetic card, or, in the case of a chip card, when the equivalent TLV objects are present.

Field Id.	Presence	Description / Remark
SPE_OPNDIG	O	Number of **numeric** digits (**even** number) to be preserved in cleartext at the beginning of the tracks. If not provided, the entire track is encrypted.
SPE_KEYIDX	MD	Slot index of the key to be used (MK:DAT or DUKPT:DAT) in the encryption of tracks. This field is mandatory if **SPE_MTHDDAT** is present and different from "**9x**".
SPE_WKENC	MD	Working Key encrypted by the MK. This field is mandatory only if **SPE_MTHDDAT** = **"1x"**.
SPE_PBKMOD	MD	RSA key modulus. This field is mandatory only if **SPE_MTHDDAT** = **"9x"**.
SPE_PBKEXP	MD	RSA public exponent. This field is mandatory only if **SPE_MTHDDAT** = **"9x"**.

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= "**GTK**").
RSP_STAT	M	Relevant return codes (see **section 3.1.1**): ↳ST_INVCALL..... No successful "**CEX"** or **"GCX"** command has been executed previously. ↳ST_INVCALL..... **"GTK"** command already used. ↳ST_ERRKEY ..... MK/DUKPT not present in the pinpad. ↳ST_INVPARM..... Slot index (**SPE_KEYIDX**) is outside the range supported by the pinpad.
PP_ENCPAN	O	Card PAN, in cleartext or encrypted, if available and requested in **SPE_TRACKS** (only for ICC/CTLS, after **"GCX"**).
PP_TRACK1	O	Card Track 1, in cleartext or encrypted, if available and requested in **SPE_TRACKS**.
PP_TRACK2	O	Card Track 2, in cleartext or encrypted, if available and requested in **SPE_TRACKS**.
PP_TRACK3	O	Card Track 3, in cleartext or encrypted, if available and requested in **SPE_TRACKS**.
PP_TRK1KSN	MD	KSN used for Track 1 encryption. This field is mandatory if **PP_TRACK1** is present and method is DUKPT.
PP_TRK2KSN	MD	KSN used for Track 2 encryption. This field is mandatory if **PP_TRACK1** is present and method is DUKPT.
PP_TRK3KSN	MD	KSN used for Track 3 encryption. This field is mandatory if **PP_TRACK1** is present and method is DUKPT.
PP_ENCPANKSN	MD	KSN used for PAN encryption. This field is mandatory if **PP_TRACK1** is present and method is DUKPT.

Field Id.	Presence	Description / Remark
PP_ENCKRAND	MD	Random key (K_{RAND}) encrypted using the provided RSA public key, if **SPE_MTHDDAT** = "**9x**".

⚠ If a magnetic card was swiped in **"CEX"** or **"GCX"** but no track could be read (reading error), **"GTK"** returns ↳ST_OK without card data.

➲ Examples

SPE requests all the three tracks with 6 (six) cleartext digits at the beginning, using DUKPT:TDES encryption (variant # 2) in ECB mode, with the key in slot index "12".

SPE ⇒	47 54 4B 30 32 35 00 03 00 02 34 30 00 07 00 04 30 31 31 31 00 08 00 01 36 00 09 00 02 31 32	GTK025....40.... 0111....6....12

Pinpad returns Tracks 1 and 2 and the respective generated KSN but does not return Track 3, as it is unknown.

⇐ PP	47 54 4B 30 30 30 31 33 33 80 44 00 4E 35 34 37 38 32 33 7A E2 FA 69 BA 8C 62 93 9E C2 38 2C 33 D5 A1 6C 06 A2 D4 F6 EA 24 1E DC 93 73 21 92 FD D5 32 74 95 66 7C 8F D2 DF E6 A0 1C B7 94 BE C5 8C 57 65 D9 4C E1 8A CD CC CB 57 68 51 64 DD 65 56 C7 35 BE 35 7E 39 45 6A 68 DB 80 47 00 0A FF FF F8 19 46 00 18 70 00 1F 80 45 00 13 54 78 23 EA 2F B6 CD 92 89 F9 70 1C B0 88 3F D6 CC 6F 79 80 48 00 0A FF FF F8 19 46 00 18 70 00 1F	GTK000133€D.N547 823zâúiºŒb"žÂ8,3 Õ¡l.¢Ôöê$.Ü"s!'ý Õ2t•f|•Òßæ .·"¾Å ŒWeÙLáŠÍÌËWhQdÝe VÇ5¾5~9EjhÛ€G..ÿ ÿø.F..p..€E..Tx# ê/¶Í'‰ùp.°^?ÖÌoy €H..ÿÿø.F..p..

SPE requests PAN and Track 2 with 4 (four) cleartext digits at the beginning, using MK/WK:TDES encryption with the key in slot index "07", in CBC mode with a provided "IV" (Initialization Vector).

SPE ⇒	47 54 4B 30 35 37 00 07 00 04 31 30 31 30 00 03 00 02 31 31 00 1D 00 08 7F 7C 1A FA C0 A8 4F B7 00 08 00 01 34 00 09 00 02 30 37 00 0A 00 10 C2 BC A2 4F 3E F8 F2 EF 1C 0F 07 A9 7D 38 C3 38	GTK057....1010.. ..11....•|.úÀ¨O· ....4....07....Â ¼¢O>øòï...©}8Ã8

Pinpad successfully returns PAN and Track 2. both encrypted as requested.

⇐ PP	47 54 4B 30 30 30 30 34 34 80 4A 00 0A 41 23 FC 45 2F 36 15 44 A7 32 80 45 00 1A 41 23 BB 80 F6 58 D4 4F BC 29 4B 8A 63 99 01 26 95 48 B8 8A C9 52 01 E8 4F BF	GTK000044€J..A#ü E/6.D§2€E..A#»€ö XÔO¼)KŠc™.&•H¸ŠÉ R.èO¿

SPE requests PAN, Track 1 and Track 2 totally encrypted using a random TDES key in CBC mode, with no “IV” (Initialization Vector).

SPE ⇒

```
47 54 4B 32 38 31 00 07 00 04 31 31 31 30 00 03   GTK281....1110..
00 02 39 31 00 24 00 80 80 45 05 9A 9D C7 D2 77   ..91.$..€E.š•ÇÒw
09 06 DC FD 01 04 E3 1E 23 CE 30 85 71 61 5D 1D   ..Üý..ã.#Î0…qa].
BA 6E C2 29 91 13 76 26 3B 6B 64 A3 CE 89 21 A7   ºnÂ)‘.v&;kd£Î‰!§
9C 94 80 E5 32 1E 52 66 28 7D 43 48 60 B7 5A 92   œ”€å2.Rf(}CH`·Z’
FD B0 4B A8 8A 59 95 C2 4B FC 02 EC 2D CB 5C 8F   ý°K¨ŠY•ÂKü.ì-Ë\•
AA C0 62 D7 60 D3 5E 79 98 9D 8E D9 8A D0 E3 56   ªÀb×`Ó^y˜□ŽÙŠÐãV
53 F4 B4 84 68 39 55 17 C3 17 12 AD E5 62 3C F5   Sô´„h9U.Ã..-åb<õ
29 4C BC CF EA CE 1A DA 9B 89 E2 21 22 D7 5C 39   )L¼ÏêÎ.Ú›‰â!"×\9
31 BC 14 E6 C1 BD 39 1B BF BF D9 E8 E8 A4 E5 4D   1¼.æÁ½9.¿¿Ùèè¤åM
F8 7B 05 AC 4E 43 E1 3F AA 93 EB A6 7D 95 D4 D3   ø{.¬NCá?ª“ë¦}•ÔÓ
B6 C3 D2 47 D3 C2 55 A7 F8 65 B3 96 82 2E 19 85   ¶ÃÒGÓÂU§øe³-,.…
08 04 95 8E C9 1B 31 A2 3D 68 6F FE 4A 76 E6 4C   ..•ŽÉ.1¢=hoþJvæL
31 B8 EA 51 BC 03 41 B5 79 7D AB 18 F6 F9 97 03   1¸êQ¼.Aµy}«.öù-.
35 6A B1 8D 9B FD 62 33 CD BC 31 DC 2C 46 F1 76   5j±□›ýb3Í¼1Ü,Fñv
1A F5 AF 5C EF C8 2A 29 32 99 0A 4D 04 67 D9 15   .õ¯\ïÈ*)2™.M.gÙ.
79 CF E1 26 83 48 DA 19 FF 3F C7 EA 96 9E B3 47   yÏá&ƒHÚ.ÿ?Çê–ž³G
37 7A EA EA 64 21 AA 55 00 25 00 03 01 00 01      7zêêd!ªU.%.....
```

Pinpad PAN and Track 1 successfully encrypted but does not return Track 2, as it is unknown.

⇐ PP

```
47 54 4B 33 35 37 30 30 30 80 4A 00 08 F1 58 F8   GTK357000€J..ñXø
C2 2E 09 59 1E 80 44 00 51 42 FB A4 60 A1 A9 17   Â..Y.€D.QBû¤`¡©.
B1 72 5C E1 E7 32 35 33 D0 7C 9F 0B 9A 6B E5 AB   ±r\áç253Ð|Ÿ.škå«
AD 0D DB A1 D6 7F F0 F7 DE A3 7F 5A 4F 5A 17 DA   -.Û¡Ö•ð÷Þ£•ZOZ.Ú
95 17 E7 3F 77 70 D7 7B 64 38 C7 FA 04 0B C4 BD   •.ç?wp×{d8Çú..Ä½
71 8F 80 56 86 7B 6F F9 51 76 A0 63 7B 67 91 F4   q□€V†{oùQv c{g‘ô
04 8D C3 38 5C 45 58 8D 82 07 80 63 01 00 13 F7   .•Ã8\EX□,.€c...÷
3B C3 B1 9D 6A 2D 25 0D 96 80 6D 1A 98 5F DF D1   ;Ã±•j-%.-€m.˜_ßÑ
96 35 02 A2 5A B1 07 E1 28 87 CC D1 C0 5E 5E 9B   -5.¢Z±.á(‡ÌÑÀ^^›
EE C6 CA 3D 81 AA 34 36 57 66 9B D1 76 0C 9B 5B   îÆÊ=•ª46Wf›Ñv.›[
FD 48 CD 77 93 F5 15 4E 6B 15 49 F3 99 33 B1 22   ýHÍw“õ.Nk.Ió™3±"
1A 15 8E 7B F7 E8 C0 6B 7B FE 5F 47 38 13 E7 FE   ..Ž{÷èÀk{þ_G8.çþ
6A 93 47 84 36 10 5F 7E 85 40 00 15 3E BC 95 38   j“G„6._~…@..>¼•8
56 12 FF 90 5D D3 8B 3F 6D 86 1F EA B9 E4 1A 7F   V.ÿ•]Ó‹?m†.ê¹ä.•
EA 6D 61 0A 71 0A 4A E4 F2 2B C6 35 A7 18 0C 2D   êma.q.Jäò+Æ5§..-
6C A4 A6 FA A3 F8 FD 51 E8 CA 0C 9E D1 DA 70 E1   l¤¦ú£øýQèÊ.žÑÚpá
FC 1D BF C6 DB CB 29 BF 90 4F 07 40 BC C1 7D FB   ü.¿ÆÛË)¿•O.@¼Á}û
82 16 D5 81 46 F6 4B 46 23 8B 85 5D 86 C6 CF 8F   ,.Õ•FöKF#‹…]†ÆÏ•
4E 8B 0B 0E DF EE 90 3C 82 01 F7 8E C7 8C 88 31   N‹..ßî•<,.÷ŽÇŒˆ1
12 0E C4 D2 F6 CA E2 A2 39 ED FF A9 94 50 EE 4D   ..ÄÒöÊâ¢9íÿ©”PîM
5C 95 B8 8B A4 A9 7A C3 2D 3A FD 62 69 88 B1 BE   \•¸‹¤©zÃ-:ýbiˆ±¾
EE D3 A4 CB 16 E1 87 0D 88 74 F6 E0 F8 B7 B6 7C   îÓ¤Ë.á‡.ˆtöàø·¶|
D7 35 B0 F7 96 1E 5A 22 18 1D D2 A6 2D 77         ×5°÷–.Z"..Ò¦-w
```

For validation purposes, this example considers the following values for the random key (**K_{RAND}**) and the private exponent (**K_{PRV}**):

K_{RAND} = FF 47 55 39 9A E4 28 93 44 D4 BB C0 7D 96 8B 5F

K_{PRV} =

```
24 2B D2 9D BC 5A AA 16 19 3C 8F 3A E5 7B AC 54
46 82 91 9A 3F D3 D5 FF 59 20 7C AE 5E 13 DF E0
7E 27 15 B5 3F BB D9 FA BB 24 01 89 20 6D FE 8C
82 64 78 81 C3 8C 51 05 5C 76 C7 8F 1A 9C 92 A7
BC E7 AF 27 4C EE A9 06 76 7F 54 20 2A 54 D0 B2
77 80 0E D5 77 D8 DA 12 F1 0F F3 8B D7 1C 3B CB
BC 9F 18 0C 63 C0 25 32 79 58 03 72 9A 63 4E 9D
50 F9 3C 04 5E 1F DF 08 DD E6 8C FA 59 AD F3 99
62 5F 01 5E 0E 32 70 BB 2B 7F 27 D2 16 E8 AE 43
28 1C 2E 43 E4 A2 4E 77 34 05 86 94 C5 93 45 35
C2 4E FD 21 B2 CC 47 AE 93 82 7F C9 38 1B 6D 59
F3 50 B2 F3 53 43 71 AF A3 E4 0D 5C A3 1A C7 74
45 83 A3 86 1E 08 E4 42 36 34 B2 9D B2 C3 BA 14
D2 F3 7E 70 4F 1A AB E6 51 F2 5C 43 E0 DE 57 7F
B5 30 EF 17 AC B8 F1 5A A5 A9 0D 20 D8 35 DA 78
2C 5D 69 6A 44 DB F8 EB 21 3E B3 E3 46 3E 53 01
```


3.3.13. “MNU” command

☐ **Obsolete**
☑ **Blocking**
☑ **Abecs**

In this command, the pinpad shows on its display, making the best use of its hardware resources, a menu of up to 20 options for cardholder to select.

- Each option can have a maximum of 24 characters.
- The pinpad will display the menu options always respecting the order in which they were provided.
- If the option is initiated by a numeric character (“0” to “9”), the pinpad may allow selection via the keyboard (hot key), by pressing the key corresponding to the character. If the SPE chooses to use this feature, it is up to it to ensure the integrity of the options so that there is no repetition.
- The SPE must provide at least one option.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**MNU**”).
SPE_TIMEOUT	O	Maximum waiting time for a cardholder action, in seconds. If absent, this command never returns ↳ST_TIMEOUT.
SPE_DSPMSG	O	Menu title. If not present, the pinpad will display only the options, with no title.
SPE_MNUOPT	M	Text for the 1st option (index “01”).
SPE_MNUOPT	O	Text for the 2nd option (index “01”).
...	...	...
SPE_MNUOPT	O	Text for the last option (index “xx”, where “xx” is the total number of provided options).

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= “**MNU**”).
RSP_STAT	M	Relevant return codes (see **section 3.1.1**): ↳ST_CANCEL..... Cardholder pressed [CANCEL]. ↳ST_TIMEOUT ..... **SPE_TIMEOUT** time has expired.
PP_VALUE	M	2-digit index to the selected menu option, considering the order in which they were provided by the SPE (from “01”).

➲ Examples

The SPE asks the pinpad to present a menu with the title "*Selecione, por favor*:" and the options:
⇨ "*5.Chamado Técnico*"
⇨ "*1.Consultas*"
⇨ "*3.Ajuda*"
⇨ "*Voltar!!*"
The timeout value is 30 seconds.

SPE ⇒	4D 4E 55 30 38 39 00 0C 00 01 1E 00 20 00 11 35 2E 43 68 61 6D 61 64 6F 20 54 E9 63 6E 69 63 6F 00 20 00 0B 31 2E 43 6F 6E 73 75 6C 74 61 73 00 20 00 07 33 2E 41 6A 75 64 61 00 20 00 08 56 6F 6C 74 61 72 21 21 00 1B 00 15 53 65 6C 65 63 69 6F 6E 65 2C 20 70 6F 72 20 66 61 76 6F 72 3A	MNU089.....•..5 .Chamado•Técnico .•..1.Consultas. •..3.Ajuda.•..Vo ltar!!....Seleci one,•por•favor:

Pinpad successfully returns the value "02", indicating that the option "*1.Consultas*" has been selected.

⇐ PP	4D 4E 55 30 30 30 30 30 36 80 4D 00 02 30 32	MNU000006€M..02

3.3.14. “RMC” command

☐ **Obsolete**
☑ **Blocking**
☐ **Abecs**

This command waits for the ICC removal. It has two different behaviors, according to the presence or absence of a card in the coupler.

Card present: It displays the message defined by **RMC_MSG**, alternating it with a “REMOVE CARD” message, remaining in this state until the card is removed.

Card absent: It just displays the message defined by **RMC_MSG** and returns immediately.

In both cases, the message defined by **RMC_MSG** remains on the display after the execution.

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= “**RMC**”).
CMD_LEN1	N3	Length of the following data (fixed “032”).
RMC_MSG	S32	32-character message to be presented on the display, already formatted for 2 rows and 16 columns.

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= “**RMC**”).
RSP_STAT	N3	See **section 3.1.1**.

➲ Examples

SPE requests removal of the card, displaying the message “OPERATION FINISHED”.

SPE ⇒	52 4D 43 30 33 32 20 20 20 20 4F 50 45 52 41 54 49 4F 4E 20 20 20 20 20 20 20 46 49 4E 49 53 48 45 44 20 20 20 20	RMC032••••OPERAT ION•••••••FINISH ED••••

Operation is successful.

⇐ PP	52 4D 43 30 30 30	RMC000

3.4. Multimedia Commands

This specification provides a series of commands for use on multimedia enabled pinpads (color graphic display and/or audio). Support for these commands is optional and depends on the device's resources.

This specification considers the following file formats, which may or may not be supported by the pinpad, being this capability informed in the command "**GIX**" (**PP_MFSUP**).

- PNG image (Portable Network Graphics), according to ISO/IEC 15948;
- JPG image (or JPEG), according to ISO/IEC 10918; and
- GIF image or animation (Graphics Interchange Format - CompuServe).

⚠ When a command in this section is not supported by the pinpad, it simply returns the response error defined in **section 2.3.4** (with **RSP_STAT** = "010"), as it does for any other unknown command.

The following commands are covered in this section:

CMD_ID	Meaning	Obsolete	Blocking	Abecs
"MLI"	Media File Load - Initialization	☐	☐	☑
"MLR"	Media File Load - Record	☐	☐	☑
"MLE"	Media File Load - End	☐	☐	☑
"LMF"	List Media Files	☐	☐	☑
"DMF"	Delete Media Files	☐	☐	☑
"DSI"	Display/Run Media File	☐	☐	☑

3.4.1. “MLI” command

☐ **Obsolete**
☐ **Blocking**
☑ **Abecs**

This command starts the process of loading (or replacing) a media file on the pinpad. This file is stored in a “non-volatile” manner and is preserved even after the pinpad is turned off.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**MLI**”).
SPE_MFNAME	M	Name of the media file to be loaded.
SPE_MFINFO	M	Information about the media file to be loaded: X4 = Size (de 0 a 4294967295 bytes). B2 = CRC of the file. B1 =Type (01h = **PNG**, 02h = **JPG**, 03h = **GIF**, other values = RFU); and B3 = RFU (000000h).

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= “**MLI**”).
RSP_STAT	M	See **section 3.1.1**.

➲ Examples

SPE requests the loading of a 3,334-byte PNG file named “QRCODE01”.

SPE ⇒	4D 4C 49 30 32 36 00 1E 00 08 51 52 43 4F 44 45 30 31 00 1F 00 0A 00 00 0D 06 F2 11 01 00 00 00	MLI026....QRCODE 01........ò.....

Operation is successful.

⇐ PP	4D 4C 49 30 30 30	MLI000

3.4.2. “MLR” command

☐ **Obsolete**
☐ **Blocking**
☑ **Abecs**

Through one or more calls to this command, the SPE sends the data from the media file whose load was initiated by “**MLI**”.

The data can be divided into several blocks to respect the standard structure of the protocol packets, as described in **section 3.1.3.1**.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**MLR**”).
SPE_DATAIN	M	File data block.
SPE_DATAIN	O	File data block.
...	..	...
SPE_DATAIN	O	File data block.

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= “**MLR**”).
RSP_STAT	M	Relevant return codes (see **section 3.1.1**): ↳ST_INVCALL..... No “**MLI**” command has been previously called. ↳ST_INTERR..... Lack of memory for managing received data.

➲ Examples

SPE starts loading data from the PNG file of the "**MLI**" command example (**section 3.4.1**). Note that the command is divided into two blocks (CMD_BLK1 and CMD_BLK2), both with 436 bytes.

SPE ⇒

```
4D 4C 52 34 33 36 00 0F 01 B0 89 50 4E 47 0D 0A  MLR436...°‰PNG..
1A 0A 00 00 00 0D 49 48 44 52 00 00 00 7D 00 00  ......IHDR...}..
00 7D 08 02 00 00 00 00 E2 FB 72 00 00 0A 37 69  .}......âûr...7i
43 43 50 73 52 47 42 20 49 45 43 36 31 39 36 36  CCPsRGB•IEC61966
2D 32 2E 31 00 00 78 9C 9D 96 77 54 53 D9 16 87  •2.1..xœ••wTSÙ.‡
CF BD 37 BD 50 92 10 8A 94 D0 6B 68 52 02 48 0D  Ï½7½P'.Š"ÐkhR.H.
BD 48 91 2E 2A 31 09 10 4A C0 90 00 22 36 44 54  ½H'.*1..JÀ•."6DT
70 44 51 91 A6 08 32 28 E0 80 A3 43 91 B1 22 8A  pDQ'¦.2(à€£C'±"Š
85 01 51 B1 EB 04 19 44 D4 71 70 14 1B 96 49 64  ….Q±ë..DÔqp..•Id
AD 19 DF BC 79 EF CD 9B DF 1F F7 7E 6B 9F BD CF  -.ß¼yïÍ›ß.÷~kŸ½Ï
DD 67 EF 7D D6 BA 00 90 FC 83 05 C2 4C 58 09 80  Ýgï}Öº.üƒ.ÂLX.€
0C A1 58 14 E1 E7 C5 88 8D 8B 67 60 07 01 0C F0  .¡X.áçÅˆ‹g`...ð
00 03 6C 00 E0 70 B3 B3 42 16 F8 46 02 99 02 7C  ..l.àp³³B.øF.™.|
D8 8C 6C 99 13 F8 17 BD BA 0E 20 F9 FB 2A D3 3F  ØŒl™.ø.½º.•ùû*Ó?
8C C1 00 FF 9F 94 B9 59 22 31 00 50 98 8C E7 F2  ŒÁ.ÿŸ"¹Y"1.P˜Œçò
F8 D9 5C 19 17 C9 38 3D 57 9C 25 B7 4F C9 98 B6  øÙ\..É8=Wœ%·OÉ˜¶
34 4D CE 30 4A CE 22 59 82 32 56 93 73 F2 2C 5B  4MÎ0JÎ"Y‚2V"sò,[
7C F6 99 65 0F 39 F3 32 84 3C 19 CB 73 CE E2 65  |ö™e.9ó2„<.ËsÎâe
F0 E4 DC 27 E3 8D 39 12 BE 8C 91 60 19 17 E7 08  ðäÜ'ã•9.¾Œ'`..ç.
F8 B9 32 BE 26 63 83 74 49 86 40 C6 6F E4 B1 19  ø¹2¾&cƒtI†@Æoä±.
7C 4E 36 00 28 92 DC 2E E6 73 53 64 6C 2D 63 92  |N6.('Ü.æsSdl•c'
28 32 82 2D E3 79 00 E0 48 C9 5F F0 D2 2F 58 CC  (2‚•ãy.àHÉ_ðÒ/XÌ
CF 13 CB 0F C5 CE CC 5A 2E 12 24 A7 88 19 26 5C  Ï.Ë.ÅÎÌZ..$§ˆ.&\
53 86 8D 93 13 8B E1 CF CF 4D E7 8B C5 CC 30 0E  S†□".‹áÏÏMç‹ÅÌ0.
37 8D 23 E2 31 D8 99 19 59 1C E1 72 00 66 CF FC  7•#â1Ø™.Y.ár.fÏü
59 14 79 6D 19 B2 22 3B D8 38 39 38 30 6D 2D 6D  Y.ym.²";Ø8980m•m
BE 28 D4 7F 5D FC 9B 92 F7 76 96 5E 84 7F EE 19  ¾(Ô□]ü›'÷v•^„□î.
44 1F F8 C3 F6 57 7E 99 0D 00 34 33 36 00 0F 01  D.øÃöW~™..436...
B0 B0 A6 65 B5 D9 FA 87 6D 69 15 00 5D EB 01 50  °°¦eµÙú‡mi..]ë.P
BB FD 87 CD 60 2F 00 8A B2 BE 75 0E 7D 71 1E BA  »ý‡Í`/.Š²¾u.}q.º
7C 5E 52 C4 E2 2C 67 2B AB DC DC 5C 4B 01 9F 6B  |^RÄâ,g+«ÜÜ\K.Ÿk
29 2F E8 EF FA 9F 0E 7F 43 5F 7C CF 52 BE DD EF  )/èïúŸ.•C_|ÏR¾Ýï
E5 61 78 F3 93 38 92 74 31 43 5E 37 6E 66 7A A6  åaxó"8't1C^7nfz¦
44 C4 C8 CE E2 70 F9 0C E6 9F 87 F8 1F 07 FE 75  DÄÈÎâpù.æŸ‡ø..þu
1E 16 11 FC 24 BE 88 2F 94 45 44 CB A6 4C 20 4C  ...ü$¾ˆ/"EDË¦L•L
96 B5 5B C8 13 88 05 99 42 86 40 F8 9F 9A F8 0F  •µ[È.ˆ.™B†@øŸšø.
C3 FE A4 D9 B9 96 89 DA F8 11 D0 96 58 02 A5 21  Ãþ¤Ù¹•‰Úø.Ð•X.¥!
1A 40 7E 1E 00 28 2A 11 20 09 7B 64 2B D0 EF 7D  .@~..(*.•.{d+Ðï}
0B C6 47 03 F9 CD 8B D1 99 98 9D FB CF 82 FE 7D  .ÆG.ùÍ‹Ñ™˜□ûÏ‚þ}
57 B8 4C FE C8 16 24 7F 8E 63 47 44 32 B8 12 51  W¸LþÈ.$□ŽcGD2¸.Q
CE EC 9A FC 5A 02 34 20 00 45 40 03 EA 40 1B E8  Îìšüz.4•.E@.ê@.è
03 13 C0 04 B6 C0 11 B8 00 0F E0 03 02 41 28 88  ..À.¶À.¸..à..A(ˆ
04 71 60 31 E0 82 14 90 01 44 20 17 14 80 B5 A0  .q`1à‚.•.D•..€µ
18 94 82 AD 60 27 A8 06 75 A0 11 34 83 36 70 18  .",-`¨.u .4ƒ6p.
74 81 63 E0 34 38 07 2E 81 CB 60 04 DC 01 52 30  t•cà48..•Ë`.Ü.R0
0E 9E 80 29 F0 0A CC 40 10 84 85 C8 10 15 52 87  .ž€)ð.Ì@.„…È..R‡
74 20 43 C8 1C B2 85 58 90 1B E4 03 05 43 11 50  t•CÈ.²…X•.ä..C.P
1C 94 08 25 43 42 48 02 15 40 EB A0 52 A8 1C AA  .".%CBH..@ë R¨.ª
86 EA A1 66 E8 5B E8 28 74 1A BA 00 0D 43 B7 A0  †ê¡fè[è(t.º..C·
51 68 12 FA 15 7A 07 23 30 09 A6 C1 5A B0 11 6C  Qh.ú.z.#0.¦ÁZ°.l
05 B3 60 4F 38 08 8E 84 17 C1 C9 F0 32 38 1F 2E  .³`O8.Ž„.ÁÉð28..
82 B7 C0 95 70 03 7C 10 EE 84 4F C3 97 E0 11 58  ‚·À•p.|.î„OÃ•à.X
0A 3F 81 A7 11 80 10 11 3A A2 8B 30 11 16 C2 46  .?•§.€..:¢‹0..ÂF
42 91 78 24 09 11 21 AB 90 12 A4 02 69 40 DA 90  B'x$..!«•.¤.i@Ú•
1E A4 1F B9 8A 48 91 A7 C8 5B 14 06 45 45 31 50  .¤.¹ŠH'§È[..EE1P
4C                                               L
```

Operation is successful.

⇐ PP

```
4D 4C 52 30 30 30  MLR000
```

SPE continues to load the data, again dividing the command into two blocks (CMD_BLK1 and CMD_BLK2), both with 436 bytes.

SPE ⇒

```
4D 4C 52 34 33 36 00 0F 01 B0 94 0B CA 1F 15 85   MLR436...°”.Ê.…
E2 A2 96 A1 56 A1 36 A3 AA 51 07 50 9D A8 3E D4   â¢•¡V¡6£ªQ.P•¨>Ô
55 D4 28 6A 0A F5 11 4D 46 6B A2 CD D1 CE E8 00   UÔ(j.õ.MFk¢ÍÑÎè.
74 2C 3A 19 9D 8B 2E 46 57 A0 9B D0 1D E8 B3 E8   t,:.‹.FW ›Ð.è³è
11 F4 38 FA 15 06 83 A1 63 8C 31 8E 18 7F 4C 1C   .ô8ú..ƒ¡cŒ1Ž.•L.
26 15 B3 02 B3 19 B3 1B D3 8E 39 85 19 C6 8C 61   &.³.³.³.ÓŽ9….ÆŒa
A6 B1 58 AC 3A D6 1C EB 8A 0D C5 72 B0 62 6C 31   ¦±X¬:Ö.ëŠ.År°bl1
B6 0A 7B 10 7B 12 7B 05 3B 8E 7D 83 23 E2 74 70   ¶.{.{.{.;Ž}ƒ#âtp
B6 38 5F 5C 3C 4E 88 2B C4 55 E0 5A 70 27 70 57   ¶8_\<Nˆ+ÄUàZp'pW
70 13 B8 19 BC 12 DE 10 EF 8C 0F C5 F3 F0 CB F1   p.¸.¼.Þ.ïŒ.ÅóðËñ
65 F8 46 7C 0F 7E 08 3F 8E 9F 21 28 13 8C 09 AE   eøF|.~.?ŽŸ!(.Œ.®
84 48 42 2A 61 2D A1 92 D0 46 38 4B B8 4B 78 41   „HB*a•¡’ÐF8K¸KxA
24 12 F5 88 4E C4 70 A2 80 B8 86 58 49 3C 44 3C   $.õˆNÄp¢€¸†XI<D<
4F 1C 25 BE 25 51 48 66 24 36 29 81 24 21 6D 21   O.%¾%QHf$6)•$!m!
ED 27 9D 22 DD 22 BD 20 93 C9 46 64 0F 72 3C 59   í'•"Ý"½•“ÉFd.r<Y
4C DE 42 6E 26 9F 21 DF 27 BF 51 A0 2A 58 2A 04   LÞBn&Ÿ!ß'¿Q *X*.
28 F0 14 56 2B D4 28 74 2A 5C 51 78 A6 88 57 34   (ð.V+Ô(t*\Qx¦ˆW4
54 F4 54 5C AC 98 AF 58 A1 78 44 71 48 F1 A9 12   TôT\¬˜¯X¡xDqHñ©.
5E C9 48 89 AD C4 51 5A A5 54 A3 74 54 E9 86 D2   ^ÉH‰-ÄQZ¥T£tTé†Ò
B4 32 55 D9 46 39 54 39 43 79 B3 72 8B F2 05 E5   ´2UÙF9T9Cy³r‹ò.å
47 14 2C C5 88 E2 43 E1 51 8A 28 FB 28 67 28 63   G.,Åˆâ CáQŠ(û(g(c
54 84 AA 4F 65 53 B9 D4 75 D4 46 EA 59 EA 38 0D   T„ªOeS¹ÔuÔFêYê8.
43 33 A6 05 D0 52 69 A5 B4 6F 68 83 B4 29 15 8A   C3¦.ÐRi¥´ohƒ´).Š
8A 9D 4A B4 4A 9E 4A 8D CA 71 15 29 1D A1 1B D1   Š•J´JžJ•Êq.).¡.Ñ
03 E8 E9 F4 32 FA 61 FA 75 FA 3B 55 2D 55 4F 55   .èéô2úaúuú;U•UOU
BE EA 26 D5 36 D5 2B AA AF D5 E6 A8 79 A8 F1 D5   ¾ê&Õ6Õ+ª¯Õæ¨y¨ñÕ
4A D4 DA D5 46 D4 DE A9 33 D4 7D D4 D3 D4 B7 A9   JÔÚÕFÔÞ©3Ô}ÔÓÔ·©
77 A9 DF D3 40 69 98 69 84 6B 34 33 36 00 0F 01   w©ßÓ@i˜i„k436...
B0 E4 6A EC D1 38 AB F1 74 0E 6D 8E CB 1C EE 9C   °äjìÑ8«ñt.mŽË.îœ
92 39 87 E7 DC D6 84 35 CD 34 23 34 57 68 EE D3   ’9‡çÜÖ„5Í4#4WhîÓ
1C D0 9C D6 D2 D6 F2 D3 CA D2 AA D2 3A A3 F5 54   .ÐœÖÒÖòÓÊÒªÒ:£õT
9B AE ED A1 9D AA BD 43 FB 84 F6 A4 0E 55 C7 4D   ›®í¡•ª½Cû„ö¤.UÇM
47 A0 B3 43 E7 A4 CE 63 86 0A C3 93 91 CE A8 64   G ³Cç¤Îc†.Ã“‘Î¨d
F4 31 A6 74 35 75 FD 75 25 BA F5 BA 83 BA 33 7A   ô1¦t5uýu%ºõºƒº3z
C6 7A 51 7A 85 7A ED 7A F7 F4 09 FA 2C FD 24 FD   ÆzQz…zíz÷ô.ú,ý$ý
1D FA BD FA 53 06 3A 06 21 06 05 06 AD 06 B7 0D   .ú½úS.:.!...-.·.
F1 86 2C C3 14 C3 5D 86 FD 86 AF 8D 8C 8D 62 8C   ñ†,Ã.Ã]†ý†¯ŒbŒ
36 18 75 19 3D 32 56 33 0E 30 CE 37 6E 35 BE 6B   6.u.=2V3.0Î7n5¾k
42 36 71 37 59 66 D2 60 72 CD 14 63 CA 32 4D 33   B6q7YfÒ`rÍ.cÊ2M3
DD 6D 7A D9 0C 36 B3 37 4B 31 AB 31 1B 32 87 CD   ÝmzÙ.6³7K1«1.2‡Í
1D CC 05 E6 BB CD 87 2D D0 16 4E 16 42 8B 06 8B   .Ì.æ»Í‡•Ð.N.B‹.‹
1B 4C 12 D3 93 99 C3 6C 65 8E 5A D2 2D 83 2D 0B   .L.Ó“™ÃleŽZÒ-ƒ-.
2D BB 2C 9F 59 19 58 C5 5B 6D B3 EA B7 FA 68 6D   -»,ŸY.XÅ[m³ê·úhm
6F 9D 6E DD 68 7D C7 86 62 13 68 53 68 D3 63 F3   o•nÝh}Ç†b.hShÓcó
AB AD 99 2D D7 B6 C6 F6 DA 5C F2 5C DF B9 AB E7   «-™-×¶ÆöÚ\ò\ß¹«ç
76 CF 7D 6E 67 6E C7 B7 DB 63 77 D3 9E 6A 1F 62   vÏ}ngnÇ·ÛcwÓžj.b
BF C1 BE D7 FE 83 83 A3 83 C8 A1 CD 61 D2 D1 C0   ¿Á¾×þƒƒ£ƒÈ¡ÍaÒÑÀ
31 D1 B1 D6 F1 06 8B C6 0A 63 6D 66 9D 77 42 3B   1Ñ±Öñ.‹Æ.cmf•wB;
79 39 AD 76 3A E6 F4 D6 D9 C1 59 EC 7C D8 F9 17   y9-v:æôÖÙÁYì|Øù.
17 A6 4B 9A 4B 8B CB A3 79 C6 F3 F8 F3 1A E7 8D   .¦KšK‹Ë£yÆóøó.ç•
B9 EA B9 72 5C EB 5D A5 6E 0C B7 44 B7 BD 6E 52   ¹ê¹r\ë]¥n.·D·½nR
77 5D 77 8E 7B 83 FB 03 0F 7D 0F 9E 47 93 C7 84   w]wŽ{ƒû..}.žG“Ç„
A7 A9 67 AA E7 41 CF 67 5E D6 5E 22 AF 0E AF D7   §©gªçAÏg^Ö^"¯.¯×
6C 67 F6 4A F6 29 6F C4 DB CF BB C4 7B D0 87 E2   lgöJö)oÄÛÏ»Ä{Ð‡â
13 E5 53 ED 73 DF 57 CF 37 D9 B7 D5 77 CA CF DE   .åSísßWÏ7Ù·ÕwÊÏÞ
6F                                                o
```

Operation is successful.

⇐ PP

```
4D 4C 52 30 30 30   MLR000
```

SPE continues to load the data, again dividing the command into two blocks (CMD_BLK1 and CMD_BLK2), both with 436 bytes.

SPE ⇒

```
4D 4C 52 34 33 36 00 0F 01 B0 85 DF 29 7F B4 7F   MLR436...°…ß)•´•
90 FF 36 FF 1B 01 5A 01 DC 80 E6 80 A9 40 C7 C0   •ÿ6ÿ..Z.Ü€æ€©@ÇÀ
95 81 7D 41 A4 A0 05 41 D5 41 0F 82 CD 82 45 C1   ••}A¤ .AÕA.‚Í‚EÁ
3D 21 70 48 60 C8 F6 90 BB F3 0D E7 0B E7 77 85   =!pH`Èö•»ó.ç.çw…
82 D0 80 D0 ED A1 F7 C2 8C C3 96 85 7D 1F 8E 09   ‚Ð€Ðí¡÷ÂŒÃ•…}.Ž.
0F 0B AF 09 7F 18 61 13 51 10 D1 BF 80 BA 60 C9   .. ¯ .□.a.Q.Ñ¿€º`É
82 96 05 AF 22 BD 22 CB 22 EF 44 99 44 49 A2 7A   ‚•.¯"½"Ë"ïD™DI¢z
A3 15 A3 13 A2 9B A3 5F C7 78 C7 94 C7 48 63 AD   £.£.¢›£_ÇxÇ”ÇHc-
62 57 C6 5E 8A D3 88 13 C4 75 C7 63 E3 A3 E3 9B   bWÆ^ŠÓˆ.ÄuÇcã£ã›
E2 A7 17 FA 2C DC B9 70 3C C1 3E A1 38 E1 FA 22   â§.ú,Ü¹p<Á>¡8áú"
E3 45 79 8B 2E 2C D6 58 9C BE F8 F8 12 C5 25 9C   ãEy‹.,ÖXœ¾øø.Å%œ
25 47 12 D1 89 31 89 2D 89 EF 39 A1 9C 06 CE F4   %G.Ñ‰1‰•‰ï9¡œ.Îô
D2 80 A5 B5 4B A7 B8 6C EE 2E EE 13 9E 07 6F 07   Ò€¥µK§¸lî.î.ž.o.
6F 92 EF CA 2F E7 4F 24 B9 26 95 27 3D 4A 76 4D   o'ïÊ/çO$¹&•'=JvM
DE 9E 3C 99 E2 9E 52 91 F2 54 C0 16 54 0B 9E A7   Þž<™âžR'òTÀ.T.ž§
FA A7 D6 A5 BE 4E 0B 4D DB 9F F6 29 3D 26 BD 3D   ú§Ö¥¾N.MÛŸö)=&½=
03 97 91 98 71 54 48 11 A6 09 FB 32 B5 33 F3 32   .•'˜qTH.¦.û2µ3ó2
87 B3 CC B3 8A B3 A4 CB 9C 97 ED 5C 36 25 0A 12   ‡³Ì³Š³¤Ëœ•í\6%..
35 65 43 D9 8B B2 BB C5 34 D9 CF D4 80 C4 44 B2   5eCÙ‹²»Å4ÙÏÔ€ÄD²
5E 32 9A E3 96 53 93 F3 26 37 3A F7 48 9E 72 9E   ^2šã•S“ó&7:÷Hžrž
30 6F 60 B9 D9 F2 4D CB 27 F2 7D F3 BF 5E 81 5A   Oo`¹ÙòMË'ò}ó¿^•Z
C1 5D D1 5B A0 5B B0 B6 60 74 A5 E7 CA FA 55 D0   Á]Ñ[ [°¶`t¥çÊúUÐ
AA A5 AB 7A 57 EB AF 2E 5A 3D BE C6 6F CD 81 B5   ª¥«zWë¯.Z=¾ÆoÍ•µ
84 B5 69 6B 7F 28 B4 2E 2C 2F 7C B9 2E 66 5D 4F   „µik•(´.,/|¹.f]O
91 56 D1 9A A2 B1 F5 7E EB 5B 8B 15 8A 45 C5 37   'VÑš¢±õ~ë[‹.ŠEÅ7
36 B8 6C A8 DB 88 DA 28 D8 38 B8 69 EE A6 AA 4D   6¸l¨Û^Ú(Ø8¸iî¦ªM
1F 4B 78 25 17 4B AD 4B 2B 4A DF 6F E6 6E BE F8   .Kx%.K-K+Jßoæn¾ø
95 CD 57 95 5F 7D DA 92 B4 65 34 33 36 00 0F 01   •ÍW•_}Ú'´e436...
B0 B0 CC A1 6C CF 56 CC 56 E1 D6 EB DB DC B7 1D   °°Ì¡lÏVÌVáÖëÛÜ·.
28 57 2E CF 2F 1F DB 1E B2 BD 73 07 63 47 C9 8E   (W.Ï/.Û.²½s.cGÉŽ
97 3B 97 EC BC 50 61 57 51 B7 8B B0 4B B2 4B 5A   •;•ì¼PaWQ·‹°K²KZ
19 5C D9 5D 65 50 B5 B5 EA 7D 75 4A F5 48 8D 57   .\Ù]ePµµê}uJõH•W
4D 7B AD 66 ED A6 DA D7 BB 79 BB AF EC F1 D8 D3   M{-fí¦Ú×»y»¯ìñØÓ
56 A7 55 57 5A F7 6E AF 60 EF CD 7A BF FA CE 06   V§UWZ÷n¯`ïÍz¿úÎ.
A3 86 8A 7D 98 7D 39 FB 1E 36 46 37 F6 7F CD FA   £†Š}˜}9û.6F7ö•Íú
BA B9 49 A3 A9 B4 E9 C3 7E E1 7E E9 81 88 03 7D   º¹I£©´éÃ~á~é□ˆ.}
CD 8E CD CD 2D 9A 2D 65 AD 70 AB A4 75 F2 60 C2   ÍŽÍÍ-š-e-p«¤uò`Â
C1 CB DF 78 7F D3 DD C6 6C AB 6F A7 B7 97 1E 02   ÁËßx□ÓÝÆl«o§·•..
87 24 87 1E 7F 9B F8 ED F5 C3 41 87 7B 8F B0 8E   ‡$‡.□›øíõÃA‡{□°Ž
B4 7D 67 F8 5D 6D 07 B5 A3 A4 13 EA 5C DE 39 D5   ´}gø]m.µ£¤.ê\Þ9Õ
95 D2 25 ED 8E EB 1E 3E 1A 78 B4 B7 C7 A5 A7 E3   •Ò%íŽë.>.x´·Ç¥§ã
7B CB EF F7 1F D3 3D 56 73 5C E5 78 D9 09 C2 89   {Ëï÷.Ó=Vs\åxÙ.Â‰
A2 13 9F 4E E6 9F 9C 3E 95 75 EA E9 E9 E4 D3 63   ¢.ŸNæŸœ>•uêééäÓc
BD 4B 7A EF 9C 89 3D 73 AD 2F BC 6F F0 6C D0 D9   ½Kzïœ‰=s-/¼oðlÐÙ
F3 E7 7C CF 9D E9 F7 EC 3F 79 DE F5 FC B1 0B CE   óç|Ï•é÷ì?yÞõü±.Î
17 8E 5E 64 5D EC BA E4 70 A9 73 C0 7E A0 E3 07   .Ž^d]ìºäp©sÀ~ ã.
FB 1F 3A 06 1D 06 3B 87 1C 87 BA 2F 3B 5D EE 19   û.:...;‡.‡º/;]î.
9E 37 7C E2 8A FB 95 D3 57 BD AF 9E BB 16 70 ED   ž7|âŠû•ÓW½¯ž».pí
D2 C8 FC 91 E1 EB 51 D7 6F DE 48 B8 21 BD C9 BB   ÒÈü'áëQ×oÞH¸!½É»
F9 E8 56 FA AD E7 B7 73 6E CF DC 59 73 17 7D B7   ùèVú-ç·snÏÜYs.}·
E4 9E D2 BD 8A FB 9A F7 1B 7E 34 FD B1 5D EA 20   äžÒ½Šûš÷.~4ý±]ê
3D 3E EA 3D 3A F0 60 C1 83 3B 63 DC B1 27 3F 65   =>ê=:ð`Áƒ;cÜ±'?e
FF F4 7E BC E8 21 F9 61 C5 84 CE 44 F3 23 DB 47   ÿô~¼è!ùaÅ„ÎDó#ÛG
C7 26 7D 27 2F 3F 5E F8 78 FC 49 D6 93 99 A7 C5   Ç&}'/?^øxüIÖ“™§Å
3F 2B FF 5C FB CC E4 D9 77 BF 78 FC 32 30 15 3B   ?+ÿ\ûÌäÙw¿xü20.;
35                                                5
```

Operation is successful.

⇐ PP

```
4D 4C 52 30 30 30   MLR000
```

SPE finishes loading the data, this time dividing the command into two blocks (CMD_BLK1 and CMD_BLK2) of 436 and 314 bytes.

SPE ⇒

```
4D 4C 52 34 33 36 00 0F 01 B0 FE 5C F4 FC D3 AF  MLR436...°þ\ôüÓ¯
9B 5F A8 BF D8 FF D2 EE 65 EF 74 D8 F4 FD 57 19  ›_¨¿ØÿÒîeïtØôýW.
AF 66 5E 97 BC 51 7F 73 E0 2D EB 6D FF BB 98 77  ¯f^•¼Q□sà•ëmÿ»˜w
13 33 B9 EF B1 EF 2B 3F 98 7E E8 F9 18 F4 F1 EE  .3¹ï±ï+?˜~èù.ôñî
A7 8C 4F 9F 7E 03 F7 84 F3 FB 8F 70 66 2A 00 00  §ŒOŸ~.÷„óû•pf*..
00 09 70 48 59 73 00 00 0B 12 00 00 0B 12 01 D2  ..pHYs.........Ò
DD 7E FC 00 00 02 75 49 44 41 54 78 9C ED 9D 41  Ý~ü...uIDATxœí•A
8E 83 30 0C 00 B7 12 FF FF 72 F7 EE 43 90 6B 27  ŽƒO...·.ÿÿr÷îC•k'
63 D0 CC 35 10 CA C8 92 95 E0 B8 D7 DF 29 BE DF  cÐÌ5.ÊÈ'•à¸×ß)¾ß
6F D7 54 9F CF E7 E7 07 85 7B C3 C5 EB 99 1B B9  o×TŸÏçç.…{ÃÅë™.¹
CE 3C 46 02 7A 67 D0 3B 83 DE 19 A2 F7 63 D9 AF  Î<F.zgÐ;ƒÞ.¢÷cÙ¯
C2 3A 19 36 3E 77 9F 0D E3 9D 41 EF 0C 7A 67 D0  Â:.6>wŸ.ã•Aï.zgÐ
3B C3 8D F7 54 8E 4A 65 A1 D4 BA B1 71 B4 42 A3  ;Ã•÷TŽJe¡Ôº±q´B£
0D E3 9D 41 EF 0C 7A 67 D0 3B C3 50 EF A9 ED D9  .ã•Aï.zgÐ;ÃPï©íÙ
D4 68 63 9A AD 30 D4 FB EB D1 3B 83 DE 19 F4 CE  Ôhcš-0ÔûëÑ;ƒÞ.ôÎ
30 D4 7B 2A 19 EE BB 78 1F 43 BD BF 1E BD 33 E8  0Ô{*.î»x.C½¿.½3è
9D 41 EF 0C 37 DE 87 AC EE 02 95 DA A3 CA 1B 35  •Aï.7Þ‡¬î.•Ú£Ê.5
DA 30 DE 19 F4 CE A0 77 06 BD 33 44 EF C7 0A 62  Ú0Þ.ôÎ w.½3DïÇ.b
8F 51 F9 FA BA CF 86 F1 CE A0 77 06 BD 33 E8 9D  •Qùúºϟñî w.½3è•
E1 1A B2 22 A5 4A 91 02 C7 6C 18 EF 0C 7A 67 D0  á.²"¥J'.Çl.ï.zgÐ
3B 83 DE 19 AE 63 E7 39 C3 CC D4 27 D3 63 0B 54  ;ƒÞ.®cç9ÃÌÔ'Óc.T
EB 81 27 A2 77 06 BD 33 E8 9D E1 66 1F 78 DF 01  ë•'¢w.½3è•áf.xß.
97 F5 C5 C7 DA 35 EC 5B 27 AF EF 35 DE 19 F4 CE  ••õÅÇÚ5ì['¯ï5Þ.ô
A0 77 06 BD 33 74 9E 5F 6D BC 77 4D A5 2F 44 65  Î w.½3tž_m¼wM¥/D
BD DA 38 B3 F1 CE A0 77 06 BD 33 E8 9D 21 57 0F  e½Ú8³ñî w.½3è•!W
5C 69 8C D0 98 B2 F6 ED 5D 37 AE 57 D7 53 19 EF  .\iŒÐ˜²öí]7®W×S.
0C 7A 67 D0 3B 83 DE 19 62 DD 52 2A 29 55 52 65  ï.zgÐ;ƒÞ.bÝR*)URe
A0 92 B2 1A A7 AA B4 98 48 BD 33 31 34 00 0F 01  ’².§ª´˜H½314..
36 AF F1 CE A0 77 06 BD 33 E8 9D 21 57 B7 44 75  .6¯ñÎ w.½3è•!W·Du
0C 6C 3C A1 5A F9 AE DB 38 6A BC 33 E8 9D 41 EF  .l<¡Zù®Û8j¼3è•Aï
0C 7A 67 E8 F4 DE B8 40 4D 8D 56 FA EE 1F 3B CE  .zgèôÞ¸@M•VúÎ.;Î
EA 3E F0 08 F4 CE A0 77 06 BD 33 E4 BC 0F 29 09  ê>ð.ôÎ w.½3ä¼.)
DA D7 51 69 4D E5 67 98 57 47 A0 77 06 BD 33 E8  .Ú×QiMåg˜WG w.½3
9D 21 E7 7D C8 E1 CF F5 C5 33 0F E5 B8 0F 3C 02  è•!ç}ÈáÏõÅ3.å¸.<
BD 33 E8 9D 41 EF 0C 37 FD 96 1A CB 89 02 8D 55  .½3è•Aï.7ý•.Ë‰.•U
BB A9 D1 C6 23 A9 95 57 30 DE 19 F4 CE A0 77 06  »©ÑÆ#©•W0Þ.ôÎ w
BD 33 C4 BA A5 7D 54 FA 2D A5 66 AE D0 58 89 B5  .½3Äº¥}Tú•¥f®ÐX‰µ
7E 05 E3 9D 41 EF 0C 7A 67 D0 3B 43 F4 7E EC 5F  ~.ã•Aï.zgÐ;CÔ~ì_
C0 1B 8F 86 1E 9B B9 72 AF DF 57 47 A0 77 06 BD  À.□†.›¹r¯ßWG w.½
33 E8 9D A1 B3 DF 52 63 4E 1E D2 8F 22 85 E7 57  3è•¡³ßRcN.Ò•"…çW
1F 80 DE 19 F4 CE A0 77 86 29 DE 1B EB 81 D7 F7  .€Þ.ôÎ w†)Þ.ë•×÷
A6 9E 9B 9A D9 BC FA 00 F4 CE A0 77 06 BD 33 3C  ¦ž›šÙ¼ú.ôÎ w.½3<
D2 FB B1 3F DD B1 8F E1 DB D0 3B 83 DE 19 F4 CE  Òû±?Ý±•áÛÐ;ƒÞ.ôÎ
90 EB 0F DC C8 BE 92 A0 CA D7 D7 7D 7D F7 03 C6  •ë.ÜȾ' Ê××}}÷.Æ
3B 83 DE 19 F4 CE A0 77 86 9B FF 89 DB 47 63 8E  ;ƒÞ.ôÎ w†›ÿ‰ÛGcŽ
AA 9C 23 4D 4D B5 C6 7D E0 07 A0 77 06 BD 33 E8  ª#MMµÆ}à. w.½3è
9D E1 1F AC 1F 66 FE AE F3 F7 6D 00 00 00 00 49  •á.¬.fþ®ó÷m....I
45 4E 44 AE 42 60 82                              END®B`‚
```

Operation is successful.

⇐ PP

```
4D 4C 52 30 30 30  MLR000
```

3.4.3. “MLE” command

☐ **Obsolete**
☐ **Blocking**
☑ **Abecs**

This command finishes the media file loading process initiated by the “**MLI**” command. Upon receiving it, the pinpad checks the data received through the “**MLR**” commands, accepting or not the file.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**MLE**”).

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= “**MLE**”).
RSP_STAT	M	Relevant return codes (see **section 3.1.1**): ↳ST_INVCALL..... No “**MLI**” command has been previously called. ↳ST_MFERR..... Received file size or CRC does not match the information provided in the **“MLI”** command (**SPE_MFINFO**). ↳ST_INTERR..... Lack of memory for managing received data.

➲ Examples

SPE indicates the completion of the media file loading.		
SPE ⇒	4D 4C 45	MLE
Operation is successful.		
⇐ PP	4D 4C 45 30 30 30	MLE000

3.4.4. “LMF” command

☐ **Obsolete**
☐ **Blocking**
☑ **Abecs**

This command returns a list with the names of the media files loaded on the pinpad.

If no files are loaded, the command is successful and the returned list is empty.
There is no specific order for assembling the list, depending exclusively on pinpad implementation characteristics.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**LMF**”).

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= “**LMF**”).
RSP_STAT	M	See **section 3.1.1**.
PP_MFNAME	O	Name of a loaded file.
PP_MFNAME	O	Name of a loaded file.
…	..	...
PP_MFNAME	O	Name of a loaded file.

➲ Examples

SPE requests the list of media files loaded on the pinpad.

SPE ⇒	4C 4D 46	LMF

Operation is successful, returning the names of 5 media files.

⇐ PP	4C 4D 46 30 30 30 30 36 30 80 5E 00 08 53 49 47 4E 41 4C 53 20 80 5E 00 08 50 52 45 53 54 4F 20 20 80 5E 00 08 51 52 43 4F 44 45 30 31 80 5E 00 08 46 45 45 44 42 41 43 4B 80 5E 00 08 4D 4F 56 4E 50 49 43 54	LMF000060€^..SIGNALS•€^..PRESTO••€^..QRCODE01€^..FEEDBACK€^..MOVNPICT

3.4.5. “DMF” command

☐ **Obsolete**
☐ **Blocking**
☑ **Abecs**

This command deletes one or more media files stored on the pinpad.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**DMF**”).
SPE_MFNAME	M	Name of the media file to be excluded.
SPE_MFNAME	O	Name of the media file to be excluded.
...	...	...
SPE_MFNAME	O	Name of the media file to be excluded.

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= “**DMF**”).
RSP_STAT	M	See **section 3.1.1**. This command does not return an error if one or more files are already missing in the pinpad.

➲ Examples

SPE requests deletion of two media files on the pinpad.		
SPE ⇒	44 4D 46 30 32 34 00 1E 00 08 54 45 53 54 45 43 48 4F 00 1E 00 08 4D 4F 56 4E 50 49 43 54	DMF024....TESTEC HO....MOVNPICT
Operation is successful.		
⇐ PP	44 4D 46 30 30 30	DMF000

3.4.6. “DSI” command

☐ **Obsolete**
☐ **Blocking**
☑ **Abecs**

This command displays a media file previously loaded on the pinpad. The content will be centered on the display if its dimensions are smaller than the device's capability.

The pinpad display is erased before the presentation and previous messages or images are not kept.

This command always returns immediately (it is non-blocking), even if the media file contains animation (or video), which will be displayed until the pinpad receives a new command.

⚠ Pinpads are not required to support all media file formats provided for by this specification. The SPE must obtain the information of the supported formats through the command **“GIX”** (parameter **PP_MFSUP)**.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**DSI**”).
SPE_MFNAME	M	Name of the media file to be presented.

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= “**DSI**”).
RSP_STAT	M	Relevant return codes (see **section 3.1.1**): ↳ST_MFNFOUND..... Media file not found. ↳ST_MFERRFMT..... File format not supported by the pinpad, or its dimensions exceed display capability.

➲ Examples

SPE requests the presentation of the media file named “QRCODE01”.		
SPE ⇒	44 53 49 30 31 32 00 1E 00 08 51 52 43 4F 44 45 30 31	DSI012....QRCODE01
Operation is successful.		
⇐ PP	44 53 49 30 30 30	DSI000

3.5. EMV Table Management Commands

As detailed in **Chapter 4**, the pinpad must store several parameter tables that are used for EMV card processing (ICC or CTLS).

This section describes the commands used to manage and load these tables on the pinpad:

CMD_ID	Meaning	Obsolete	Blocking	Abecs
"GTS"	Get Table Version	☑	☐	☐
"TLI"	Table Load - Initialization	☐	☐	☐
"TLR"	Table Load - Record	☐	☐	☐
"TLE"	Table Load - End	☐	☐	☐

3.5.1. "GTS" command

☑ **Obsolete**
☐ **Blocking**
☐ **Abecs**

This command retrieves the version of the EMV Tables loaded on the pinpad. For more information, see **section 4.2**.

⚠ This command is **obsolete**, the SPE shall use "**GIX**" with **PP_TABVERnn** instead.

➲ mand

Field Id.	Format	Description
CMD_ID	A3	Command code (= "**GTS**").
CMD_LEN1	N3	Length of the following data (fixed "002").
GTS_ACQIDX	N2	Acquirer identifier of the EMV Tables whose version is being requested. The value "**00**" shall be used when having a single version for all Acquirer Networks (this only makes sense if the tables were loaded using "**00**" also in the "**TLI**" command).

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**GTS**").
RSP_STAT	N3	See **section 3.1.1**.
RSP_LEN1	N3	Length of the following data (fixed "010")
GTS_TABVER	A10	Current version of the EMV Tables for the provided Acquirer index (or for the total set of tables if **GTS_ACQIDX** = "00"). If there is no table loaded for the provided Acquirer index, this field returns zeros ("0000000000"). If the tables have been loaded separately for different Acquirers (with different versions) and **GTS_ACQIDX** = "00", this field also returns zeros ("0000000000"), since there is no "general" version representing the tables.

➲ Examples

SPE requests the version of the EMV Tables of the Acquirer Network index "02".

SPE ⇒	47 54 53 30 30 32 30 32	GTS00202

Pinpad returns version “XEMVST0003”.		
⇐ **PP**	47 54 53 30 30 30 30 31 30 58 45 4D 56 53 54 30 30 30 33	GTS000010XEMVST0003

3.5.2. "TLI" command

☐ **Obsolete**
☐ **Blocking**
☐ **Abecs**

This command starts the process of loading (or updating) tables. If it returns ↳ST_OK or ↳ST_TABVERDIF, the process can continue through the commands **"TLR"** and **"TLE"**.

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= "**TLI**").
CMD_LEN1	N3	Length of the following data (fixed "012").
TLI_ACQIDX	N2	Identifier of the Acquirer Network whose EMV Tables will be updated. To cover all Acquirers, the value "**00**" must be used.
TLI_TABVER	A10	New version of the EMV Tables to be loaded (free format managed by the SPE).

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**TLI**").
RSP_STAT	N3	Relevant return codes (see **section 3.1.1**): ↳ST_OK ..... Loading process started, but **TLI_TABVER** coincides with the current version of the EMV Tables already loaded. ↳ST_TABVERDIF ..... Loading process started, but **TLI_TABVER** differs from the current version of the EMV Tables already loaded.

➲ Examples

SPE requests the complete load of tables (all Acquirer Networks), informing the new version of EMV Tables ("TABVER0008").

SPE ⇒	54 4C 49 30 31 32 30 30 54 41 42 56 45 52 30 30 30 38	TLI01200TABVER0008

Pinpad starts the process successfully, stating that the version provided differs from the current version.

⇐ PP	54 4C 49 30 32 30	TLI020

3.5.3. “TLR” command

☐ **Obsolete**
☐ **Blocking**
☐ **Abecs**

This command loads on the pinpad one or more EMV Table records. The pinpad stores these records temporarily to preserve the current tables in the event of an error in the update operation, which is terminated by the **“TLE”** command.

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= “**TLR**”).
CMD_LEN1	N3	Length of the following data.
TLR_NREC	N2	Number of records in the following field.
----	???	One or more concatenated records, each starting with the size information, according to the format described in **section 4.1**. When concatenating the records, one must pay attention to the maximum size allowed by **CMD_LEN1** (“999”).

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= “**TLR**”).
RSP_STAT	N3	Relevant return codes (see **section 3.1.1**): ↳ST_INVCALL .....No “**TLI**” command has been previously called. ↳ST_TABERR .....Error trying to store records (out of memory, for example).

➲ Examples

SPE sends AID Table records “01” and “02” of Acquirer Network “03”.

SPE ⇒	54 4C 52 36 33 30 30 32 33 31 34 31 30 33 30 31 30 37 41 30 30 30 30 30 30 30 30 34 31 30 31 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 31 43 54 4C 45 53 53 2D 2D 43 52 45 44 49 54 4F 20 30 33 30 30 30 31 30 30 30 32 30 30 30 31 30 37 36 38 34 30 32 30 32 30 35 30 33 30 30 30 30 30 30 30 30 34 30 30 30 30 30 30 30 30 30 30 30 30 45 30 46 38 45 38 37 30 30 30 46 30 46 30 30 31 32 32 32 30 35 30 30 34 41 30 30 30 44 38 30 30 45 38 30 30 30 30 32 30 35 30 30 34 46 38 30 30 30 30 30 30 30 30 30 30 52 30 34 30 30 30 30 31 33 38 37 30 30 30 30 30 35 44 42 30 30 30 30 30 39 43 33 31 32 33 34 30 39 46 30 32 30 36 35 46 32 41 30 32 39 41 30 33 39 43 30 31 39 35 30 35 39 46 33 37 30 34 30 30 30 30 30 30 30 30 30 30 39 46 33 37 30 34 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 59 31 5A 31 59 33 5A 33 46 30 30 30 30 34 38 30 30 30 30 30 30 30 30 30 30 30 30 30 46 30 30 30 30 34 38 30 30 30 33 31 34 31 30 33 30 32 30 37 41 30 30 30 30 30 30 30 30 34 33 30 36 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 32 43 54 4C 45 53 53 2D 2D 44 45 42 49 54 4F 20 20 30 33 30 30 30 31 30 30 30 32 30 30 30 31 30 37 36 38 34 30 32 30 32 30 35 30 33 30 30 30 30 30 30 30 30 34 30 30 30 30 30 30 30 30 30 30 30 30 45 30 46 38 45 38 37 30 30 30 46 30 46 30 30 31 32 32 32 30 35 30 30 34 41 30 30 30 44 38 30 30 45 38 30 30 30 30 32 30 35 30 30 34 46 38 30 30 30 30 30 30 30 30 30 30 52 30 34 30 30 30 30 31 33 38 37 30 30 30 30 30 35 44 42 30 30 30 30 30 39 43 33 31 32 33 34 30 39 46 30 32 30 36 35 46 32 41 30 32 39 41 30 33 39 43 30 31 39 35 30 35 39 46 33 37 30 34 30 30 30 30 30 30 30 30 30 30 39 46 33 37 30 34 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 59 31 5A 31 59 33 5A 33 46 30 30 30 30 34 38 30 30 30 30 30 30 30 30 30 30 30 30 30 46 30 30 30 30 34 38 30 30 30	TLR6300231410301 07A0000000041010 0000000000000000 0001CTLESS--CRED ITO•030001000200 0107684020205030 0000000400000000 0000E0F8E87000F0 F00122205004A000 D800E80000205004 F80000000000R040 0001387000005DB0 00009C3123409F02 065F2A029A039C01 95059F3704000000 00009F3704000000 0000000000000000 000000000000Y1Z1 Y3Z3F00004800000 00000000F0000480 003141030207A000 0000043060000000 00000000000002CT LESS--DEBITO••03 0001000200010768 4020205030000000 04000000000000E0 F8E87000F0F00122 205004A000D800E8 0000205004F80000 000000R040000138 7000005DB000009C 3123409F02065F2A 029A039C0195059F 370400000000009F 3704000000000000 0000000000000000 000000Y1Z1Y3Z3F0 0004800000000000 00F000048000
Pinpad successfully accepts the records.		
⇐ PP	54 4C 52 30 30 30	TLR000

SPE sends CAPK Table record “13” of Acquirer Network “02” followed by Certification Revocation Table records “01”, “02” and “03” of Acquirer Network “01”.

SPE ⇒	54 4C 52 36 39 31 30 34 36 31 31 32 30 32 31 33 41 30 30 30 30 30 30 30 30 34 45 46 30 30 31 30 33 30 30 30 30 32 34 38 41 31 39 31 43 42 38 37 34 37 33 46 32 39 33 34 39 42 35 44 36 30 41 38 38 42 33 45 41 45 45 30 39 37 33 41 41 36 46 31 41 30 38 32 46 33 35 38 44 38 34 39 46 44 44 46 46 39 43 30 39 31 46 38 39 39 45 44 41 39 37 39 32 43 41 46 30 39 45 46 32 38 46 35 44 32 32 34 30 34 42 38 38 41 32 32 39 33 45 45 42 42 43 31 39 34 39 43 34 33 42 45 41 34 44 36 30 43 46 44 38 37 39 41 31 35 33 39 35 34 34 45 30 39 45 30 46 30 39 46 36 30 46 30 36 35 42 32 42 46 32 41 31 33 45 43 43 37 30 35 46 33 44 34 36 38 42 39 44 33 33 41 45 37 37 41 44 39 44 33 46 31 39 43 41 34 30 46 32 33 44 43 46 35 45 42 37 43 30 34 44 43 38 46 36 39 45 42 41 35 36 35 42 31 45 42 43 42 34 36 38 36 43 44 32 37 34 37 38 35 35 33 30 46 46 36 46 36 45 39 45 45 34 33 41 41 34 33 46 44 42 30 32 43 45 30 30 44 41 45 43 31 35 43 37 42 38 46 44 36 41 39 42 33 39 34 42 41 42 41 34 31 39 44 33 46 36 44 43 38 35 45 31 36 35 36 39 42 45 38 45 37 36 39 38 39 36 38 38 45 46 45 41 32 44 46 32 32 46 46 37 44 33 35 43 30 34 33 33 33 38 44 45 41 41 39 38 32 41 30 32 42 38 36 36 44 45 35 33 32 38 35 31 39 45 42 42 43 44 36 46 30 33 43 44 44 36 38 36 36 37 33 38 34 37 46 38 34 44 42 36 35 31 41 42 38 36 43 32 38 43 46 31 34 36 32 35 36 32 43 35 37 37 42 38 35 33 35 36 34 41 32 39 30 43 38 35 35 36 44 38 31 38 35 33 31 32 36 38 44 32 35 43 43 39 38 41 34 43 43 36 41 30 42 44 46 46 46 44 41 32 44 43 43 41 33 41 39 34 43 39 39 38 35 35 39 45 33 30 37 46 44 44 46 39 31 35 30 30 36 44 39 41 39 38 37 42 30 37 44 44 41 45 42 33 42 31 32 31 37 36 36 45 42 42 30 45 45 31 32 32 41 46 42 36 35 44 37 38 34 35 42 37 33 44 42 34 36 42 41 42 36 35 34 32 37 41 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 32 36 33 30 31 30 31 41 30 30 30 30 30 30 30 30 33 30 31 34 34 34 34 34 34 30 32 36 33 30 31 30 32 41 30 30 30 30 30 30 30 30 33 39 37 35 35 35 35 35 35 30 32 36 33 30 31 30 33 41 30 30 30 30 30 30 30 30 33 39 34 36 36 36 36 36 36	TLR6910461120313 A000000004EF0010 30000248A191CB87 473F29349B5D60A8 8B3EAEE0973AA6F1 A082F358D849FDDF F9C091F899EDA979 2CAF09EF28F5D224 04B88A2293EEBBC1 949C43BEA4D60CFD 879A1539544E09E0 F09F60F065B2BF2A 13ECC705F3D468B9 D33AE77AD9D3F19C A40F23DCF5EB7C04 DC8F69EBA565B1EB CB4686CD27478553 0FF6F6E9EE43AA43 FDB02CE00DAEC15C 7B8FD6A9B394BABA 419D3F6DC85E1656 9BE8E76989688EFE A2DF22FF7D35C043 338DEAA982A02B86 6DE5328519EBBCD6 F03CDD686673847F 84DB651AB86C28CF 1462562C577B8535 64A290C8556D8185 31268D25CC98A4CC 6A0BDFFFDA2DCCA3 A94C998559E307FD DF915006D9A987B0 7DDAEB3B121766EB B0EE122AFB65D784 5B73DB46BAB65427 A000000000000000 0000000000000000 0000000000002630 301A000000003014 4444402630302A00 0000003975555550 2630303A00000000 394666666

Pinpad successfully accepts the records.

⇐ PP	54 4C 52 30 30 30	TLR000

3.5.4. "TLE" command

☐ **Obsolete**
☐ **Blocking**
☐ **Abecs**

This command ends the process of loading (or updating) tables, making the records provided through "**TLR**" to be permanently stored, replacing the previous EMV Tables (if any). At this moment, **TLI_TABVER** is effective for the new tables.

If no **"TLR"** command is called between **"TLI"** and **"TLE"**, all EMV Tables of the referred Acquirer Network are simply deleted.

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= "**TLE**").

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**TLE**").
RSP_STAT	N3	Relevant return codes (see **section 3.1.1**): ↳ST_INVCALL .....No "**TLI**" command has been previously called. ↳ST_TABERR .....Error trying to store records (out of memory, for example).

➲ Examples

SPE requests the completion of the table loading process.		
SPE ⇒	54 4C 45	TLE
Pinpad accepts the command successfully, updating the tables.		
⇐ PP	54 4C 45 30 30 30	TLE000

3.6. Card Processing Commands (obsolete)

This section details high-level commands responsible for the complete processing of a card during a payment transaction, whether magnetic, ICC or CTLS.

⚠ All commands described in this section are **obsolete**. For these functionalities, the SPE must use commands described in **section 3.7**.

The following commands are covered in this section:

CMD_ID	Meaning	Obsolete	Blocking	Abecs
"GCR"	Get Card	☑	☑	☐
"CNG"	Change EMV Parameter	☑	☐	☐
"GOC"	Go On Chip Processing	☑	☑	☐
"FNC"	Finish Chip Processing	☑	☐	☐

3.6.1. "GCR" command

☑ **Obsolete**
☑ **Blocking**
☐ **Abecs**

This command initiates a payment card transaction process (be it magnetic, ICC or CTLS), as presented in **section 3.6.5**.

When activated, the pinpad shows a message on the display requesting the presentation of a card. If a chip card (ICC or CTLS) is used, EMV processing starts automatically. For this, the pinpad requires the EMV Tables to be loaded in its memory (see **Chapter 4**).

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= "**GCR**").
CMD_LEN1	N3	Length of the following data.
GCR_ACQIDXREQ	N2	Acquirer Network identifier (**TAB_ACQ**) whose EMV Tables will be used if an ICC or CTLS is presented. To cover the tables of all Acquirer Networks, the value **GCR_ACQIDXREQ** = "00" must be used (see **Note #1**).
GCR_APPTYPREQ	N2	Type of application required, in order to consider only AID Table records where **T1_APPTYPE** = **GCR_APPTYPREQ** (**see section 4.1.1**). ▪ To ignore **T1_APPTYPE**, use **GCR_APPTYPREQ** = "99". ▪ To use a specific list of records from the AID Tables, use **GCR_APPTYPREQ** = "00" (the list goes at the end of the command).
GCR_AMOUNT	N12	Initial transaction amount in cents (*Amount, authorized*), which should be zero (0) if this data is not available at the beginning of the transaction.
GCR_DATE	N6	Transaction date ("YYMMDD")
GCR_TIME	N6	Transaction time ("HHMMSS")
GCR_TABVER	A10	Expected version of the EMV Tables for the Acquirer Network defined in **GCR_ACQIDXREQ** (or the "general" table version if **GCR_ACQIDXREQ** = "**00**").
GCR_QTDAPP	N2	Number of entries in the list below (only if **GCR_APPTYPREQ** = "**00**"). IMPORTANT: This field is not optional and shall receive the value "00" if the following list does not exist.
GCR_IDAPP1	A4	Direct reference to a record of the AID Tables, composed of the concatenation of **TAB_ACQ** and **TAB_RECIDX**.
…	…	
GCR_IDAPPn	A4	Direct reference to a record of the AID Tables, composed of the concatenation of **TAB_ACQ** and **TAB_RECIDX**.

Field Id.	Format	Description
GCR_CTLSON (optional!)	N1	Enable contactless card interface (see **Note #2**): "**1**" = Yes (default); or "**0**" = No. **IMPORTANT**: To maintain compatibility with systems prior to this specification, this field is optional. A pinpad that supports CTLS will consider the value "**1**" (yes) if this parameter is missing at the end of the command.

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**GCR**").
RSP_STAT	N3	Relevant return codes (see **section 3.1.1**): ↳ST_MCDATAERR .....A magnetic card was swiped, but there was a reading error (no tracks could be read). ↳ST_TABVERDIF.....**GCR_TABVER** differs from the current version of the EMV Tables already loaded. See the procedure to be followed in "**Command (after ↳ST_TABVERDIF)**". ↳ST_CARDINVALIDAT ...ICC application is invalidated. ↳ST_CARDBLOCKED.....ICC is blocked. ↳ST_CARDPROBLEMS...Invalid or faulty ICC. ↳ST_CARDINVDATA .....ICC with invalid of missing data. ↳ST_CARDAPPNAV .....Invalid mode for the ICC. ↳ST_CARDAPPNAUT.....ICC not accepted. ↳ST_ERRFALLBACK.....ICC error that allows fallback to magnetic card. ↳ST_CTLSINVALIDAT.....CTLS is invalidated/blocked. ↳ST_CTLSPROBLEMS.....Invalid or faulty CTLS. ↳ST_CTLSAPPNAV.....Invalid mode for the CTLS. ↳ST_CTLSAPPNAUT.....CTLS not accepted. ↳ST_CTLSEXTCVM.....Request verification on the cardholder's device. ↳ST_CTLSIFCHG.....Change interface (use ICC or magnetic card).
RSP_LEN1	N3	Length of the following data.
GCR_CARDTYPE	N2	Processed card type: **"00"** = Magnetic; **"03"** = ICC EMV; **"05"** = CTLS magstripe mode; or **"06"** = CTLS EMV.

Field Id.	Format	Description
GCR_STATCHIP	N1	Status of the last ICC processing. The SPE uses this information to refuse (or not) a magnetic card (**GCR_CARDTYPE**="00") if its tracks indicate chip presence. "**0**" = Successful (or another status that does not imply fallback); or "**1**" = Error allowing to fallback; or "**2**" = Required application not supported (fallback depends on the Acquirer Network settings).
GCR_APPTYPE	N2	Returns the value of **T1_APPTYPE** from the AID Table record used in the chip card processing.
GCR_ACQIDX	N2	Returns the value of **TAB_ACQ** from the AID Table record used in the chip card processing.
GCR_RECIDX	A2	Returns the value of **TAB_RECIDX** from the AID Table record used in the chip card processing.
GCR_TRK1LEN	N2	Length of Track 1.
GCR_TRK1	A76	Track 1 (without the sentinels and with the format byte - first alphanumeric character), left aligned with trailing spaces.
GCR_TRK2LEN	N2	Length of Track 2.
GCR_TRK2	A37	Track 2 (without the sentinels), left aligned with trailing spaces.
GCR_TRK3LEN	N3 (or A3**)	Length of Track 3.
GCR_TRK3	A104	Track 3 (without the sentinels), left aligned with trailing spaces.
GCR_PANLEN	N2	Length of PAN.
GCR_PAN	A19	PAN, left aligned with trailing spaces.
GCR_PANSEQNO	N2	*Application PAN Sequence Number*
GCR_APPLABEL	A16	Label of the application being processed, with trailing spaces.
GCR_SRVCODE	N3	*Service Code*
GCR_CHNAME	A26	*Cardholder Name*, with trailing spaces.
GCR_CARDEXP	N6	Card expiration date *(Application Expiration Date)*, in "YYMMDD" format.
GCR_RFU1	N29	RFU (shall be ignored by the SPE).
GCR_ISSCNTRY	N3	*Issuer Country Code*.
GCR_ACQRDLEN	N3	Length **GCR_ACQRD**, in characters. ▪ If **GCR_ACQIDX** = **"01"**, **GCR_ACQRDLEN** is "066"; ▪ If **GCR_ACQIDX** = **"02"**, **GCR_ACQRDLEN** is "010"; and ▪ For other **GCR_ACQIDX values**, **GCR_ACQRD** field does not exist (**GCR_ACQRDLEN** is "000").
GCR_ACQRD	A..66	Return data specific to the selected Acquirer Network (see tables below).

⚠ If the pinpad is in "Encrypted PAN" mode (see section **5.3**), **GCR_PAN** and the PANs in the tracks are encrypted by the **WK_{PAN}** key.

⚠ If the pinpad is in "Encrypted PAN" mode, **GCR_TRK3LEN** is not filled, as Track 2 can reach up to 40 characters (see explanation in **section 5.3**).
** In this case its format change from "N3" to "A3"!!

For GCR_ACQIDX = "01":

Field Id.	Format	Description
GCR_ACQRD	N2	Number bytes represented in *Application Identifier* (length ÷ 2).
	H32	*Application Identifier* (tag 84h), with trailing FFh.
	A16	*Application Label* (tag 50h), with trailing spaces.
	A16	*Application Preferred Name* (tag 9F12h), with trailing spaces. If *Issuer Code Table Index* is not compatible to *Additional Terminal Capabilities*, this field is filled with spaces.

For GCR_ACQIDX = "02":

Field Id.	Format	Description
GCR_ACQRD	H10	*Application Usage Control* (tag 9F07h), in the format: "**9F0702xxxx**"

➲ Command (after ↳ST_TABVERDIF)

If the response to "**GCR**" informs ↳ST_TABVERDIF, the command was not processed because the EMV Tables have a version different from **GCR_TABVER**.

In this case, the SPE may or may not proceed with updating the tables (using the commands described in **section 3.4.4**) and then resubmit "**GCR**" without parameters, according to the following format:

Field Id.	Format	Description
CMD_ID	A3	Command code (= "**GCR**").

➲ Note #1

The processing of EMV cards requires knowledge of the supported AIDs, which are provided in the AID Tables (see **section 4.1.1**), and different Acquirers can support the processing of the same AIDs. Thus, when using option **GCR_ACQIDXREQ** = "00", the SPE must ensure that the combined set of loaded AID Tables does not have conflicting AID records. The pinpad does not do any treatment to solve this type of conflict and, if this restriction is not observed by the SPE, pinpad behavior will be unpredictable.

➲ Note #2

An SPE that supports CTLS shall call "**GCR**" initially allowing this interface using **GCR_CTLSON** = "1" (or omitting this parameter). However, the SPE shall disable this interface with **GCR_CTLSON** = "0" and resubmit the command in the following cases:

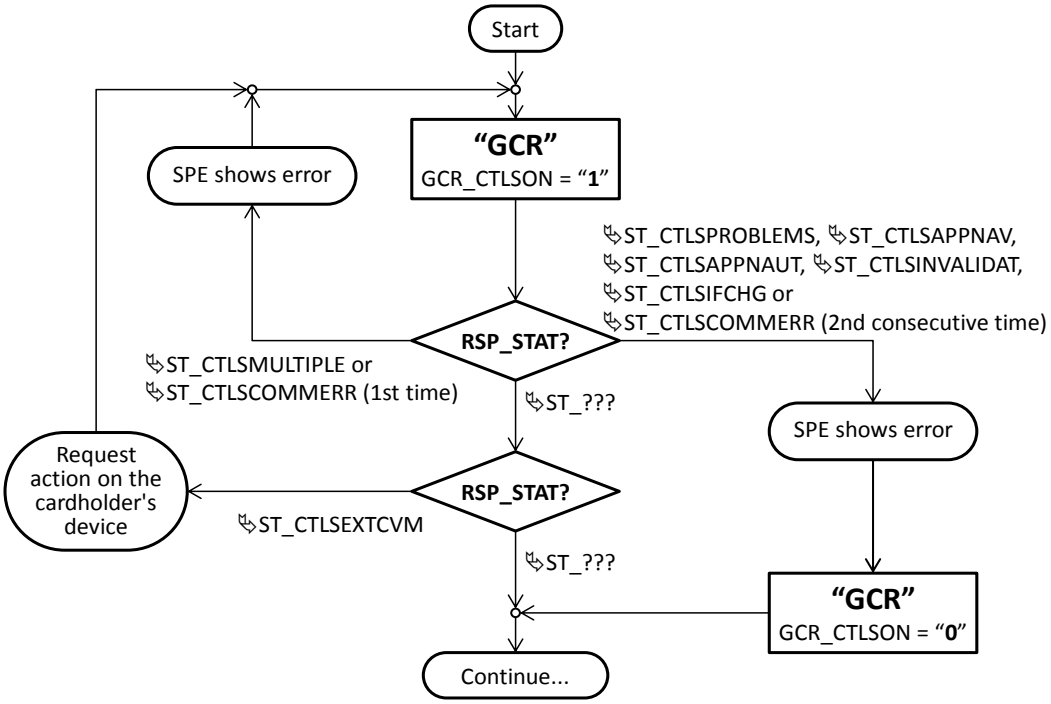
• When the command returns ↳ST_CTLSPROBLEMS, ↳ST_CTLSAPPNAV, ↳ST_CTLSAPPNAUT, ↳ST_CTLSINVALIDAT or ↳ST_CTLSIFCHG; or

• When the command returns ↳ST_CTLSCOMMERR for the second consecutive time.

➲ Note #3

If "**GCX**" returns ↳ST_CTLSEXTCVM, the SPE shall present a message to the cardholder requesting an action on his device (ex: "FOLLOW INSTRUCTIONS ON THE PHONE") and call the command again.

The following diagram illustrates this process:



➲ Examples

The SPE requests a card providing a list of three indexes from the network "02" (the CTLS interface is activated, as **GCR_CTLSON** is not provided).

SPE ⇒	47 43 52 30 35 32 30 30 30 30 30 30 30 30 30 30 30 30 31 30 30 30 31 33 31 32 30 37 31 30 32 33 35 35 38 37 36 35 32 33 34 35 36 38 30 33 30 32 31 34 30 32 32 32 30 32 31 37	GCR0520000000000 0010001312071023 5587652345680302 1402220217

Pinpad notifies the SPE about the application selected on the card.

⇐ PP	4E 54 4D 30 30 30 30 33 32 53 45 4C 45 43 49 4F 4E 41 44 4F 3A 20 20 20 20 43 52 45 44 49 54 4F 20 20 20 20 20 20 20 20 20	NTM000032SELECIO NADO:••••CREDITO •••••••••

An EMV ICC is inserted and successfully processed.

⇐ PP

```
47 43 52 30 30 30 33 35 32 30 33 30 30 31 30 32  GCR0003520300102
32 32 30 30 20 20 20 20 20 20 20 20 20 20 20 20  2200••••••••••••
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  ••••••••••••••••
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  ••••••••••••••••
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  ••••••••••••••••
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  ••••••••••••••••
33 37 35 34 36 34 39 39 39 39 31 30 37 36 39 39  3754649999107699
39 30 3D 31 39 31 32 32 30 36 32 38 31 32 33 36  90=1912206281236
36 39 30 37 37 30 30 30 30 30 20 20 20 20 20 20  6907700000••••••
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  ••••••••••••••••
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  ••••••••••••••••
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  ••••••••••••••••
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  ••••••••••••••••
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  ••••••••••••••••
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  ••••••••••••••••
20 20 31 36 35 34 36 34 39 39 39 39 31 30 37 36  ••16546499991076
39 39 39 30 20 20 20 30 30 43 52 45 44 49 54 4F  9990•••00CREDITO
20 20 20 20 20 20 20 20 20 30 30 30 4C 45 45 20  •••••••••000LEE•
57 45 49 4E 52 49 42 20 2F 20 47 45 44 44 59 20  WEINRIB•/•GEDDY•
20 20 20 20 20 20 31 39 31 32 33 31 30 30 30 30  ••••••1912310000
30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  0000000000000000
30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  0000000000000000
30 30 30 30 37 36 30 30 30                       000076000
```

SPE requests a card considering all the Acquirers and all table records, amount $ 28.37, enabling CTLS.

SPE ⇒

```
47 43 52 30 34 31 30 30 39 39 30 30 30 30 30 30  GCR0410099000000
30 30 32 38 33 37 31 34 30 37 32 34 31 31 33 39  0028371407241139
32 32 54 41 42 56 45 52 30 39 31 38 30 30 30     22TABVER0918000
```

A magnetic card is swiped, its data returns successfully, and there is also the indication of an error allowing fallback in a previous chip card processing.

⇐ PP

```
47 43 52 30 30 30 33 35 32 30 30 31 30 30 30 30  GCR0003520010000
30 30 37 36 42 34 34 34 34 33 33 33 33 32 32 32  0076B44443333222
32 31 31 31 31 5E 54 4F 4D 20 53 41 57 59 45 52  21111^TOM•SAWYER
5E 31 36 30 38 31 30 31 38 31 32 37 33 36 35 34  ^1608101812736 54
37 36 31 35 32 33 36 34 35 31 37 38 36 32 33 35  7615236451786235
34 38 37 36 31 32 33 37 36 34 35 37 36 31 32 33  4876123764576123
33 37 34 34 34 34 33 33 33 33 32 32 32 32 31 31  3744443333222211
31 31 3D 31 36 30 38 31 30 31 38 31 32 37 33 36  11=1608101812736
35 34 37 36 31 35 34 30 39 30 34 34 34 34 33 33  5476154090444433
33 33 32 32 32 32 31 31 31 31 3D 3D 31 36 30 38  3322221111==1608
31 30 31 38 31 32 37 33 36 35 34 37 36 31 35 32  1018127365476152
33 36 34 35 31 37 38 36 32 33 35 34 38 37 36 31  3645178623548761
32 33 37 36 34 35 37 36 31 32 33 3D 38 33 37 34  23764576123=8374
38 32 37 34 37 38 37 32 33 36 38 34 30 30 30 30  8274787236840000
31 39 39 31 20 20 20 20 20 20 20 20 20 20 20 20  1991••••••••••••
20 20 30 30 20 20 20 20 20 20 20 20 20 20 20 20  ••00••••••••••••
20 20 20 20 20 20 20 20 30 30 20 20 20 20 20 20  •••••••00•••••••
20 20 20 20 20 20 20 20 20 20 30 30 30 20 20 20  •••••••••000••••
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  ••••••••••••••••
20 20 20 20 20 20 30 30 30 30 30 30 30 30 30 30  ••••••0000000000
30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  0000000000000000
30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  0000000000000000
30 30 30 30 30 30 30 30 30                       000000000
```

3.6.2. "CNG" command

☑ **Obsolete**
☐ **Blocking**
☐ **Abecs**

This command allows the SPE to provide additional EMV parameters (including proprietary ones) to the pinpad to be used in the processing of "**GOC**" and "**FNC**" commands. These parameters can match those existing in the AID Table record (see **section 4.1.1**) corresponding to the application selected on the EMV chip card. In this case, the values are not changed in the tables, being only relevant for the current processing.

This command is extremely useful for solving specific situations not provided for by the EMV Table structure, such as, for example, the case of merchants that use more than one type of currency, or in the case of cards that require proprietary parameters outside the EMV standard.

⚠ This command can only be used after the successful execution of the "**GCR**" command, in the specific case of **GCR_CARDTYPE** = **"03"** (ICC EMV).

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= "**CNG**").
CMD_LEN1	N3	Length of the following data.
CNG_EMVDTLEN	N2	Number of bytes represented in **CNG_EMVDAT** (length ÷ 2).
CNG_EMVDAT	H..198	Sequence of specific parameters to be used for EMV processing in "**GOC**" and/or "**FNC**" commands, in TLV format (see section **7.1**).

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**CNG**").
RSP_STAT	N3	Relevant return codes (see **section 3.1.1**): ↳ST_INVCALL .....Previous **"GCR"** call did not successfully process an ICC EMV card. ↳ST_INVPARM .....TLV structure in **CNG_EMVDAT** does not parse correctly.

➲ Examples

SPE provides the values of the following EMV parameters for use in the processing:

→ *Terminal Capabilities* (tag 9F33h) = E0D0C8h
→ *Transaction Currency Code* (tag 5F2Ah) = 0840h
→ Issuer proprietary data (tag DF04h) = 169937823Fh

SPE ⇒	43 4E 47 30 34 30 31 39 39 46 33 33 30 33 45 30 44 30 43 38 35 46 32 41 30 32 30 38 34 30 44 46 30 34 30 35 31 36 39 39 33 37 38 32 33 46	CNG040199F3303E0 D0C85F2A020840DF 0405169937823F

Pinpad successfully accepts the data.		
⇐ **PP**	43 4E 47 30 30 30	CNG000

3.6.3. "GOC" command

☑ **Obsolete**
☑ **Blocking**
☐ **Abecs**

This command continues the chip card processing, as presented in **section 3.6.5**.

this command shall not be used if "**GCR**" has reported the swipe of a magnetic card (or CTLS magstripe mode).

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= "**GOC**").
CMD_LEN1	N3	Length of the following data (de **GOC_AMOUNT** a **GOC_ACQPR**).
GOC_AMOUNT	N12	New transaction amount (*Amount, authorized*) in cents, which may include new amounts submitted to the SPE after "**GCR**" (such as service fee, cash withdrawal or change). If there are no additions to the amount, it shall be same used in "**GCR**".
GOC_CASHBACK	N12	Portion of the transaction amount referring to cashback (*Amount, other*) in cents. If this value does not exist, this field must be filled with zeros.
GOC_EXCLIST	N1	Result of querying the Exception File (only for ICC EMV), if this feature is supported by the Acquire Network: "0" = Card appears on the Exception File. "1" = Card does not appear on the Exception File.
GOC_CONNECT	N1	Connection requirement (only for ICC EMV): "**0**" = Transaction may be offline approved. "**1**" = Transaction shall never be offline approved.
GOC_RFU1	N1	RFU (fixed "0").
GOC_METHOD	N1	Online PIN encryption method, to be used if required by EMV processing: **"1"** = MK/WK:TDES:PIN **"3"** = DUKPT:TDES:PIN
GOC_KEYIDX	N2	Slot index of the key to be used (MK:PIN or DUKPT:PIN).
GOC_WKENC	H32	Working Key encrypted by the MK. If **GOC_METHOD** = "3", the pinpad ignores this field.
GOC_RISKMAN	N1	**"1"** (fixed) = Always perform *Terminal Risk Management* using **GOC_FLRLIMIT**, **GOC_TPBRS**, **GOC_TVBRS** and **GOC_MTPBRS**:
GOC_FLRLIMIT	H8	*Terminal Floor Limit* (in cents)
GOC_TPBRS	N2	*Target Percentage to be used for Biased Random Selection*
GOC_TVBRS	H8	*Threshold Value for Biased Random Selection* (in cents)

Field Id.	Format	Description
GOC_MTPBRS	N2	*Maximum Target Percentage to be used for Biased Random Selection*
GOC_ACQPRLEN	N3	Length of **GOC_ACQPR**, in characters. ▪ If **GCR_ACQIDX** = "**01**", **GOC_ACQPRLEN** is "003"; ▪ If **GCR_ACQIDX** = "**02**", **GOC_ACQPRLEN** is "032"; and ▪ For other values of **GCR_ACQIDX**, the **GOC_ACQPR** field does not exist (**GOC_ACQPRLEN** is "000").
GOC_ACQPR	A..32	Input parameters specific to the selected Acquirer Network (see tables below).
CMD_LEN2	N3	Length of the following data (**GOC_TAGS1LEN** and **GOC_TAGS1**).
GOC_TAGS1LEN	N3	Number of bytes represented in **GOC_TAGS1** (length ÷ 2).
GOC_TAGS1	H..256	First tag list identifying the EMV data objects to be returned in **GOC_EMVDAT**. The tags must be simply concatenated, respecting their formation rule (see **section 7.1**).
CMD_LEN3	N3	Length of the following data.
GOC_TAGS2LEN	N3	Number of bytes represented in **GOC_TAGS2** (length ÷ 2).
GOC_TAGS2	H..256	Second tag list, additional to **GOC_TAGS1**. This field exists simply for historical reasons.

For GCR_ACQIDX = "01":

Field Id.	Format	Description
GOC_ACQPR	N2	*Transaction Type* (tag 9Ch)
	N1	**"0"** – PIN *bypass* not allowed. **"1"** – PIN *bypass* allowed.

For GCR_ACQIDX = "02":

Field Id.	Format	Description
GOC_ACQPR	S32	Message to be displayed during PIN capture, either online or offline, already formatted for 2 rows and 16 columns.

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**GOC**").

Field Id.	Format	Description
RSP_STAT	N3	Relevant return codes (see **section 3.1.1**): ↳ST_INVCALL .....Previous **"GCR"** call did not successfully process an ICC/CTLS EMV card ↳ST_ERRKEY.....MK/DUKPT not present in the pinpad. ↳ST_TIMEOUT.....PIN capture timeout. ↳ST_CARDPROBLEMS ...Invalid or faulty ICC. ↳ST_CARDINVDATA .....ICC with invalid or missing data. ↳ST_ERRFALLBACK.....ICC error that allows fallback to magnetic card.
RSP_LEN1	N3	Length of the following data.
GOC_DECISION	N1	Transaction outcome: **"0"** = Offline approved. **"1"** = Offline declined. **"2"** = Transaction requires online authorization.
GOC_SIGNAT	N1	Paper signature must be obtained ("0"-no / "1"-yes).
GOC_PINOFF	N1	PIN offline verified ("0"-no / "1"-yes).
GOC_ERRPINOFF	N1	Number of invalid offline PIN submissions in this transaction.
GOC_PBLOCKED	N1	Offline PIN was blocked on the last invalid presentation in this transaction ("0"-no / "1"-yes).
GOC_PINONL	N1	PIN was captured for online verification ("0"-no / "1"-yes). If this field is "0", **GOC_PINBLK** and **GOC_KSN** shall be ignored.
GOC_PINBLK	H16	Encrypted PIN.
GOC_KSN	H20	KSN (Key Serial Number) of the key used, in case of DUKPT method (**GOC_METHOD** = "3"). For MK / WK, this field is returned with zeros.
GOC_EMVDTLEN	N3	Number of bytes represented in **GOC_EMVDAT** (length ÷ 2).
GOC_EMVDAT	H..512	EMV transaction data to be sent to the Acquirer Network, in TLV format (see **section 7.1**). The pinpad concatenates the data objects requested by **GOC_TAGS1** and **GOC_TAGS2**, if found, respecting the order in which they were requested. EMV objects that contain card (or PAN) track information will not be returned by the pinpad!
GOC_ACQRDLEN	N3	Length of the Acquirer Network specific return data (**not used** - fixed "000").

➲ Examples

SPE requests the transaction continuation, changing the amount to $ 12.00, providing parameters for possible online PIN capture and EMV risk management parameters.

SPE ⇒	47 4F 43 30 38 36 30 30 30 30 30 30 30 31 32 30 30 30 30 30 30 30 30 30 30 32 30 30 30 30 31 33 30 31 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 31 30 30 30 30 31 33 38 38 32 30 30 30 30 30 30 33 45 38 38 30 30 30 30 30 32 35 30 31 31 38 32 39 46 32 37 39 46 32 36 39 46 33 36 39 35 38 46 39 46 33 37 30 30 33 30 30 30	GOC0860000000012 0000000000020000 1301000000000000 0000000000000000 0000100001388200 00003E8800000250 11829F279F269F36 958F9F37003000

Pinpad notifies the SPE of the need to capture the PIN.

⇐ PP	4E 54 4D 30 30 30 30 33 32 53 4F 4C 49 43 49 54 45 20 41 20 53 45 4E 48 41 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	NTM000032SOLICIT E•A•SENHA••••••• •••••••••

Operation is successful, with offline PIN capture, and the card asks for online authorization.

⇐ PP	47 4F 43 30 30 30 31 33 30 32 30 31 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 34 31 38 32 30 32 35 38 30 30 39 46 32 37 30 31 38 30 39 46 32 36 30 38 37 36 35 44 43 31 33 38 30 37 44 31 45 34 43 38 39 46 33 36 30 32 30 30 30 36 39 35 30 35 30 30 31 30 30 30 30 30 30 30 38 46 30 31 30 35 39 46 33 37 30 34 35 41 37 37 41 43 46 30 30 30 30	GOC0001302010000 0000000000000000 0000000000000000 000041820258009F 2701809F2608765D C13807D1E4C89F36 0200069505001000 00008F01059F3704 5A77ACF0000

3.6.4. "FNC" command

☑ **Obsolete**
☐ **Blocking**
☐ **Abecs**

This command ends the chip card processing and must be called if "**GOC**" has requested online approval (**GOC_DECISION** = "2"), as presented in **section 3.6.5**.

In the case of offline approval or denial (**GOC_DECISION** = "0" or "1"), this command may be called only to keep the same operational flow as an online transaction.

➲ Command

Field Id.	Format	Description
CMD_ID	A3	Command code (= "**FNC**").
CMD_LEN1	N3	Length of the following data (de **FNC_COMMST** a **FNC_ACQPRLEN)**.
FNC_COMMST	N1	Communication status with the Acquirer Network: **"0"** = Successful communication, with a valid response received in the online transaction (or the transaction was ended offline in "**GOC**"). **"1"** = It was not possible to communicate with the Acquirer Network. In this case, the remaining fields of this command must be zeros. **"9"** = Successful communication, transaction approved, but the *Authorization Response Code* is different from "00".
FNC_ISSMODE	N1	Issuer mode: fixed **"0"** (full grade EMV)
FNC_ARC	A2	*Authorization Response Code* (approval/denial code returned by Acquirer Network).
FNC_ISSDATLEN	N3	Number of bytes represented in **FNC_ISSDAT** (length ÷ 2).
FNC_ISSDAT	H..512	EMV objects received from the Acquirer Network, in TLV format (see section **7.1**).
FNC_ACQPRLEN	N3	Length of Acquirer Network specific input parameters (**not used** - fixed "000").
CMD_LEN2	N3	Length of the following data.
FNC_TAGSLEN	N3	Number of bytes represented in **FNC_TAGS** (length ÷ 2).
FNC_TAGS	H..256	List of tags identifying EMV data objects to be returned in **FNC_EMVDAT**. The tags must be simply concatenated, respecting their formation rule (see **section 7.1**).

➲ Response

Field Id.	Format	Description
RSP_ID	A3	Response code (= "**FNC**").

Field Id.	Format	Description
RSP_STAT	N3	Relevant return codes (see **section 3.1.1**): ↳ST_INVCALL ..... "**GOC"** has not been successfully executed previously. ↳ST_CARDPROBLEMS ...Invalid or faulty ICC. ↳ST_CARDINVDATA .....ICC with invalid of missing data.
RSP_LEN1	N3	Length of the following data.
FNC_DECISION	N1	Outcome: "0" = Transaction approved. "1" = Transaction declined by the card. "2" = Transaction declined by the Acquirer Network.
FNC_EMVDTLEN	N3	Number of bytes represented in **FNC_EMVDAT** (length ÷ 2).
FNC_EMVDAT	H..512	EMV transaction data to be sent to the Acquirer Network, in TLV format (see **section 7.1**). The pinpad concatenates the data objects requested by **FNC_TAGS**, if found, respecting the order in which they were requested. EMV objects that contain card (or PAN) track information will not be returned by the pinpad!
FNC_ISRLEN	N2	Number of bytes represented in **FNC_ISR** (length ÷ 2).
FNC_ISR	H..100	*Issuer Script Results*
FNC_ACQRDLEN	N3	Length of the Acquirer Network specific return data (**not used** - fixed "000").

➲ Examples

The SPE requests the EMV transaction completion. The Acquirer Network approves the transaction, also returning the *Issuer Authentication Data* (tag 91h).

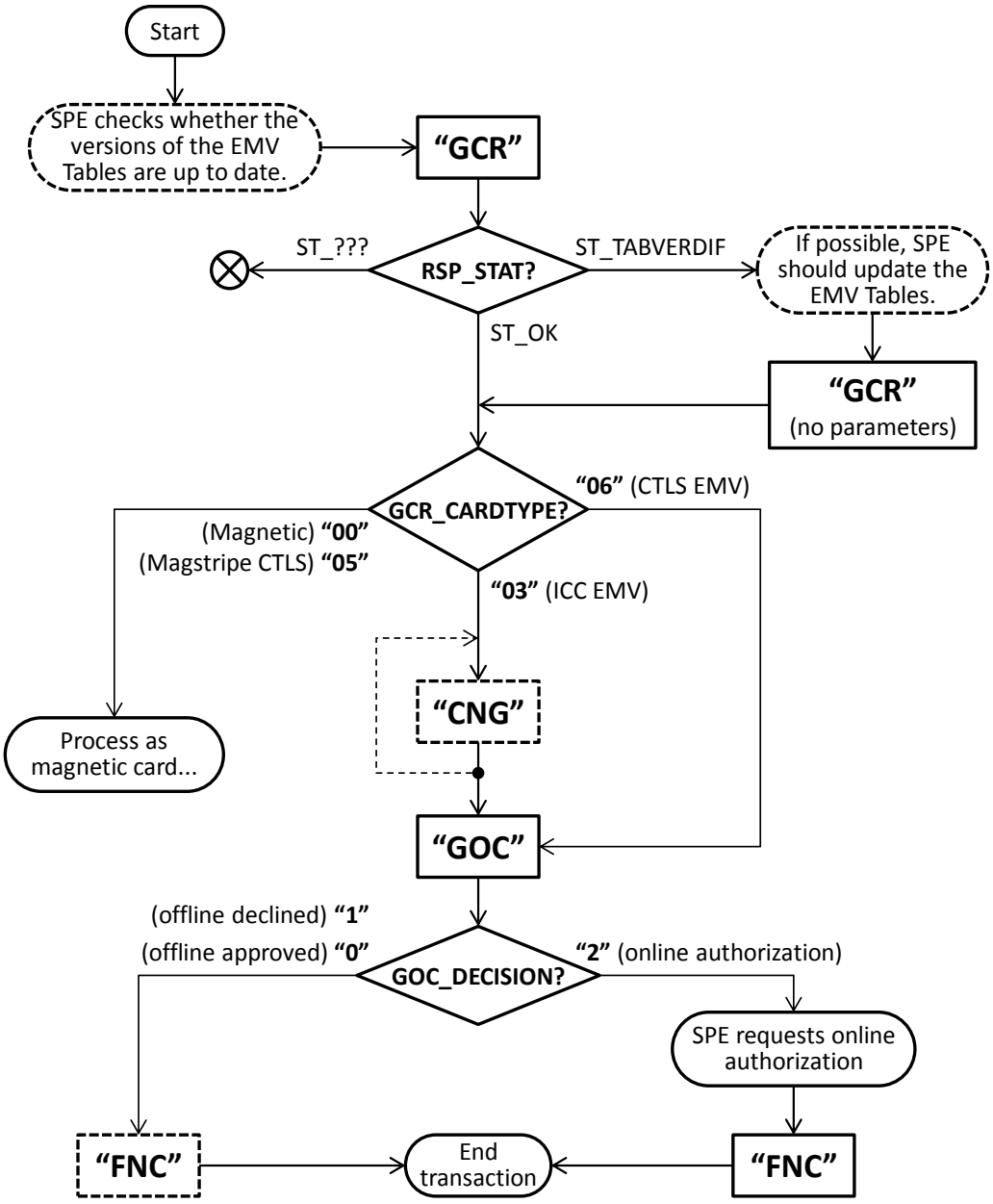
SPE ⇒	46 4E 43 30 33 30 30 30 30 30 30 31 30 39 31 30 38 45 36 34 41 32 46 45 32 31 46 44 38 38 36 37 32 30 30 30 30 32 35 30 31 31 38 32 39 46 32 37 39 46 32 36 39 46 33 36 39 35 38 46 39 46 33 37	FNC0300000010910 8E64A2FE21FD8867 2000025011829F27 9F269F36958F9F37

Operation is successful, but the card declines the transaction at the end (the SPE must undo the transaction with Acquirer Network).

⇐ PP	46 4E 43 30 30 30 30 39 31 31 30 34 31 38 32 30 32 35 38 30 30 39 46 32 37 30 31 30 30 39 46 32 36 30 38 36 39 45 42 41 33 42 45 31 43 43 38 42 33 38 44 39 46 33 36 30 32 30 30 30 36 39 35 30 35 30 30 31 30 30 30 30 30 30 30 38 46 30 31 30 35 39 46 33 37 30 34 35 41 37 37 41 43 46 30 30 30 30 30 30	FNC0000911041820 258009F2701009F2 60869EBA3BE1CC8B 38D9F36020006950 500100000008F010 59F37045A77ACF00 0000

3.6.5. Operation workflow

The following flow illustrates the calling sequence for obsolete card processing commands. Dotted blocks refer to optional processing that depends on the Acquirer Network specification.



3.7. Abecs Card Processing Commands

This section details high-level commands responsible for the complete processing of a card during a payment transaction, whether magnetic, ICC or CTLS.

The following commands are covered in this section:

CMD_ID	Meaning	Obsolete	Blocking	Abecs
"GCX"	Get Card - Extended	☐	☑	☑
"GED"	Get EMV Data	☐	☐	☑
"GOX"	Go On Chip Processing - Extended	☐	☑	☑
"FCX"	Finish Chip Processing - Extended	☐	☑	☑

⚠ Commands presented in this section are very flexible and their form of use depends deeply on the specifications of the Acquirer Network payment systems.

3.7.1. "GCX" command

☐ **Obsolete**
☑ **Blocking**
☑ **Abecs**

This command initiates a payment card transaction process (be it magnetic, ICC or CTLS), as presented in **section 3.7.5**.

It is equivalent to the "**GCR**" command, but with the following differences:

- Uses the Abecs format to allow flexibility and facilitate future developments.
- Automatically resolves any AID conflicts when considering the tables of all Acquirer Networks during processing.
- Does not perform version control of the EMV Tables. The SPE must perform this control independently through the commands in **section 3.5**, checking the version and, if necessary, updating the necessary tables before executing this command.
- Allows the SPE to send to the pinpad a list of EMV parameters to be used in the processing.
- Allows the SPE to obtain a list of any EMV data objects from the card.
- Returns incomplete track data, according to the security process described in **section 5.4**. To obtain the complete tracks (open or encrypted), one must use the "**GTK**" command.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= "**GCX**").
SPE_TRNTYPE	O	Transaction type to be performed: 00h = Payment; 01h = Cash; 09h = Payment with cashback; 20h = Refund; 30h = Balance inquiry; or Other values according to ISO 8583:1987. If this parameter is not provided, the pinpad will consider the transaction to be "payment" (if **SPE_CASHBACK** is absent) or "cashback" (if **SPE_CASHBACK** is present and not zero).
SPE_ACQREF	O	Acquirer Network identifier (**TAB_ACQ**) whose EMV Tables will be used if an ICC or CTLS is presented. If this parameter is not provided, the pinpad will consider the tables of all Acquirer Networks.
SPE_APPTYPE	O	Value(s) of **T1_APPTYPE** from the records of the AID Tables to be used in the processing. If this parameter is not provided, the pinpad will consider any value.
SPE_AIDLIST	O	Specific list of AID Table records to be used in the processing, each entry being composed of the concatenation of **TAB_ACQ** and **TAB_RECIDX**. **IMPORTANT**: If this parameter is present, **SPE_ACQREF** and **SPE_APPTYPE** will simply be ignored by the pinpad if they exist in the command.

Field Id.	Presence	Description / Remark
SPE_AMOUNT	O	Transaction amount in cents *(Amount, authorized)*. If this parameter is absent, the pinpad will consider this data to be zero.
SPE_CASHBACK	O	Cashback amount (*Amount, other*) in cents. If this parameter is absent, the pinpad will consider this data to be zero.
SPE_TRNCURR	O	*Transaction Currency Code*, only for ICC. If this parameter is absent, the pinpad uses the value defined in **T1_TRNCURR**.
SPE_TRNDATE	M	*Transaction Date.*
SPE_TRNTIME	M	*Transaction Time.*
SPE_GCXOPT	O	Command options: "0xxxx" = Wait for magnetic card or ICC; or "1xxxx" = Wait for magnetic card; ICC or CTLS; "x0xxx" = Show transaction amount on the card waiting prompt, if not zero. "x1xxx" = Do not show transaction amount. "xx000" = RFU. If this parameter is absent, the pinpad will consider this data to be zero ("00000").
SPE_PANMASK	O	Definitions for PAN masking in **PP_PAN**, **PP_TRK1INC**, **PP_TRK2INC** and **PP_TRK3INC** response fields. If absent, there is no masking.
SPE_EMVDATA	O	Optional list of EMV parameters (in TLV format). The data provided here have priority over the objects in the AID Tables, if they coincide.
SPE_TAGLIST	O	List of tags of the EMV objects to be returned in the response to the command.
SPE_TIMEOUT	O	Maximum waiting time, in seconds, for the cardholder to present the card or other action. If absent, this command never returns ↳ST_TIMEOUT.
SPE_DSPMSG	O	Message to be displayed on the pinpad display for the card request. If this parameter is not provided, the pinpad uses a standard message.

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= "**GCX**").

Field Id.	Presence	Description / Remark
RSP_STAT	M	Relevant return codes (see **section 3.1.1**): ↳ST_RSPOVRFL..... EMV data length exceeds maximum allowed for **PP_EMVDATA**. ↳ST_CARDINVALIDAT... ICC application is invalidated. ↳ST_CARDBLOCKED ..... ICC is blocked. ↳ST_CARDPROBLEMS... Invalid or faulty ICC. ↳ST_CARDINVDATA ..... ICC with invalid of missing data. ↳ST_CARDAPPNAV..... Invalid mode for the ICC. ↳ST_CARDAPPNAUT..... ICC not accepted. ↳ST_ERRFALLBACK..... ICC error that allows fallback to magnetic card. ↳ST_CTLSINVALIDAT .... CTLS is invalidated/blocked. ↳ST_CTLSPROBLEMS.... Invalid or faulty CTLS. ↳ST_CTLSAPPNAV ..... Invalid mode for the CTLS. ↳ST_CTLSAPPNAUT..... CTLS not accepted. ↳ST_CTLSEXTCVM..... Request verification on the cardholder's device. ↳ST_CTLSIFCHG..... Change interface (use ICC or magnetic card).
PP_CARDTYPE	M	Processed card type: **"00"** = Magnetic; **"03"** = ICC EMV; **"05"** = CTLS magstripe mode; or **"06"** = CTLS EMV.
PP_ICCSTAT	MD	This field is returned only if **PP_CARDTYPE** = "00" (magnetic card), being mandatory in this case. Status of the last ICC processing. The SPE uses this information to refuse (or not) a magnetic card if its tracks indicate chip presence. "**0**" = Successful (or another status that does not imply fallback); or "**1**" = Error allowing to fallback; or "**2**" = Required application not supported (fallback depends on the Acquirer Network settings).
PP_AIDTABINFO	MD	This field is returned only if **PP_CARDTYPE** ≠ "00" (ICC or CTLS), being mandatory in this case. It contains a list of which the records of the AID Tables were used in the processing, being the concatenation of **TAB_ACQ**, **TAB_RECIDX** and **T1_APPTYPE**. **IMPORTANT**: If more than one Acquirer Network is able to process the card, this field may contain a list with multiple entries.

Field Id.	Presence	Description / Remark
PP_PAN	MD	Card number read (PAN), which can be masked according to **SPE_PANMASK**. This field is only returned if **PP_CARDTYPE** = “03” (ICC EMV) or “06” (CTLS EMV), being mandatory in these cases.
PP_PANSEQNO	MD	*Application PAN Sequence Number*. This field is only returned if **PP_CARDTYPE** = “03” (ICC EMV) or “06” (CTLS EMV), being mandatory in these cases.
PP_TRK1INC	O	Incomplete Track 1, if read from the magnetic card. PAN may be masked according to **SPE_PANMASK**.
PP_TRK2INC	O	Incomplete Track 1, if read from the magnetic card. PAN may be masked according to **SPE_PANMASK**
PP_TRK3INC	O	Incomplete Track 1, if read from the magnetic card. PAN may be masked according to **SPE_PANMASK**
PP_CHNAME	O	*Cardholder Name*, if present in the card (ICC or CTLS). This field is not returned by the pinpad if **PP_CARDTYPE** = "00" (magnetic card) or "05" (CTLS magstripe mode).
PP_LABEL	MD	Label of the application being processed. This field is returned only if **PP_CARDTYPE** ≠ "00" (ICC or CTLS), being mandatory in this case.
PP_ISSCNTRY	O	*Issuer Country Code*, if present in the card (ICC or CTLS). This field is not returned by the pinpad if **PP_CARDTYPE** = "00" (magnetic card) or "05" (CTLS magstripe mode).
PP_CARDEXP	O	*Application Expiration Date*, if present in the card (ICC or CTLS). This field is not returned by the pinpad if **PP_CARDTYPE** = "00" (magnetic card) or "05" (CTLS magstripe mode).
PP_EMVDATA	MR	List of EMV objects defined by **SPE_TAGLIST**. Objects not found are simply not returned by the pinpad, as well as objects that contain card track information (or PAN). This field is mandatory whenever **SPE_TAGLIST** exists in the command, even if no object is found (in which case it is returned with zero length).
PP_DEVTYPE	MD	Type of CTLS device used (if **PP_CARDTYPE** = “05” or “06”): “00” = Card; “01” = Mobile device (i.e. smartphone); “02” = Keyring; “03” = Watch; “04” = Mobile tag; “05” = Bracelet; “06” = Mobile device case/sleeve; “10” = Tablet or e-reader; Other values = Future use. In the absence of this field, the “card” device is assumed.

> ⚠ If a magnetic card has been swiped (**PP_CARDTYPE** = “00”) but no track could be successfully read, **RSP_STAT** = ↳ST_OK and the **PP_TRK1INC**, **PP_TRK2INC** and **PP_TRK3INC** fields will not be returned.

➲ Note #1

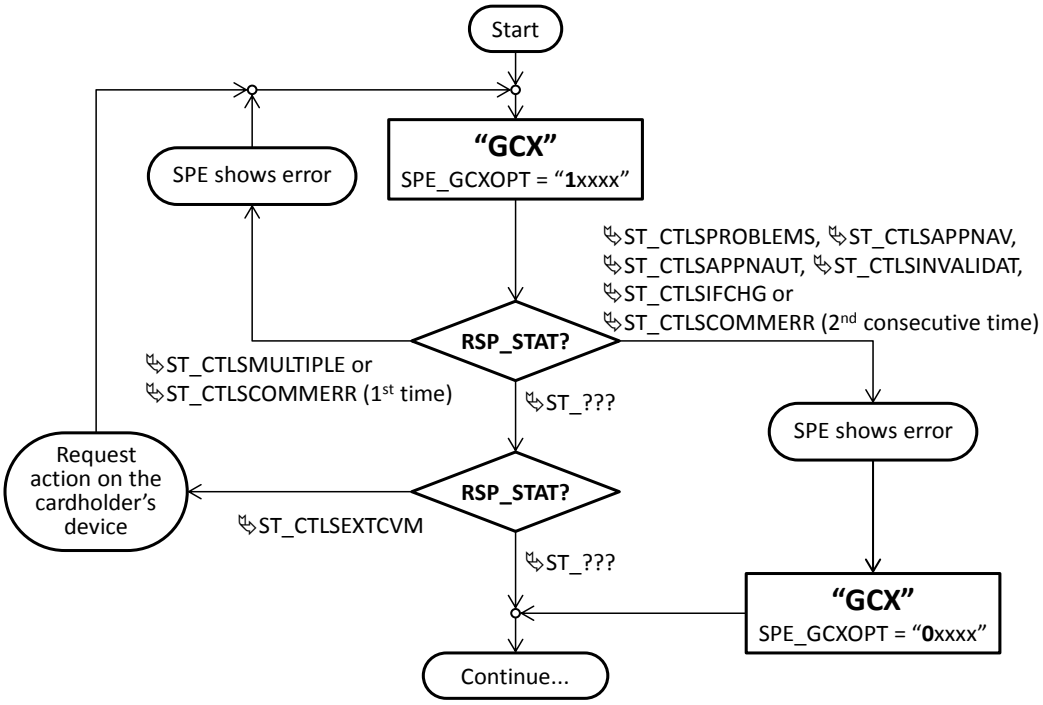
A SPE that supports CTLS must call “**GCX**” initially allowing this interface using **SPE_GCXOPT** = “**1**xxxx”. However, the SPE must disable this interface using **SPE_GCXOPT** = “**0**xxxx” (or omitting this parameter) and resubmit the command in the following cases:

- When the command returns ↳ST_CTLSPROBLEMS, ↳ST_CTLSAPPNAV, ↳ST_CTLSAPPNAUT, ↳ST_CTLSINVALIDAT or ↳ST_CTLSIFCHG; or
- When the command returns ↳ST_CTLSCOMMERR for the second consecutive time.

➲ Note #2

If “**GCX**” returns ↳ST_CTLSEXTCVM, the SPE must present a message to the cardholder requesting an action on his device (ex: “FOLLOW INSTRUCTIONS ON THE PHONE”) and call the command again.

The following diagram illustrates this process:



➲ Examples

SPE starts processing a transaction with the following characteristics:

- Amount $483.00, without cashback;
- Magnetic card or ICC only (CTLS not allowed);
- Use all records in the AID Table of Acquirer Network "08";
- Force the E0F8C8h value for *Terminal Capabilities* (tag 9F33h); and
- Request the following EMV data objects if they exist in the card: *Issuer Country Code* (tag 5F28h) and *Application Expiration Date* (tag 5F24h).

SPE ⇒	47 43 58 30 36 39 00 17 00 05 30 30 30 30 30 00 10 00 02 30 38 00 13 00 0C 30 30 30 30 30 30 30 34 38 33 30 30 00 15 00 06 31 33 30 39 30 31 00 16 00 06 32 30 31 38 34 37 00 05 00 06 9F 33 03 E0 F8 C8 00 04 00 04 5F 28 5F 24	GCX069....00000. ...08....0000000 48300....130901. ...201847....Ÿ3. àøÈ...._(_$

Pinpad successfully processes an ICC EMV.

⇐ PP	47 43 58 30 30 30 31 32 35 80 55 00 08 4A 4F 48 4E 20 44 4F 45 80 52 00 10 34 34 34 34 33 33 33 33 32 32 32 32 31 31 31 31 80 42 00 18 34 34 34 34 33 33 33 33 32 32 32 32 31 31 31 31 3D 31 36 30 38 32 30 31 80 4F 00 02 30 33 80 51 00 06 30 38 30 33 30 31 80 53 00 02 30 31 80 54 00 0B 5F 28 02 00 76 5F 24 03 16 08 31 80 5B 00 06 52 C9 44 49 54 4F 80 5C 00 04 30 30 37 36 80 5D 00 06 31 36 30 38 33 31	GCX000125€U..JOH N•DOE€R..4444333 322221111€B..444 4333322221111=16 08201€O..03€Q..0 80301€S..01€T.._ (..v_$...1€[..RÉ DITO€\..0076€].. 160831

SPE starts processing a transaction with the following characteristics:

- Amount $1,128.00, with $128.00 cashback;
- All types of card are allowed (magnetic, ICC and CTLS);
- Use a specific list of records from the AID Tables;
- Set a 42-second timeout; and
- Define the message to be used when requesting the card as "PLEASE MY FRIEND, USE YOUR CARD AS YOU WANT!".

SPE ⇒	47 43 58 31 33 34 00 0C 00 01 2A 00 1B 00 2C 50 4C 45 41 53 45 20 4D 59 20 46 52 49 45 4E 44 2C 20 55 53 45 20 59 4F 55 52 20 43 41 52 44 20 41 53 20 59 4F 55 20 57 41 4E 54 21 00 12 00 10 30 31 30 31 30 32 30 35 30 33 30 38 32 35 30 34 00 13 00 0C 30 30 30 30 30 30 31 31 32 38 30 30 00 14 00 0C 30 30 30 30 30 30 30 31 32 38 30 30 00 15 00 06 31 34 30 37 32 35 00 16 00 06 30 38 32 35 35 39 00 17 00 05 31 30 30 30 30	GCX134....*...,P LEASE•MY•FRIEND, •USE•YOUR•CARD•A S•YOU•WANT!....0 101020503082504. ...000000112800. ...000000012800. ...140725....082 559....10000

Pinpad successfully processes a CTLS, however it informs that the transaction can be processed by two different Acquirer Networks in the SPE.

⇐ PP	47 43 58 30 30 30 30 39 36 80 42 00 19 35 30 30 39 38 32 33 37 32 33 34 32 33 38 30 30 32 3D 31 37 30 31 36 30 30 80 4F 00 02 30 36 80 51 00 0C 30 32 30 35 30 33 32 35 30 34 30 33 80 52 00 11 35 30 30 39 38 32 33 37 32 33 34 32 33 38 30 30 32 80 53 00 02 30 30 80 5B 00 07 50 41 59 50 41 53 53 80 5C 00 03 38 34 30	GCX000096€B..500 98237234238002=1 701600€O..06€Q.. 020503250403€R.. 5009823723423800 2€S..00€[..PAYPA SS€\..840

3.7.2. “GED” command

☐ **Obsolete**
☐ **Blocking**
☑ **Abecs**

This command allows the SPE to obtain data from the EMV processing, provided that the “**GCX**” command has previously been successfully executed for an ICC EMV (**PP_CARDTYPE** = “03”), a CTLS magstripe mode (**PP_CARDTYPE** = “05”) or a CTLS EMV (**PP_CARDTYPE** = “06”).

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**GED**”).
SPE_TAGLIST	M	List of tags of the EMV objects to be returned in the response to the command.

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= “**GED**”).
RSP_STAT	M	Relevant return codes (see **section 3.1.1**): ↳ST_INVCALL..... “**GCX**” has not been successfully executed previously for ICC/CTLS. ↳ST_RSPOVRFL..... EMV data length exceeds maximum allowed for **PP_EMVDATA**.
PP_EMVDATA	**M**	List of EMV objects defined by **SPE_TAGLIST**. Objects not found are simply not returned by the pinpad, as well as objects that contain card track information (or PAN).

➲ Examples

SPE requests the following EMV objects if they exist on the card: *Application Usage Control* (tag 9F07h), *Application Version Number* (tag 9F08h), *ADF Name* (4Fh) and a proprietary object of tag DF55h.

SPE ⇒	47 45 44 30 31 31 00 04 00 07 9F 07 9F 08 4F DF 55	GED011....Ÿ.Ÿ.Oß U

Pinpad returns the requested objects except for the *Application Version Number* (tag 9F08h), as it is unknown in this processing.

⇐ PP	47 45 44 30 30 30 30 32 39 80 54 00 19 9F 07 02 FF 00 4F 07 A0 00 00 00 03 10 10 DF 55 08 11 22 33 44 55 66 77 88	GED000029€T..Ÿ.. ÿ.O. ......ßU..” 3DUfwˆ

3.7.3. “GOX” command

☐ **Obsolete**
☑ **Blocking**
☑ **Abecs**

This command continues the chip card process if the “**GCX**” command has previously been successfully executed for an ICC EMV (**PP_CARDTYPE** = “03”) or CTLS EMV (**PP_CARDTYPE** = “06”), as shown in the flow in **section 3.7.5**.

It is equivalent to the “**GOC**” command, but with the following differences:

- Uses the Abecs format to allow flexibility and facilitate future developments.
- Allows the SPE to send to the pinpad a list of EMV parameters to be used in the processing (useful only in the case of ICC EMV!).
- Allows the SPE to define the message to be presented on the display if a PIN entry is required.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**GOX**”).
SPE_ACQREF	M	Identifier of the Acquirer Network whose EMV Tables will be used in the processing.
SPE_TRNTYPE	O	Transaction type to be performed: 00h = Payment; 01h = Cash; 09h = Payment with cashback; 20h = Refund; 30h = Balance inquiry; or Other values according to ISO 8583:1987. If this parameter is not provided, the pinpad will consider the transaction to be “payment” (if **SPE_CASHBACK** absent) or “cashback” (if **SPE_CASHBACK** present and not zero).
SPE_AMOUNT	O	Transaction amount in cents *(Amount, authorized)*. If this parameter is absent, the pinpad will consider this data to be zero.
SPE_CASHBACK	O	Cashback amount (*Amount, other*) in cents. If this parameter is absent, the pinpad will consider this data to be zero.
SPE_TRNCURR	O	*Transaction Currency Code*, only for ICC. If this parameter is absent, the pinpad uses: ▪ The value informed in **SPE_TRNCURR** in “**GCX**”, if any; or ▪ The value defined in **T1_TRNCURR**.
SPE_GOXOPT	O	Command options: “1xxxx” = PAN is in the Exception List (only for ICC EMV). “x1xxx” = Transaction shall not be offline approved (only for ICC EMV). “xx1xx” = Do not allow PIN bypass. “xxx00” = RFU. If this parameter is absent, the pinpad will consider this data to be zero (“00000”).

Field Id.	Presence	Description / Remark
SPE_MTHDPIN	M	Online PIN encryption method, to be used if required by EMV processing: **"1"** = MK/WK:TDES:PIN; and **"3"** = DUKPT:TDES:PIN (see **section 5.1.1**).
SPE_KEYIDX	M	Slot index of the key to be used (MK:PIN or DUKPT:PIN).
SPE_WKENC	MD	Working Key encrypted by the MK, to be used for Online PIN capture when **SPE_MTHDPIN** = "**1**" (being mandatory in this case).
SPE_DSPMSG	O	Message to be displayed in the event of a PIN capture. If this parameter is not provided, the pinpad uses a standard message.
SPE_TRMPAR	O	*Terminal Risk Management* parameters, the concatenation of the following data: ▪ *Terminal Floor Limit* ("X4" format, in cents); ▪ Target *Percentage to be used for Biased Random Selection* ("X1" format); ▪ Threshold *Value for Biased Random Selection* ("X4" format, in cents); and ▪ Maximum *Target Percentage to be used for Biased Random Selection (*"X1" format). If this field is absent, the pinpad performs *Terminal Risk Management* with zero values.
SPE_EMVDATA	O	Optional parameter list (in TLV format), for use in **ICC EMV** processing only. The data provided here have priority over the objects in the AID Tables, if they coincide.
SPE_TAGLIST	O	List of tags of the EMV objects to be returned in the response to the command.
SPE_TIMEOUT	O	PIN capture timeout. If this field is absent, the pinpad will consider 1 minute (60 seconds).

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= "**GOX**").

Field Id.	Presence	Description / Remark
RSP_STAT	M	Relevant return codes (see **section 3.1.1**): ↳ST_INVCALL..... **"GCX**" has not been successfully executed previously for ICC CTLS/EMV. ↳ST_RSPOVRFL..... EMV data length exceeds maximum allowed for **PP_EMVDATA**. ↳ST_ERRKEY ..... MK/DUKPT not present in the pinpad. ↳ST_TIMEOUT ..... PIN capture timeout. ↳ST_CARDPROBLEMS... Invalid or faulty ICC. ↳ST_CARDINVDATA..... ICC with invalid of missing data. ↳ST_ERRFALLBACK ..... ICC error that allows fallback to magnetic card.
PP_GOXRES	M	EMV processing outcome: "0xxxxx" = Transaction offline approved; "1xxxxx" = Transaction declined; or "2xxxxx" = Transaction requires online approval. "x1xxxx" = Signature on paper. "xx1xxx" = Successful offline PIN verification. "xx2xxx" = PIN captured for online verification. "xxx1xx" = Cardholder verification performed on the mobile device (smartphone, for example) "xxxx00" = RFU.
PP_PINBLK	MD	Encrypted PIN for online verification. This field is mandatory if **PP_GOXRES** = "xx**2**xxx".
PP_KSN	MD	KSN (Key Serial Number) of the key used for PIN encryption, in case of DUKPT method. This field is mandatory if **PP_GOXRES** = "xx**2**xxx" and **SPE_MTHDPIN** = **"3"** (DUKPT:TDES:PIN).
PP_EMVDATA	MR	List of EMV objects defined by **SPE_TAGLIST**. Objects not found are simply not returned by the pinpad, as well as objects that contain card track information (or PAN). This field is mandatory whenever **SPE_TAGLIST** exists in the command, even if no object is found (in which case it is returned with zero length).

➲ Examples

SPE requests the continuation of an ICC EMV processing with the following characteristics:

- Use EMV Tables of Acquirer Network “08”;
- Transaction amount $234.50, with a $100.00 cashback;
- If an online PIN is required, use DUKPT:TDES slot “07”;
- Perform *Terminal Risk Management* with: Floor Limit = $100.00; *Target Percentage to be used for Biased Random Selection* = 20%; *Threshold Value for Biased Random Selection* = $25.00; *Maximum Target Percentage to be used for Biased Random Selection* = 80%;
- No optional EMV parameters; and
- Request the following EMV data objects if they exist: *TVR* (tag 95h), *Application Cryptogram* (tag 9F26h), *Cryptogram Information Data* (tag 9F27h), *Issuer Application Data* (tag 9F10h), *CVM Results* (tag 9F34h) and *ATC* (tag 9F36h).

SPE ⇒	47 4F 58 31 31 36 00 13 00 0C 30 30 30 30 30 30 30 32 33 34 35 30 00 14 00 0C 30 30 30 30 30 30 30 31 30 30 30 30 00 02 00 01 33 00 09 00 02 30 37 00 1B 00 22 43 52 C9 44 49 54 4F 0D 52 24 20 32 33 34 2C 35 30 0D 44 49 47 49 54 45 20 53 55 41 20 53 45 4E 48 41 00 1A 00 0A 00 00 27 10 14 00 00 00 19 50 00 04 00 0B 95 9F 26 9F 27 9F 10 9F 34 9F 36 00 10 00 02 30 38	GOX116....000000 023450....000000 010000....3....0 7...”CRÉDITO.R$• 234,50.DIGITE•SU A•SENHA.....’.. ....P....•Ÿ&Ÿ’Ÿ. Ÿ4Ÿ6....08
Pinpad successfully performs the operation (the card requires online authorization), returning the required EMV data.		
⇐ PP	47 4F 58 30 30 30 30 38 38 80 56 00 06 32 30 32 30 30 30 80 54 00 30 95 05 00 80 00 00 00 9F 26 08 E0 DB 51 A3 74 2F EA 83 9F 27 01 80 9F 10 0C 2C 51 4D 27 0F C3 CD 87 6C A4 00 00 9F 34 03 42 03 02 9F 36 02 00 4C 80 57 00 08 B9 DF 0A 99 6E A6 CC B7 80 4C 00 0A FF FF F7 98 41 00 34 40 00 08	GOX000088€V..202 000€T.0•..€...Ÿ& .àÛQ£t/êƒŸ’.€Ÿ.. ,QM’.ÃÍ‡l¤..Ÿ4.B ..Ÿ6..L€W..¹ß.™n ¦Ì·€L..ÿÿ÷˜A.4@. .

3.7.4. “FCX” command

☐ **Obsolete**
☑ **Blocking**
☑ **Abecs**

This command is equivalent to the “**FNC**” command but using Abecs format. It finalizes chip card processing and shall always be called if “**GCX**” has requested online approval (**PP_GOXRES** = “**2**xxxx”), as shown in **section 3.7.5**.

In case of offline approval or denial (**PP_GOXRES** = “**0**xxxx” or “**1**xxxx”), this command may be called, according to the Acquirer Network specification (for example, for the execution of maintenance *Issuer Script Processing* on ICC).

In case of CTLS, this command can request a new presentation of the same card processed in “**GCX**” for the execution of maintenance *Issuer Scripts*, a situation in which the command assumes a blocking behavior.

➲ Command

Field Id.	Presence	Description / Remark
CMD_ID	M	Command code (= “**FCX**”).
SPE_FCXOPT	M	Result of communication with the Acquirer Network: “0xxx” = Transaction approved by the Acquirer. “1xxx” = Transaction declined by the Acquirer. “2xxx” = Unable to go online (or invalid response from the Acquirer). “x000” = RFU.
SPE_ARC	MD	*Authorization Response Code* (approval/denial code returned by Acquirer Network), mandatory if **SPE_FCXOPT** = “**0**xxx” or “**1**xxx”.
SPE_EMVDATA	O	TLV objects optionally received from the Acquirer Network, which may contain the *Issuer Authentication Data* (tag 91h) and *Issuer Scripts* (tags 71h and 72h).
SPE_TAGLIST	O	List of tags of the EMV objects to be returned in the response to the command.
SPE_TIMEOUT	O	Maximum waiting time, in seconds, for the cardholder to present the CTLS a second time (if required).

➲ Response

Field Id.	Presence	Description / Remark
RSP_ID	M	Response code (= “**FCX**”).

Field Id.	Presence	Description / Remark
RSP_STAT	M	Relevant return codes (see **section 3.1.1**): ↳ST_INVCALL..... **"GOX"** has not been successfully executed previously. ↳ST_RSPOVRFL..... EMV data length exceeds maximum allowed for para **PP_EMVDATA**. ↳ST_CARDPROBLEMS... Invalid or faulty ICC. ↳ST_CARDINVDATA..... ICC with invalid of missing data.
PP_FCXRES	M	Outcome: "**0xx**" = Transaction approved; or "**1xx**" = Transaction declined. "x00" = RFU.
PP_EMVDATA	MR	List of EMV objects defined by **SPE_TAGLIST**. Objects not found are simply not returned by the pinpad, as well as objects that contain card track information (or PAN). This field is mandatory whenever **SPE_TAGLIST** exists in the command, even if no object is found (in which case it is returned with zero length).
PP_ISRESULTS	O	*Issuer Script Results*, only present if the command receives *Issuer Scripts* in **SPE_EMVDATA**.

➲ Examples

SPE requests the completion of an ICC EMV processing with the following characteristics:

- Acquirer Network approves the online transaction, but with "Y3" as response code;
- Acquirer Network returns *Issuer Authentication Data* (tag 91h) and *Issuer Script* (tag 72h); and
- Request the following EMV data objects if they exist: *TVR* (tag 95h), *Application Cryptogram* (tag 9F26h), *Cryptogram Information Data* (tag 9F27h) and *Issuer Application Data* (tag 9F10h).

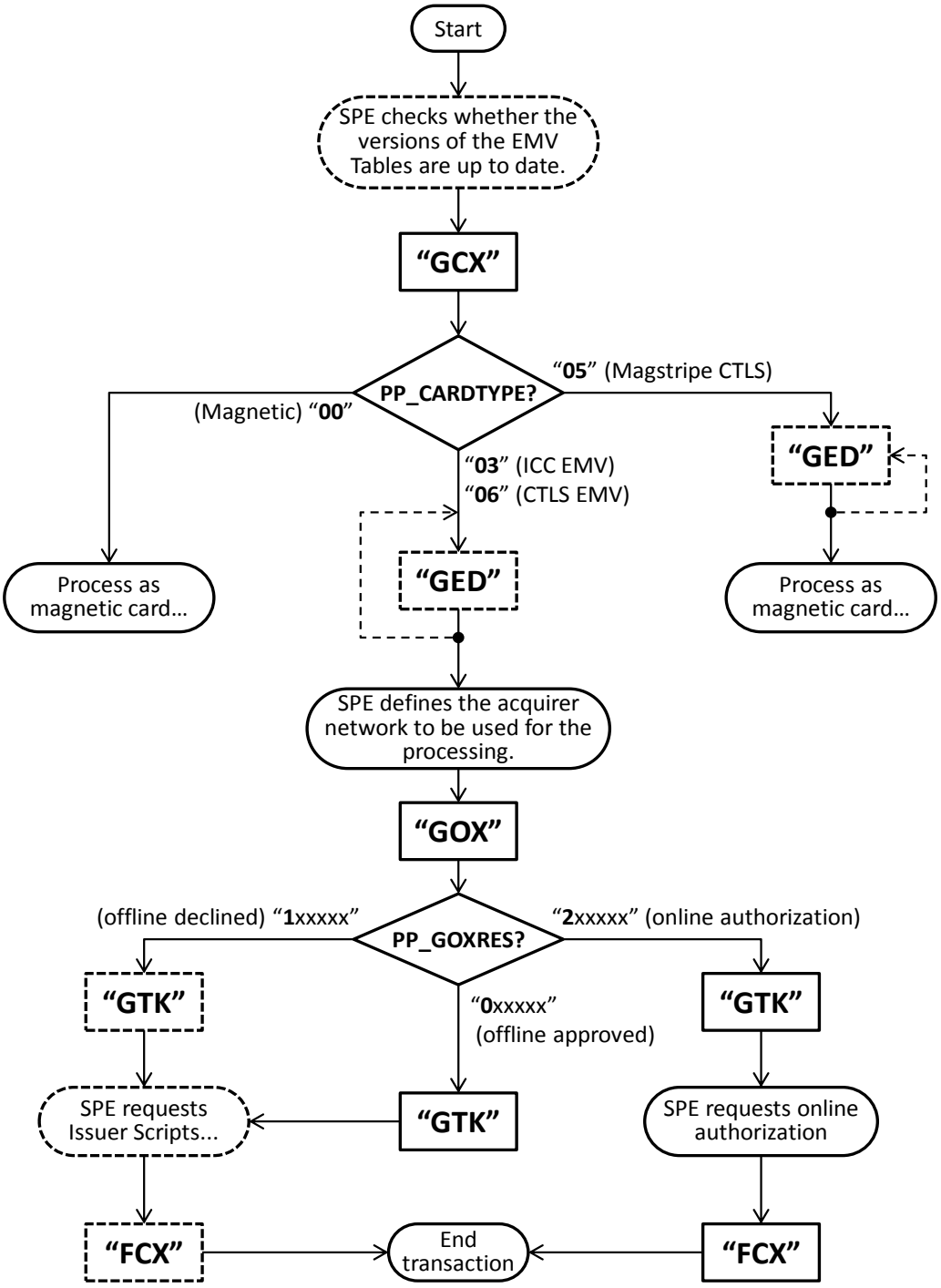
SPE ⇒	46 43 58 30 35 39 00 05 00 1E 91 08 A1 02 DB 6D 41 C6 79 63 72 12 9F 18 00 86 0D 84 24 00 00 08 A0 71 54 4A 23 76 1A A1 00 04 00 07 95 9F 26 9F 27 9F 10 00 1C 00 02 59 33 00 19 00 04 30 30 30 30	FCX059....'.¡.Ûm AÆycr.Ÿ..†.„$... qTJ#v.¡....•Ÿ&Ÿ 'Ÿ.....Y3....000 0

Pinpad successfully completes the operation (approval) and returns the *Issuer Script Results*, as well as the requested EMV objects.

⇐ PP	46 43 58 30 30 30 30 35 35 80 56 00 03 30 30 30 80 59 00 05 20 00 00 00 00 80 54 00 23 95 05 00 80 00 00 00 9F 26 08 95 24 B3 FC 02 5E 51 72 9F 27 01 40 9F 10 0A 7D 89 5F FF F0 15 D7 72 FB C9	FCX000055€V..000 €Y.......€T.#•.. €...Ÿ&.•$³ü.^QrŸ '.@Ÿ..}‰_ÿð.×rûÉ

3.7.5. Operation workflow

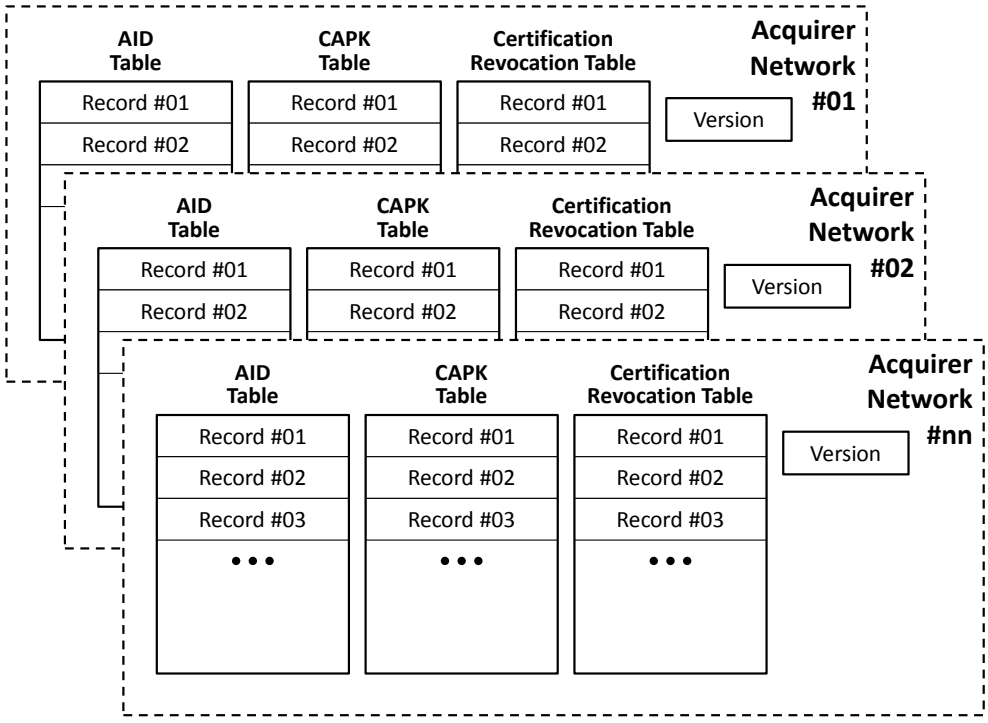
The following flow illustrates the calling sequence for Abecs card processing commands. Dotted blocks refer to optional processing that depends on the acquirer network specification.



4. EMV Tables Management

To optimize EMV card processing (ICC or CTLS) in the commands described in sections **3.6** and **3.7**, the pinpad needs to be preloaded with a set of parameter tables, which are stored in a non-volatile manner (they are preserved even when the pinpad is turned off).

These tables are separated by acquirer network according to the diagram:



These tables are generated by the SPE (from the information received from the Acquirer Networks) and are transferred to the pinpad using the commands described in **section 3.5**.

⚠ Consistency of the records sent to the pinpad is the sole responsibility of the SPE, since pinpad does not make complex criticisms, such as, for example, identifying collisions of indexes and records. The pinpad simply ignores records whose contents are notably invalid.

4.1. Types of Tables

The records of the tables, regardless of their type, have the following standardized format:

Field Id.	Format	Description
TAB_LEN	N3	Total record length, including this field.
TAB_ID	N1	Table type: "1" = AID Table; "2" = CAPK Table; or "3" = Certification Revocation Table.
TAB_ACQ	N2	Acquirer Network identifier responsible for the table (from "01" to "99").
TAB_RECIDX	A2	Record index (for "01" to "ZZ").
...	..	...

Notes:

- Each record must have a unique **TAB_RECIDX** (not necessarily sequential) for a given Acquirer Network.
- **TAB_ID**, **TAB_ACQ** and **TAB_RECIDX** together uniquely identify a record in a table.

4.1.1. AID Tables

These tables contain the *Application Identifiers* (AIDs) of supported EMV applications and several other parameters to be used in the processing, either for ICC or CLTS. The parameters that have direct correspondence with the EMV standards are identified by their "tags".

Each table is composed of one or more records with the following layout, with the AID (*Application Identifier*) as the "search key":

Field Id.	Format	Tag	Description
TAB_LEN	N3		Record length, including this field. The pinpad must be able to accept records of: ▪ **284 bytes**: corresponding to the 🕮**BibComp** specification (fields after **T1_ARCOFFLN** are not provided). ▪ **314 bytes**: corresponding to the v2.0x specification (fields after **T1_CTLSTACONL** are not provided). ▪ **340 bytes**: corresponding to this specification. ▪ **>340 bytes**: for future specifications (disregard any extra data received).
TAB_ID	N1		AID Table type (fixed **"1"**).
TAB_ACQ	N2		Acquirer Network identifier responsible for the table (from "01" to "99").
TAB_RECIDX	A2		Record index (for "01" to "ZZ").
T1_AIDLEN	N2		AID length, in bytes (from "05" to "16").
T1_AID	H32		AID - *Application Identifier* (left aligned).
T1_APPTYPE	N2		Application type, for use in "**GCR**" or "**GCX**" commands (from "01" to "98").
T1_DEFLABEL	S16		Default application label (obsolete - not used).
T1_ICCSTD	N2		Application standard: fixed "**03**" (EMV).
T1_APPVER1	H4	9F09h	*Application Version Number (Terminal)* - option #1
T1_APPVER2	H4	9F09h	*Application Version Number (Terminal)* - option #2
T1_APPVER3	H4	9F09h	*Application Version Number (Terminal)* - option #3
T1_TRMCNTRY	N3	9F1Ah	*Terminal Country Code*
T1_TRNCURR	N3	5F2Ah	*Transaction Currency Code*
T1_TRNCRREXP	N1	5F36h	*Transaction Currency Exponent*
T1_MERCHID	A15	9F16h	*Merchant Identifier*
T1_MCC	N4	9F15h	*Merchant Category Code*
T1_TRMID	A8	9F1Ch	*Terminal Identification*
T1_TRMCAPAB	H6	9F33h	*Terminal Capabilities*
T1_ADDTRMCP	H10	9F40h	*Additional Terminal Capabilities*
T1_TRMTYP	N2	9F35h	*Terminal Type*
T1_TACDEF	H10	DF9F0Dh	*Terminal Action Code – Default*
T1_TACDEN	H10	DF9F0Eh	*Terminal Action Code – Denial*
T1_TACONL	H10	DF9F0Fh	*Terminal Action Code – Online*

Field Id.	Format	Tag	Description
T1_FLRLIMIT	H8	9F1Bh	*Terminal Floor Limit* (Default value to be used before **"GOC"**), in cents, expressed in the currency defined in **T1_TRNCURR**.
T1_TCC	A1	9F53h	*Transaction Category Code*
T1_CTLSZEROAM	A1		Indicates the action for CTLS if the transaction amount is zero: "1" = Supported, but online only; "0" or other value = Not supported.
T1_CTLSMODE	A1		Ability to handle the AID, if it is found in a CTLS: "1" or "2" = Supports VISA qVSDC; "3" or "4" = Supports MasterCard PayPass M/Chip; "5" or "6" = Supports Amex Expresspay EMV Mode; "7" = Supports Pure Contactless; "8" or "9" = Supports Discover D-PAS EMV Mode; "A" = Supports JCB Contactless (future use) "B" = Supports UnionPay QuickPass (future use); and "C" = Supports Interac Flash (future use) "0" or other value= Not supported
T1_CTLSTRNLIM	H8	DF8124h	*Terminal/Reader Contactless Transaction Limit*, in cents, expressed in the currency defined in **T1_TRNCURR**.
T1_CTLSFLRLIM	H8	DF8123h	*Terminal/Reader Contactless Floor Limit*, in cents, expressed in the currency defined in **T1_TRNCURR**.
T1_CTLSCVMLIM	H8	DF8126h	*Terminal/Reader CVM Required Limit*, in cents, expressed in the currency defined in **T1_TRNCURR**.
T1_CTLSAPPVER	H4	9F6Dh	*PayPass Mag Stripe Application Version Number* (Terminal)
T1_RFU1	N1		RFU (fixed "0").
T1_TDOLDEF	H40		*Default Transaction Certificate Data Object List (TDOL)* (filled with "00" to the right)
T1_DDOLDEF	H40		*Default Dynamic Data Authentication Data Object List (DDOL)* (filled with "00" to the right)
T1_ARCOFFLN	A8		*Authorization Response Codes* for offline transactions. This field is ignored by the pinpad, as these codes were fixed since the EMV 4.0 standard and are no longer parameters. Just by convention, keep "Y1Z1Y3Z3".
T1_CTLSTACDEF	H10(B5)	DF8120h	*Terminal Action Code – Default* for CTLS. If **T1_LEN** < 314, the pinpad assumes the value of **T1_TACDEF**.
T1_CTLSTACDEN	H10(B5)	DF8121h	*Terminal Action Code – Denial* for CTLS. If **T1_LEN** < 314, the pinpad assumes the value of **T1_TACDEN**.

Field Id.	Format	Tag	Description
T1_CTLSTACONL	H10(B5)	DF8122h	*Terminal Action Code – Online* for CTLS. If **T1_LEN** < 314, the pinpad assumes the value of **T1_TACONL**.
T1_CTLSTRMCP	H6(B3)	9F33h	*Terminal Capabilities* for CTLS. If **T1_LEN** < 340, the pinpad assumes the value of **T1_TRMCAPAB**.
T1_MOBCVM	N1		Support to cardholder verification on the device used to make the transaction (smartphone, for example). "1" = Yes / "0" = No If **T1_LEN** < 340, the pinpad assumes the value "0".
T1_CTLSADDTC	H10(B5)	9F40h	*Additional Terminal Capabilities* for CTLS. If **T1_LEN** < 340, the pinpad assumes the value of **T1_ADDTRMCP**.
T1_CTLSMBTLIM	H8	DF8125h	*Terminal/Reader Contactless Transaction Limit - Mobile*, in cents, expressed in the currency defined in **T1_TRNCURR**. If **T1_LEN** < 340, the pinpad assumes the value of **T1_CTLSTRNLIM**.
T1_CTLSISSSCR	N1		Support to *Issuer Scripts* for CTLS "1" = Yes / "0" = No If **T1_LEN** < 340, the pinpad assumes the value "0".

➲ Examples

TAB_ACQ = "02", TAB_RECIDX = 4D43h ("MC"): MasterCard credit with CTLS support (current specification)

```
340102MC07A0000000041010000000000000000001CTLESS CREDIT 03000100020001
07698620205030000000040000000000E0F8E87000F0F00122205004A000D800E8000020
5004F80000000000R040000138700009C3000005DB123409F02065F2A029A039C0195059F
370400000000009F37040000000000000000000000000000000000Y1Z1Y3Z3F00000000000
00000000F000000000E0484817000F0F001000013870
```

TAB_ACQ = "17", TAB_RECIDX = 3031h ("01"): MasterCard credit with CTLS support (V2.0x specification).

```
3141170107A0000000041010000000000000000001CTLESS CREDIT 03000100020001
07698620205030000000040000000000E0F8E87000F0F00122205004A000D800E8000020
5004F80000000000R040000138700009C3000005DB123409F02065F2A029A039C0195059F
370400000000009F37040000000000000000000000000000000000Y1Z1Y3Z3F00000000000
00000000F000000000
```

TAB_ACQ = “23”, TAB_RECIDX = 3132h (“12”): Visa Electron with no CTLS support (specification prior to V2.0x).

```
2841231207A0000000032010000000000000000000002Electron          03008400830082
0769862MERCHID9182672X1234TID01877E0F0C07000F0F00122000000000004800000000000
0004F800000007D010000001F499000007D010000001F4999F02069F030600000000000000000
00000000000000000000000000000000000000000000000000000000Y1Z1Y3Z3
```

4.1.2. CAPK Tables

These tables contain the *Certification Authority Public Keys*, used by EMV cards in offline authentication and PIN encryption processes.

Each table is composed of one or more records with the following layout, with *RID* and *CAPK Index* together as the “search key”.

Field Id.	Format	Tag	Description
TAB_LEN	N3		Record length, including this field (fixed “611”).
TAB_ID	N1		CAPK Table type (fixed **“2”**).
TAB_ACQ	N2		Acquirer Network identifier responsible for the table (from “01” to “99”).
TAB_RECIDX	A2		Record index (for “01” to “ZZ”).
T2_RID	H10		RID - *Registered Application Provider Identifier*
T2_CAPKIDX	H2	9F22h	*Certification Authority Public Key Index*
T2_RFU1	N2		RFU - fixed “00”.
T2_EXPLEN	N1		Length in bytes of the *Certification Authority Public Key Exponent* (“1” or “3”)
T2_EXP	H6		*Certification Authority Public Key Exponent* (left aligned)
T2_MODLEN	N3		Length in bytes of the *Certification Authority Public Key Modulus* (up to “248”)
T2_MOD	H496		*Certification Authority Public Key Modulus* (left aligned).
T2_CHKSTAT	N1		**T2_CHECKSUM** field status. “0” = Not used (obsolete, preferably use “1”); or “1” = Present.
T2_CHECKSUM	H40		*Certification Authority Public Key Check Sum*
T2_RFU2	N42		RFU - Fill with zeros (“0000...00”).

➲ Examples

TAB_ACQ = “01”, TAB_RECIDX = 3033h (“03”): American Express public key of index 0Eh.

```
61120103A0000000250E001030000144AA94A8C6DAD24F9BA56A27C09B01020819568B81A0
26BE9FD0A3416CA9A71166ED5084ED91CED47DD457DB7E6CBCD53E560BC5DF48ABC380993B
6D549F5196CFA77DFB20A0296188E969A2772E8C4141665F8BB2516BA2C7B5FC91F8DA04E8
D512EB0F6411516FB86FC021CE7E969DA94D33937909A53A57F907C40C22009DA7532CB3BE
509AE173B39AD6A01BA5BB850000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000
00000000001A7266ABAE64B42A3668851191D49856E17F8FBCD000000000000000000000000
0000000000000000000
```

➲ Table Merging

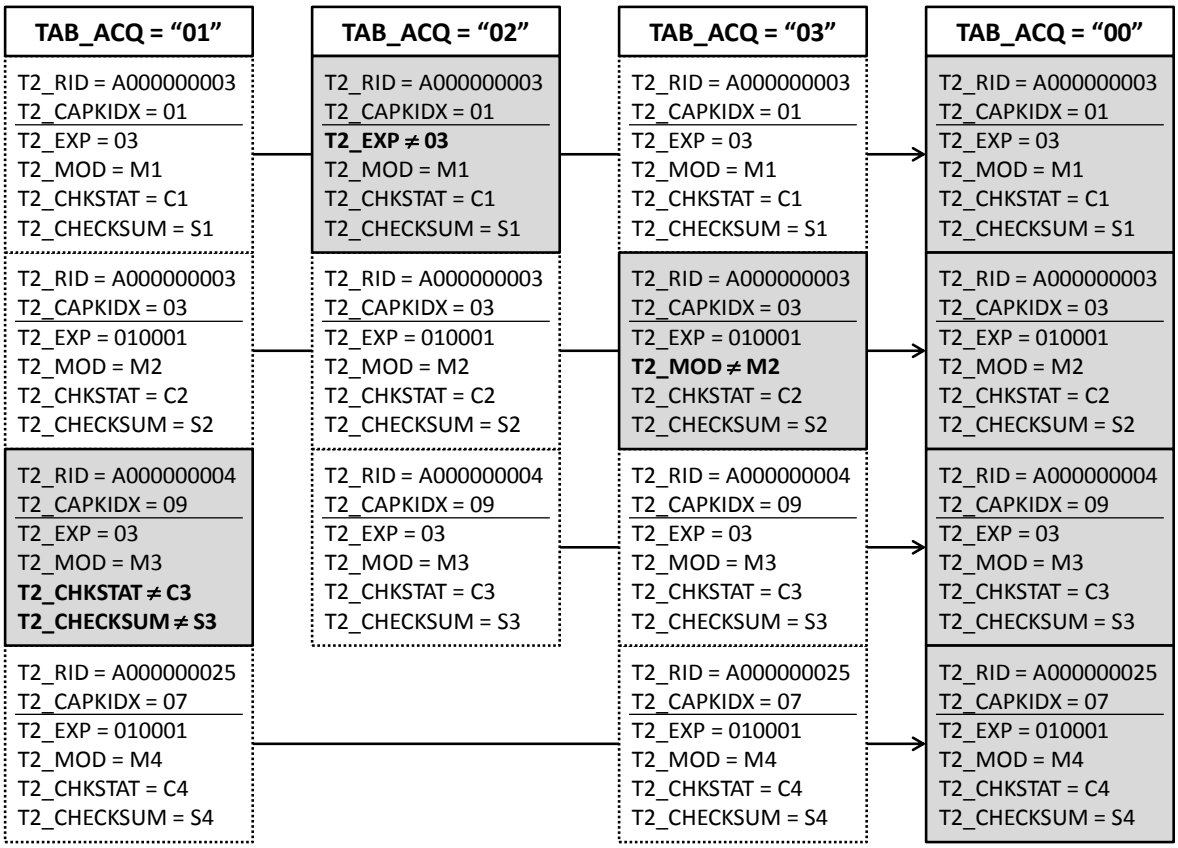
The CAPK Tables contain RSA public keys defined by the Card Associations, which, other than in exceptional situations, must be the same for all Acquirer Networks able to process their cards.

The records in these tables are large and, as the number of Acquirer Networks increases, they generate overhead in the communication between the SPE and the pinpad during the update process (to say nothing of the device's memory consumption).

To minimize overhead in this process, this specification provides for an optional mechanism through which these tables can be “merged” by the SPE when identified as “redundant”:

- Public keys identified as “redundant” among two or more Acquirer Networks may be loaded into a table with **TAB_ACQ** = “**00**”. In this case, the key no longer exists in the specific tables for these Acquirers.
- Keys are considered “redundant” when all of their fields are identical (**T2_RID**, **T2_CAPKIDX**, **T2_EXP**, **T2_MOD**, **T2_CHKSTAT** and **T2_CHECKSUM**).
- Although the fields **T2_RID** and **T2_CAPKIDX** uniquely define a public key for a Card Association, it is necessary to foresee the situation in which its data is defined differently by the Acquirer Networks. In this case, these keys are not considered “redundant” and, therefore, must be kept in the specific tables of the Acquirer Networks.

The following diagram illustrates this process:



⚠ Merged tables are only used by Abecs card processing commands (described in **section 3.7**) and are not recognized by obsolete commands.

⚠ This merging process only makes sense when the SPE uses a "unified" table management (see **section 4.2.1**), since records with TAB_ACQ = "00" cannot be loaded in the pinpad when the management is "separated" (see **section 4.2.2**).

4.1.3. Certification Revocation Tables

These tables contain the serial numbers of revoked *Issuer Public Key Certificates*.

Each table is composed of one or more records with the following layout, with *RID*, *CAPK Index* and *Certificate Serial Number* together as the "search key".

Field Id.	Format	Tag	Description
TAB_LEN	N3		Record length, including this field (fixed "026").
TAB_ID	N1		Certification Revocation Table type (fixed **"3"**).
TAB_ACQ	N2		Acquirer Network identifier responsible for the table (from "01" to "99").
TAB_RECIDX	A2		Record index (for "01" to "ZZ").

Field Id.	Format	Tag	Description
T3_RID	H10		RID - *Registered Application Provider Identifier*
T3_CAPKIDX	H2	9F22h	*Certification Authority Public Key Index*
T3_CERTSN	H6		*Certificate Serial Number*

➲ Examples

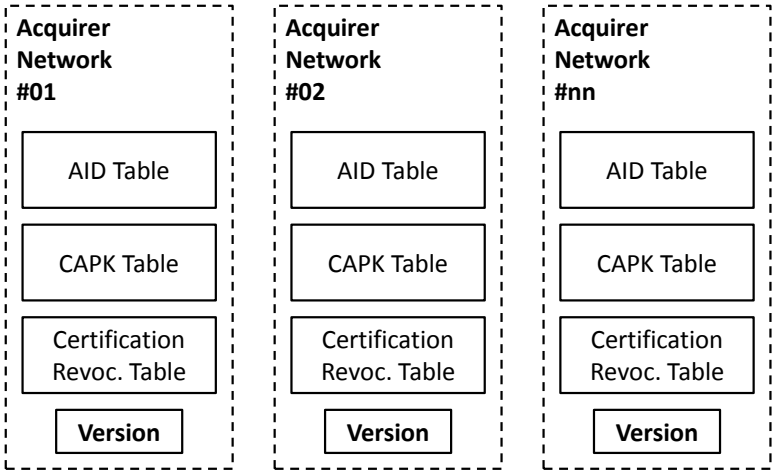
TAB_ACQ = “02”, TAB_RECIDX = 3233h (“23”): MasterCard certificate of serial number 333333h.

02620223A000000004FE333333

4.2. Table Versions

The EMV Tables have version information so that the SPE can control the need (or not) to update them on the pinpad.

Each set of tables for an Acquirer Network has a different version, as shown in the diagram:



This version information consists of a 10-character field that can be obtained using “**GTS**” or “**GIX**” (with **PP_TABVERnn**) commands.

Depending on the philosophy of the SPE, it can operate in two ways:

- Manage the tables of all Acquirer Networks in a unified way; or
- Manage the Acquirer Network tables independently.

4.2.1. Unified Management

When the SPE operator does not pre-select the Acquirer Network before making a transaction, a unified management of the tables is recommended, through the following rules:

- The tables of all Acquirer Networks are loaded in a single moment, using **TLI_ACQIDX = "00"** in the "**TLI**" command.
- **TLI_TABVER** version informed in the "**TLI**" command becomes valid for the tables of all Acquirer Networks.
- The "**GCR**" command must be activated with **GCR_ACQIDXREQ** = "00", with **GCR_TABVER** referring to the common version of all tables.
- The "**GCX**" command must be called without the **SPE_ACQREF** parameter.

4.2.2. Separated Management

When the SPE pre-selects the Acquirer Network before carrying out a transaction, a separated table management is recommended, through the following rules:

- The tables of each Acquirer Network may be loaded at different moments, using **TLI_ACQIDX ≠ "00"** in the "**TLI**" command. In this case, only the tables of the referred Network are changed, the others being preserved.
- **TLI_TABVER** version informed in the "**TLI**" command becomes valid only for the tables of the referred Acquirer Network. From this moment on, the "**GTS**" will return the version "0000000000" if called with **GTS_ACQIDX** = "**00**".
- The "**GCR**" command must be activated with **GCR_ACQIDXREQ ≠ "00"**, with **GCR_TABVER** referring only to the table version of the desired Acquirer Network.
- The "**GCX**" command must be activated with parameter **SPE_ACQREF** ≠ "**00**".

5. Security

This chapter details the cryptographic security mechanisms used by this specification, providing explanations regarding the keys injected by the pinpad manufacturer, as well as the processes designed to ensure the confidentiality of the information transmitted in the communication with the SPE.

5.1. Key Mapping

The pinpads have in their memory, in a protected area, several encryption keys “injected” by the manufacturer, considering two different algorithms:

- **MK/WK TDES**; and
- **DUKPT TDES.**

These keys are used by commands of this specification for encryption of cardholder PIN and for other data (“DAT”), being referenced by a two-digit numeric index.

Thus, this specification considers the following key mapping, differentiating four types for each existing numeric index:

Index ⇩	MK:TDES		DUKPT:TDES	
	PIN	DAT	PIN	DAT
“00”				
“01”				
“02”				
...				
“31”				
“32”				

➲ Important considerations:

- From the factory injection point of view, PIN and data keys (“DAT”) do not have any special treatment. It is just a logical separation to comply with PCI restrictions (a key used for PIN encryption cannot be used for other purposes).
- The following commands use only PIN keys: **“GDU”**, “**GPN**”, “**GOC**” and “**GOX**”.
- The following commands use only data (“DAT”) keys: “**DWK**”, “**EBX**”, “**ENB**” and “**GTK**”.
- Index “00” is valid and, considering that the maximum allowed index is “99”, one can have up to 100 keys of each type. However, the number of possible keys for each type depends on the pinpad model (for example, a given pinpad allows up to 18 DUKPT:TDES keys, from indexes "00" to "17").
- DUKPT:TDES data keys (“DAT”) allow different variants at the time of use (see **section 5.1.1**), however the existence of these variants does not require any special treatment in the factory injection process.

5.1.1. DUKPT:TDES encryption

DUKPT:TDES encryption is defined by the **ANSI X9.24:2009** standard, which includes five variants for modifying the key used. This specification considers only some of these variants, as shown in the following table:

Description in ANSI X9.24:2009	Constant used to change the key	Reference in this specification
PIN Encryption	00 00 00 00 00 00 00 FF 00 00 00 00 00 00 00 FF	**DUKPT:TDES:PIN**
Message Authentication, request or both ways	00 00 00 00 00 00 FF 00 00 00 00 00 00 00 FF 00	Not used.
Data Encryption, request or both ways **(*)**	00 00 00 00 00 FF 00 00 00 00 00 00 00 FF 00 00	**DUKPT:TDES:DAT#3**
Message Authentication, response	00 00 00 00 FF 00 00 00 00 00 00 00 FF 00 00 00	Not used.
Data Encryption, response **(*)**	00 00 00 FF 00 00 00 00 00 00 00 FF 00 00 00 00	Not used.

(*) In addition to the modification constant, these two variants add an additional diversification of the key using TDES, as described in **section A.4.1** of the **ANSI X9.24:2009** standard.

⚠ Whenever this specification considers data block encryption using DUKPT, regardless of the modality (ECB or CBC) or the variant used, the pinpad must use the same "Current Transaction Key" (a single KSN) for all 8-byte parts of the block, regardless of the number of iterations required for the process.

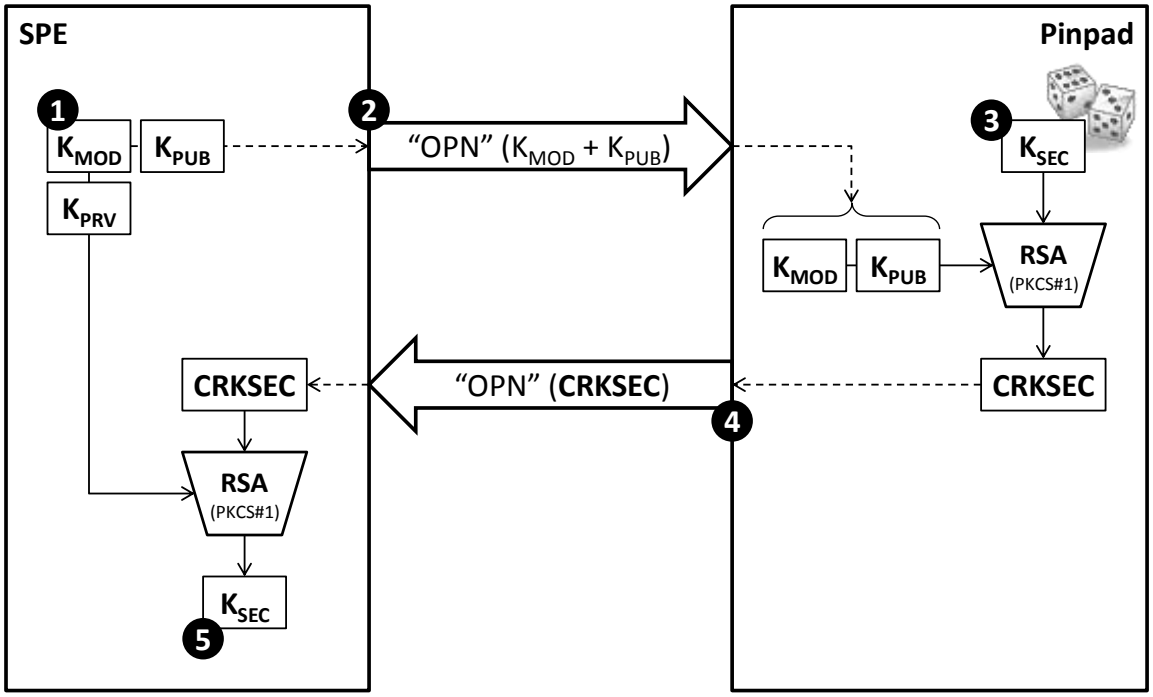
5.2. Secure Communication

This specification provides for a modality called "**Secure Communication**" in which data transmitted via the serial interface between the SPE and the pinpad are encrypted using **AES** algorithm with a "$\mathbf{K_{SEC}}$" key.

This method is intended to make it difficult to monitor the serial interface, minimizing the risk of fraud.

5.2.1. Establishment

The following flow illustrates the process of establishing "**Secure Communication**":



❶ The SPE creates an RSA key (or uses a fixed hardcoded key). This specification considers that the module must have 256 bytes (it may be increased in the future).

❷ The SPE sends $\mathbf{K_{MOD}}$ and $\mathbf{K_{PUB}}$ to the pinpad using the "**OPN**" command.

❸ The pinpad randomly generates a 16-byte $\mathbf{K_{SEC}}$ and encrypts it using RSA with $\mathbf{K_{MOD}/K_{PUB}}$ key. For this purpose, the block format recommended by the PKCS # 1 standard (table below) is used as the algorithm input, which must have the same size as the $\mathbf{K_{MOD}}$.

❹ The pinpad returns the generated cryptogram (**CRKSEC**) in the response to the "**OPN**" command.

❺ The SPE decodes the cryptogram (**CRKSEC**) received using RSA with $\mathbf{K_{MOD}/K_{PRV}}$ key, thus obtaining the random $\mathbf{K_{SEC}}$ key generated by the pinpad.

PKCS #1 block format:

Format	Description
B2	Header (fixed: **00h 02h**).
Bxxx	Random bytes other than 00h. The size "xxx" must be calculated so that this structure has same the total size as $\mathbf{K_{MOD}}$.
B1	Separator (fixed: **00h**).
B16	Random key generated by the pinpad ($\mathbf{K_{SEC}}$).

➲ Example:

A detailed example of the process for establishing "Secure Communication" is found in **section 3.2.2**.

5.2.2. Packet exchange

With "Secure Communication" established, the SPE and the pinpad now are able exchange encrypted packets through the serial interface. For this, command and response data must be encapsulated in the following format:

Nome	Format	Description
DATALEN	X2	Length of **CLRDATA** field (up to 2044 bytes).
DATACRC	X2	CRC-16 of the data contained in the **CLRDATA** field.
CLRDATA	???	Command or response data.
FILLER	B..15	It must be filled with 00h bytes so that the total size of this structure is a multiple of 16.

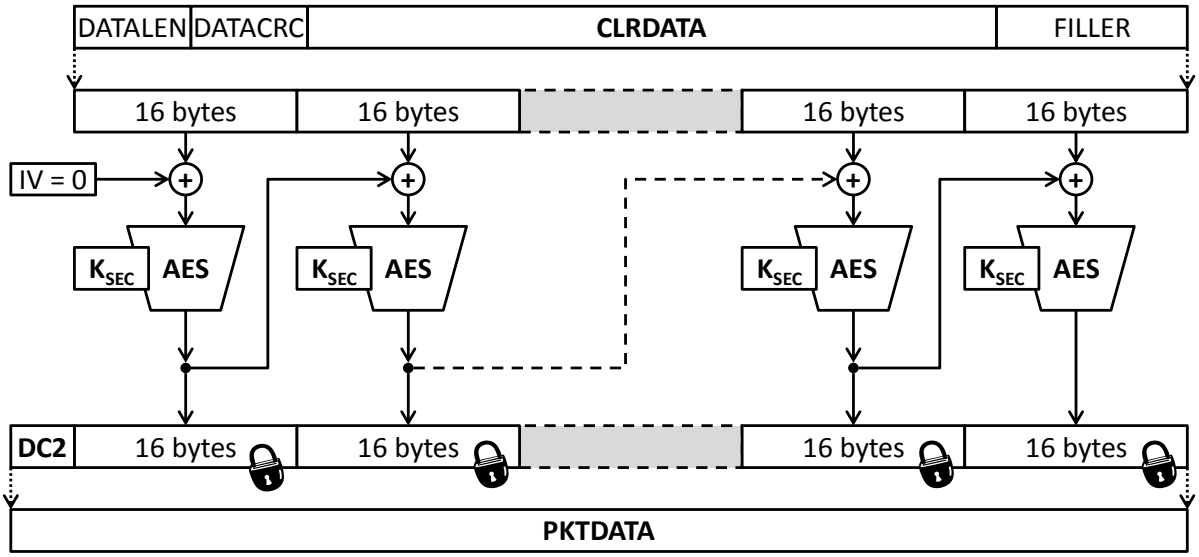
⚠ After "Secure Communication" is established, the SPE should only send encrypted commands (except for "**OPN**"). If the pinpad receives a cleartext command in this mode, it will return ↳ST_ERRPKTSEC for the command in question. The response error will be returned in cleartext, although "Secure Communication" remains active.

⚠ After "Secure Communication" is established, the pinpad will always return encrypted responses, including notification messages ("NTM"), except for "**CLO**" and "**CLX**" responses, which are always returned "cleartext".

⚠ Regardless of the "Secure Communication" status, the "**OPN**" **(secure or classic)** command, can only be sent in cleartext.

5.2.2.1. Encrypted Packet Sending

Regardless of the direction (SPE ↔ pinpad), the command/response data (**CLRDATA**) shall be embedded in the layout described above and encrypted using AES algorithm with $\mathbf{K_{SEC}}$ key in CBC mode, as shown in the following diagram:



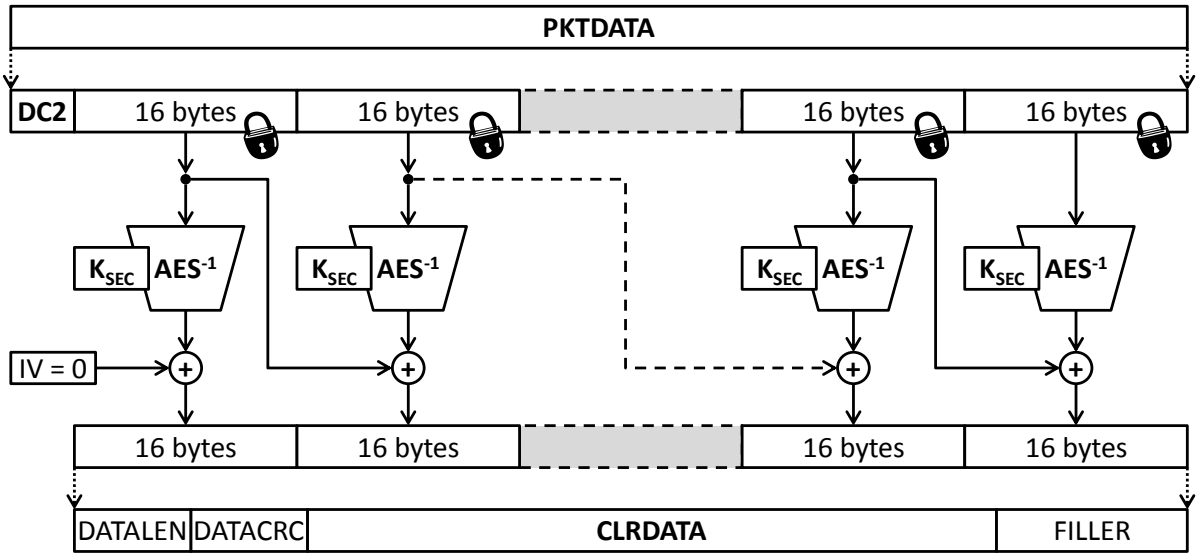
As described in the Link Level (**section 2.2.1**), if **PKTDATA** is encrypted, it must be started with the **«DC2»** byte.

➲ Example:

Considering **K_{SEC}** = DB3B4D015432AB3223555A1F81759A94, the SPE wishes to send the "**GIX**" command below in "Secure Communication":		
CLRDATA	47 49 58 30 31 34 00 01 00 0A 80 01 80 04 80 34 91 01 91 0E	GIX014....€.€.€4 ,.,.
Including the control fields (**DATALEN**, **DATACRC** and **FILLER**), the block to be encrypted is:		
DATALEN **DATACRC** **CLRDATA** **FILLER**	00 14 8D F2 47 49 58 30 31 34 00 01 00 0A 80 01 80 04 80 34 91 01 91 0E 00 00 00 00 00 00 00 00	..•òGIX014....€. €.€4'.'.....
The following result is obtained applying AES (CBC) with the **K_{SEC}** key (preceded by the **«DC2»** byte):		
PKTDATA	12 EA 22 9E DD 36 F8 4C 2A A7 E0 02 75 10 5C 3A 8A 78 7F C9 B2 88 35 40 AE E8 27 BA 1C 5A 03 94 96	.ê"žÝ6øL*§à.u.\: Šx.É²ˆ5@®è'º.Z.” –

5.2.2.2. Encrypted Packet Reception

Upon receiving an encrypted packet (detected by the presence of the **«DC2»** byte at the beginning of **PKTDATA**), the SPE or the pinpad must decrypt it using **AES^{-1}** in CBC mode using the **K_{SEC}** key:



Upon receiving an encrypted packet, the following verifications must be carried out:

- The size of **PKTDATA** (excluding the **«DC2»** byte) must be a multiple of 16;
- The value of **DATALEN** must be consistent (smaller than **PKTDATA**, excluding 5 bytes of **DATALEN**, **DATACRC** and **«DC2»**); and
- The CRC-16 calculated over **CLRDATA** must be equal to the value informed in **DATACRC**.

⚠ If the SPE detects any of these inconsistencies in an encrypted response, it must end the operation with a fatal error.

➲ Example:

SPE receives the following response started with **«DC2»**, indicating “Secure Communication”.

PKTDATA

```
12 BA 90 C3 82 65 12 69 B2 2D 0E FC 90 B9 2B C3   .º□Ã‚e.i²-.ü.¹+Ã
08 83 71 38 6A 69 B9 A7 A8 5B C6 AC 76 E4 84 37   .fq8ji¹§¨[Æ¬vä„7
BC 73 A2 02 86 EC B6 73 A4 93 4C 85 35 4E 47 16   ¼s¢.†ì¶s¤“L…5NG.
0F 27 2E 1A 2B 53 BA C1 B7 95 85 9E 4C 62 2F C8   .'..+SºÁ·•…žLb/È
66 1A 4B AE 1F EE 45 09 75 B7 CA 04 20 C6 18 A1   f.K®.îE.u·Ê.•Æ.¡
FC 74 47 65 C3 E7 08 AF 56 02 25 6B 75 A9 07 C3   ütGeÃç.¯V.%ku©.Ã
F9 A2 56 89 CB 11 23 9C 01 E3 6F C6 18 B4 17 A0   ù¢V‰Ë.#œ.ãoÆ.´.
2A 21 77 E3 C3 C8 73 B1 F0 6E 3B D6 20 8F F2 B4   *!wãÃÈs±ðn;Ö•ò´
96 A2 B0 BD F8 12 32 FD A0 97 30 0C 7D 19 B0 07   •¢°½ø.2ý •0.}.°.
DD C1 7E 6D EF 8B E7 BB 0E 82 58 8C 07 11 C0 1B   ÝÁ~mï‹ç».‚XŒ..À.
39 B1 21 BB 8C 66 E3 E0 31 3C 82 69 27 FB 7F 13   9±!»Œfãà1<‚i'û•.
36                                                6
```

SPE decrypts the message (without the **«DC2»**) using AES (CBC) with
K_{SEC} = DB3B4D015432AB3223555A1F81759A94, obtaining:

DATALEN **DATACRC** **CLRDATA** **FILLER**	00 A0 66 EB 47 49 58 30 30 30 31 35 31 80 01 00 0C 39 39 31 32 37 34 33 36 36 31 35 35 80 04 00 0D 48 45 4D 49 53 50 48 45 52 45 53 20 20 80 34 00 64 30 31 31 31 30 30 31 31 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 91 01 00 0A FF FF F9 13 25 00 43 20 04 43 00 00 00 00 00 00 00 00 00 00 00 00	. fëGIX000151€.. .991274366155€.. .HEMISPHERES••€4 .d01110011000000 0000000000022222 2222222222222222 2222222222222222 2222222222222222 2222222222222222 222222'...ÿÿù.%. C .C.....

SPE identifies the length **DATALEN** = 00A0h (160 bytes) and verifies **DATACRC** = 66EBh, extracting the **CLRDATA** block, response to a "**GIX**" command.

CLRDATA	47 49 58 30 30 30 31 35 31 80 01 00 0C 39 39 31 32 37 34 33 36 36 31 35 35 80 04 00 0D 48 45 4D 49 53 50 48 45 52 45 53 20 20 80 34 00 64 30 31 31 31 30 30 31 31 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 91 01 00 0A FF FF F9 13 25 00 43 20 04 43	GIX000151€...991 274366155€...HEM ISPHERES••€4.d01 1100110000000000 0000000222222222 2222222222222222 2222222222222222 2222222222222222 2222222222222222 22'...ÿÿù.%.C .C

5.2.2.3. Ending

The "Secure Communication" process is ended and the **K_{SEC}** key is cleared from memory in the following cases:

- A "**CLO**"/"**CLX**" command is received.
- The pinpad detects any inconsistencies in the encrypted command, returning "**ERR009**" (↳ST_ERRPKTSEC, as described in **section 2.3.4**).
- The pinpad receives an encrypted "**OPN**" command.

5.3. Encrypted PAN

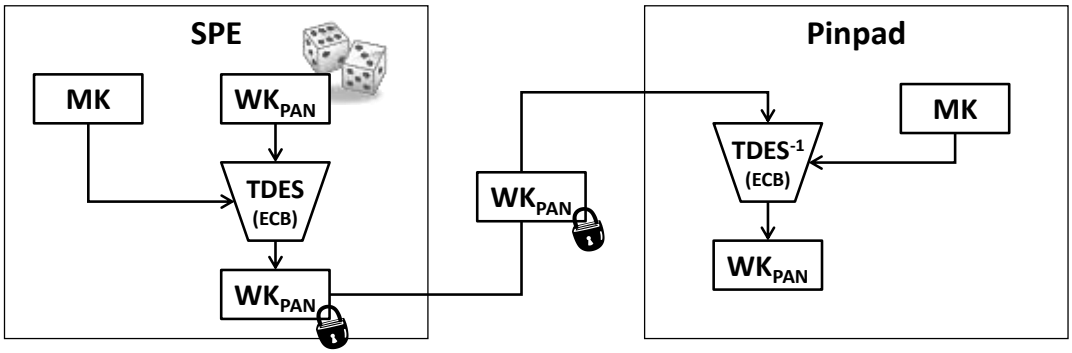
To prevent sensitive data (such as the card number - PAN) from traveling freely through the serial interface, this specification implements a working modality called "**Encrypted PAN**".

⚠ This modality is **obsolete**, having been replaced by the "Secure Communication" described in **section 5.2**. The SPE should use this mode only if the pinpad is not recognized as an "Abecs Pinpad".

In this mode, some data is encrypted using a TDES key called **WK_{PAN}**, which can be generated in two ways:

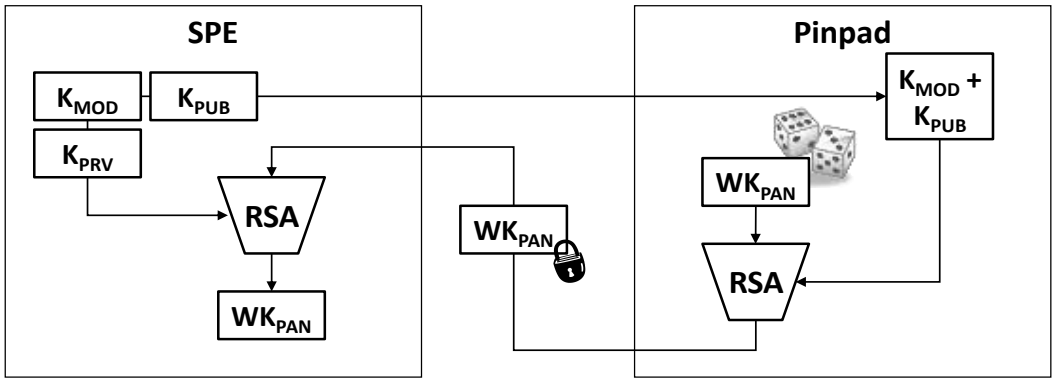
➲ Mode 1:

If a pinpad Master Key (TDES) is known, **WK_{PAN}** can be generated externally by the SPE and sent to the pinpad encrypted by this Master Key.



➲ Mode 2:

A random **WK_{PAN}** can be generated by the pinpad and returned to the SPE through an RSA cryptogram, as described in **section 5.3.3**.



The activation of the "Encrypted PAN" mode, as well as the definition of the **WK_{PAN}** key, is done through the command "**DWK**" (see **section 3.2.5**). The following table lists the commands and data affected by this mode:

Command	Affected data	Algorithm	Remarks
"**CKE**"	**CKE_TRK1** **CKE_TRK2** **CKE_TRK3**	TDES	The pinpad may return the PAN encrypted inside the card tracks as defined in **section 5.3.1**.
"**GCR**"	**GCR_PAN** **GCR_TRK1** **GCR_TRK2** **GCR_TRK3**	TDES	The pinpad may return the PAN (and the PAN inside the card tracks) encrypted as defined in **section 5.3.1**.
"**ENB**"	**ENB_INPUT**	$TDES^{-1}$	The SPE shall always encrypt the input data before sending it to the pinpad (although it is not necessarily a PAN, this command input data usually represents sensitive information).
"**GPN**"	**GPN_PAN**	$TDES^{-1}$	The PAN provided to the pinpad shall be encrypted by the SPE if it has 16 or more digits. The PAN may (or may not) be encrypted by the SPE if it is less than 16 digits long.

5.3.1. PAN Encoding

The encoding of the card number must respect the following rules:

- Only the least significant 16 digits of the PAN are encrypted, considering that they make up an 8-byte block in BCD encoding. Since the command parameters are in ASCII, the PAN's numeric decimal digits can be replaced directly by the hexadecimal digits of the generated cryptogram.
- Blank spaces in the middle of the card number (typically on Track 1 of some issuers) shall be converted to 'E' hexadecimal digit.
- The following rule shall be used to identify the PAN within the tracks (either 1, 2 or 3):
  - ⇒ From left to right, find the first numeric character ('0' to '9') or blank. It marks the beginning of the PAN.
  - ⇒ Continue examining the track to locate the separator character ("^" or "=") or until you reach the end.
- The PAN obtained will not be encrypted in the following cases:
  - ⇒ If it has less than 13 digits.
  - ⇒ If it contains any non-numeric character ('0' to '9') other than blank space.
- If the PAN has less than 16 digits, it will be padded with 'F' hexadecimal digits to the right, until this size is complete.
- The PAN or track length information contained in the input and output parameters of the commands must respect the length of the final exchanged information, including encryption. The entity receiving the encrypted data, whether SPE or pinpad, shall eliminate any 'F's at the end of the decoded PAN and recalculate its actual length.

⚠ These rules do not apply to the "**ENB**" command even if **ENB_INPUT** contains PAN information, since it is intended to encrypt generic data that is not interpreted in any way by the pinpad.

➲ Examples

The following examples consider **WK_{PAN}** = '**EA 52 8A 43 B0 26 52 FD EB 53 8B 42 B1 27 53 FC**':

Example 1: Track 1 returned by the pinpad, with PAN containing blank spaces.

- Cleartext (59 characters):

  "B3764 361234 56006^NAME NAME NAME NAME NAME N^0905060640431"

- Identified PAN (17 characters):

  "3764 361234 56006"

- Encoding:

  "764E361234E56006" ⇒ TDES ⇒ "5716A983F0E4643B"

- Encrypted (59 characters):

  "B35716A983F0E4643B^NAME NAME NAME NAME NAME N^0905060640431"

Example 2: A 19-digit PAN sent by the SPE to the pinpad.

- Cleartext (19 characters):

  "6234987432874320001"

- Encoding:

  "4987432874320001" ⇒ $TDES^{-1}$ ⇒ "407E5D4F32598B98"

- Encrypted (19 characters):

  "623407E5D4F32598B98"

Example 3: Track 1 returned by the pinpad, containing a 13-digit PAN.

- Cleartext (55 characters):

  "B3764361234006^NAME NAME NAME NAME NAME N^0905060640431"

- Identified PAN (13 characters):

  "3764361234006"

- Encoding:

  "3764361234006FFF" ⇒ TDES ⇒ "A4F4729D58CAA7DA"

- Encrypted (58 characters):

  "BA4F4729D58CAA7DA^NAME NAME NAME NAME NAME N^0905060640431"

Example 4: A 15-digit PAN sent by the SPE to the pinpad.

- Cleartext (15 characters):

  "376436123456006"

- Encoding:

  "376436123456006F" ⇒ TDES^{-1} ⇒ "431E6D386E688B0B"

- Encrypted (16 characters):

  "431E6D386E688B0B"

Example 5: Track 2 returned by the pinpad, containing a 16-digit PAN.

- Cleartext (37 characters):

  "6002938264523821=09050606404312376450"

- Identified PAN (16 characters):

  "6002938264523821"

- Encoding:

  "6002938264523821" ⇒ TDES ⇒ "BC27B145C5DE8BEB"

- Encrypted (37 characters):

  "BC27B145C5DE8BEB=09050606404312376450"

Example 6: A 37-character Track 2 returned by the pinpad, containing a 13-digit PAN, resulting in 40 characters after encryption.

- Cleartext (37 characters):

  "3827418937101=090506064043123764501 23"

- Identified PAN (13 characters):

  "3827418937101"

- Encoding:

  "3827418937101FFF" ⇒ TDES ⇒ "1CCE9197C5C6E3FF"

- Encrypted (**40 characters!!!**):

  "1CCE9197C5C6E3FF=090506064043123764501 23"

Example 7: Track 3 returned by the pinpad, containing a 19-digit PAN.

- Cleartext (104 characters):

  "4916748362525378000==5300053205322056019300000010000004050=00000000000000000=00000000000000000=7=3012056"

- Identified PAN (19 characters):

  "4916748362525378000"

- Encoding:

  "6748362525378000" ⇒ TDES ⇒ "FE8E271A114C1A35"

- Encrypted (104 characters):

  "491FE8E271A114C1A35==530005320532205601930000010000004050=00000000000000000=0000000000000000=7=3012056"

Example 8: Track 2 returned by the pinpad, without separator. In this case, to maintain consistency with the defined rule, it is as if the entire track was the PAN.

- Cleartext (37 characters):

  "98237465892736489562394865879234978 51"

- Identified PAN (37 characters):

  "98237465892736489562394865879234978 51"

- Encoding:

  "9486587923497851" ⇒ TDES ⇒ "2C05DF894573C7FA"

- Encrypted (37 characters):

  "98237465892736489562 32C05DF894573C7FA"

5.3.2. Track Decoding on the SPE

Even though the "Encrypted PAN" mode is enabled, in some situations (as explained in **section 5.3.1**) one or more tracks returned by the pinpad may not be encrypted, in the event that it was not possible to isolate a valid PAN. However, this specification does not provide a way to inform the SPE of this occurrence, which can generate errors when it tries to decrypt a received track.

This section seeks to define a standardized rule so that the SPE can identify whether the track actually contains an encrypted PAN:

- Scan the track from left to right until you find a separator ("^" or "=") or until you reach the end. The rightmost block should be considered as an encrypted PAN
  - ⇒ If the block found is less than 16 characters long, then there was no encryption.
  - ⇒ If the block found has any character outside the hexadecimal range ('0' to '9' / 'A' to 'F'), then there was no encryption.
- Decrypt the block using the **WK_{PAN}** key. Only numeric characters ('0' to '9'), blanks (encoded as 'E') or trailer characters ('F', 'FF' or 'FFF') are accepted in this result. If it does not show this consistency, it is deduced that there was no encryption.

5.3.3. RSA Cryptogram

When "Mode 2" is required in the "**DWK**" command, the pinpad will return an RSA cryptogram generated using the provided public key, with the following 128-byte cleartext layout:

Format	Description
A1	Block header (fixed = “T” / 54h).
N1	Layout version (fixed = “1” / 31h).
N9	Sequential number generated by the pinpad for cryptogram diversification.
H32	Random **WK_{PAN}** generated by the pinpad.
N84	Not used (zeros = “00000...0000”).
A1	Block trailer (fixed = “X”)

When opening the cryptogram, the SPE shall verify that the header, version and trailer are correct, validating their integrity. The sequential number must be disregarded.

5.4. “End-to-End” Cryptography

End-to-End Encryption is a feature in which the SPE never obtains complete card tracks (unless absolutely necessary), working only with the minimum information necessary for the local processing of the transaction.

This process is based on the following principles:

- The **“CEX”** and **“GCX”** commands never return the complete tracks of the cards;
- The “**GTK**” is able to return tracks already encrypted using a method and a key defined by the Acquirer Network, so they travel safely in the authorization message (optionally, a random $\mathbf{K_{RAND}}$ key generated by the pinpad may be used instead); and
- The “**GPN**” command does not need to receive the PAN when it is previously obtained from a card and, therefore, is already known to the pinpad (as long as the “**GTK**” command has not yet been used).

5.4.1. Incomplete Tracks and Masking

The **PP_TRK1INC**, **PP_TRK2INC** and **PP_TRK3INC** fields returned by the pinpad contain truncated card tracks so that only the following information necessary for the SPE local processing is disclosed:

- PAN (card number), which can be masked according to parameter **SPE_PANMASK**;
- Cardholder name (if Track 1);
- Card expiration date; and
- Service Code.

For this, the pinpad respects the following rules when assembling the fields:

PP_TRK1INC	Go through Track 1 from left to right and truncate in seven positions after the second separator “^” (5Eh). If this rule is not possible, consider the leftmost 19 positions.
PP_TRK2INC	Go through Track 2 from left to right and truncate in seven positions after the second separator “=” (3Dh). If this rule is not possible, consider the leftmost 19 positions.
PP_TRK3INC	Consider always the leftmost 19 positions.

If the **SPE_PANMASK** parameter is present in the command, the pinpad will mask the PAN as follows:

- Identify as PAN the first consecutive sequence of numeric characters to the left of the field, ignoring any blank spaces.
- Follow the **SPE_PANMASK** definition which indicates how many numeric digits must be kept open on the right (“RR”) and on the left (“LL”).
  - ⇒ If the sum of the sizes “RR” and “LL” exceeds the number of numeric digits in the PAN, there is no masking.
  - ⇒ The remaining numeric digits are replaced by asterisks (2Ah).

⇒ Blank spaces in the PAN are not considered in this count.

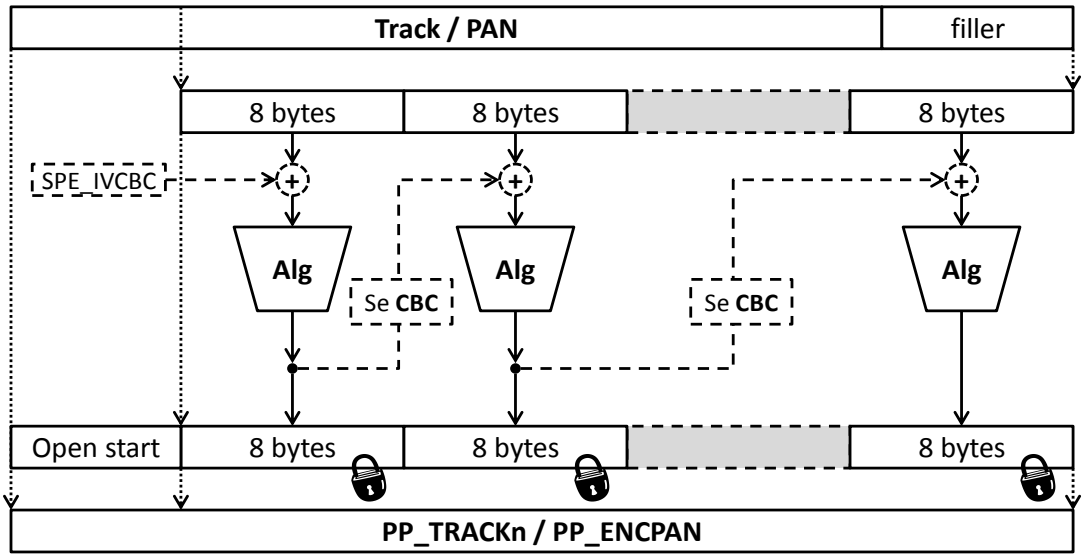
Besides the card tracks, this masking also affects the **PP_PAN** return field in the "**GCX**" command.

➲ Examples

- Assuming a Track 2 containing "**66733246732413=151260123487953427 5432**", the value of **PP_TRK2INC** would be "**66733246732413=1512601**".
- Assuming a Track 1 containing "**B9994444333322221111^NOME^1512601234879**", the value of **PP_TRK1INC** would be "**B9994444333322221111^NOME^1512601**".
- Assuming a Track 2 containing "**667332467324131512601234879534275432**", the value of **PP_TRK2INC** would be "**6673324673241315126**".
- Assuming a Track 1 containing "**B3764 329710 01006^JOE^2108100265123756**" and **SPE_PANMASK** = "0604", the value of **PP_TRK1INC** would be "**B3764 32**** *1006^JOE^2108100**".
- Assuming a Track 2 containing "**4444333322221111=2212601019923625524**" and **SPE_PANMASK** = "0700", the value of **PP_TRK2INC** would be "**4444333*********=2212601**".
- Assuming a Track 1 containing "**A756325325535^PROPRIETARYFORMAT=6562532**" and **SPE_PANMASK** = "0005", the value of **PP_TRK1INC** would be "**A*******25535^PROPR**".

5.4.2. Track Cryptography

Whenever the SPE requests encrypted tracks in the "**GTK**" command, the pinpad encode them according to the diagram below:



The algorithm to be used for encryption ("Alg") is selected in **SPE_MTHDDAT**, using the **SPE_KEYIDX** key. However, when **SPE_MTHDDAT = "9x"**, the following rule must be adopted:

- Encryption will be done using a random **TDES** key (K_{RAND}) generated by the pinpad itself. This key must be generated every time "**GTK**" is executed and it cannot be reused.
- The SPE must provide an RSA public key in the **SPE_PBKMOD** and **SPE_PBKEXP** input fields.

- The K_{RAND} key is encrypted by the pinpad using the RSA public key, in the same PKCS #1 format presented in **section 5.2.1**, generating the **PP_ENCKRAND** output field.

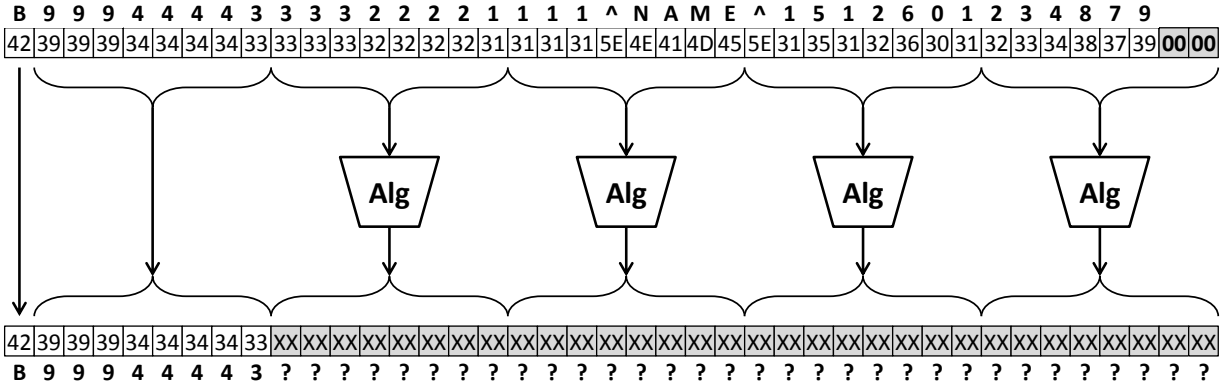
5.4.2.1. Track 1

The Track 1 allows alphanumeric characters, so it is always treated as information in ASCII encoding, with each symbol occupying one byte. Thus, the following rule is adopted:

- The pinpad preserves in cleartext the initial characters of Track 1 according to the quantity requested in **SPE_OPNDIG**, disregarding the format character (usually "B")
- The block to be encrypted must have a size multiple of 8 (eight) bytes. If necessary, it must be filled trailing 00h bytes.

➲ Example

The following diagram illustrates the encryption of a 39-character track ("**B9994444333322221111^ NAME^1512601234879**") using ECB block mode, preserving the first 8 characters in cleartext:



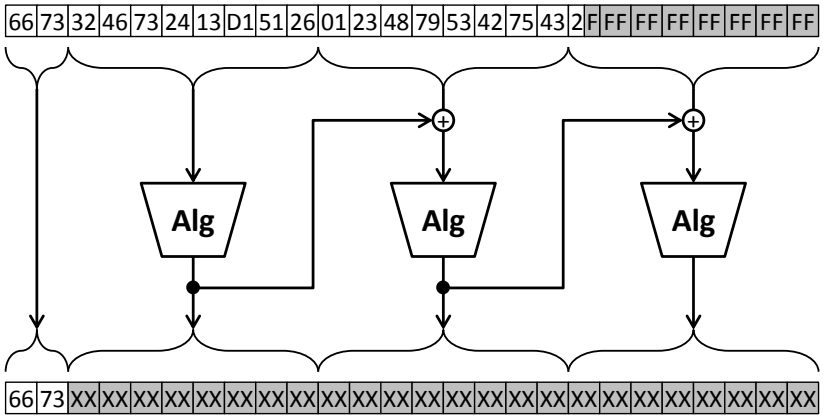
5.4.2.2. PAN and Tracks 2/3

PAN, Track 2 and Track 3 follow the same coding, in which each symbol occupies a nibble (half byte). Thus, the following rule is adopted:

- The pinpad preserves in cleartext the initial digits according to the quantity requested in **SPE_OPNDIG**, considering that each byte represents two digits.
- The block to be encrypted must have a size multiple of 8 (eight) bytes. If necessary, it must be filled with trailing Fh nibbles.

➲ Example

The following diagram illustrates the encryption of a 37-position track ("**66733246732413= 151260123487953427 5432**") using CBC block mode, without "IV "(Initialization Vector), preserving the first 4 digits in cleartext:


66 73 32 46 73 24 13 D1 51 26 01 23 48 79 53 42 75 43 2F FF FF FF FF FF FF FF
Alg
Alg
Alg
66 73 XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX

6. Pinpad internal operation

This chapter defines the rules for the internal operation of the pinpads and is intended for its manufacturers and developers, and its knowledge is not necessary for SPE suppliers.

For more information, see the Portuguese version of this specification.

7. Additional information

The sections in this chapter provide supplementary information useful for understanding this specification.

7.1. TLV Encoding

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

- The “tag” field (T) consists of one or more consecutive bytes.
- The “length” field (L) consists of one or more consecutive bytes. It indicates the size of the next field.
- The “value” field (V) indicates the value of the data object. If L = 00h, the “value” field is not present.

The following sub-items define the coding for these fields.

7.1.1. Tag (T) Field Encoding

The following table describes the first byte of the “tag” field of a BER-TLV object:

<table>
<tr><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>Meaning</th></tr>
<tr><td>x</td><td>x</td><td>x</td><td colspan="5"></td><td>Object class and type</td></tr>
<tr><td colspan="3" rowspan="2"></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>See subsequent bytes</td></tr>
<tr><td colspan="5">Any other value < 31</td><td>Tag number</td></tr>
</table>

According to ISO/IEC 8825, the following table 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'):

<table>
<tr><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>Meaning</th></tr>
<tr><td>1</td><td colspan="7" rowspan="2"></td><td>Another byte follows</td></tr>
<tr><td>0</td><td>Last tag byte</td></tr>
<tr><td></td><td colspan="7">Any value > 0</td><td>(Part of) tag number</td></tr>
</table>

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

7.1.2. Length (L) Field Encoding

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 to 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 to 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.

7.2. CRC Calculation

Whenever this specification refers to CRC calculation, it refers to the **CRC-16-CCITT**, with a generator polynomial $\mathbf{x^{16} + x^{12} + x^5 + x^0}$.

The following C language code illustrates this implementation:

```
#define CRC_MASK 0x1021          /* x^16 + x^12 + x^5 + x^0 */

UINT16 CRC_Calc (unsigned char *pbData, int iLength)
{
   UINT16 wData, wCRC = 0;
   int i;

   for ( ;iLength > 0; iLength--, pbData++) {
      wData = (UINT16) (((UINT16) *pbData) << 8);
      for (i = 0; i < 8; i++, wData <<= 1) {
         if ((wCRC ^ wData) & 0x8000)
            wCRC = (UINT16) ((wCRC << 1) ^ CRC_MASK);
         else
            wCRC <<= 1;
      }
   }
   return wCRC;
}
```

7.3. Pinpad Display

7.3.1. Use by the commands

The commands specified in this document may or may not use the pinpad display to show messages, depending on the situation. The following table lists all commands that can use the display and how.

Command	Display use
"**OPN**"	Display is erased and backlight is activated.
"**CLO**"	Backlight is deactivated and **CLO_MSG** message is left on the display.
"**CLX**"	Backlight is deactivated and **SPE_DSPMSG** message is left on the display (or **SPE_MFNAME** media file is presented).
"**CHP**"	The display is only used if there is a PIN capture (**CHP_OPER** = "3"), being erased at the end, whether the capture is successful or unsuccessful. In the other modes of the command (**CHP_OPER** ≠ "3") the display must not be modified or deleted.
"**DEX**"	The **DEX_MSG** message is left on the display.
"**DSP**"	The **DSP_MSG** message is left on the display.
"**GCD**"	The display is used in the data capture process and is always erased at the end, whether the capture is successful or unsuccessful.
"**GPN**"	The display is used in the PIN capture process and is always erased at the end, whether the capture is successful or unsuccessful.
"**MNU**"	The display is used to show the menu and is always cleared at the end, whether the selection is successful or unsuccessful.
"**RMC**"	The **RMC_MSG** message is left on the display.
"**DSI**"	The image indicated by **SPE_MFNAME** is left on the display.
"**TLR**"	Optionally, the pinpad may leave an informational message on the display indicating the table loading in progress.
"**TLE**"	Erases the display only if it has been modified in "**TLR**", otherwise does not change its contents.
"**GCR**"	Uses the display to request a card and to display the application selection menu. ▪ For ICC or CTLS successfully processed, leave a message on the display indicating the selected application. ▪ In case of error, erase the display at the end.
"**GOC**"	If required, it uses the display to capture the PIN, erasing it at the end, whether the capture is successful or unsuccessful. If there is no PIN capture, the display is not modified.

"**GCX**"	Uses the display to request a card and to display the application selection menu. ▪ For ICC or CTLS successfully processed, leave a message on the display indicating the selected application. ▪ In case of error, erase the display at the end.
"**GOX**"	If required, it uses the display to capture the PIN, erasing it at the end, whether the capture is successful or unsuccessful. If there is no PIN capture, the display is not modified.
"**FCX**"	It may use the display to request a card in the case of CTLS with *Issuer Script Processing*, erasing it at the end. For other situations, the display is not modified.

⚠ Other commands not listed in this table shall not erase or modify the contents of the display.

7.3.2. Character Table

For the presentation of display messages on the pinpad, this specification uses the ISO/IEC 8859-1 codepage, whose main symbols are defined in the following table:

032(20h)		033(21h)	!	034(22h)	"	035(23h)	#	036(24h)	$
037(25h)	%	038(26h)	&	039(27h)	'	040(28h)	(	041(29h)	)
042(2Ah)	*	043(2Bh)	+	044(2Ch)	,	045(2Dh)	-	046(2Eh)	.
047(2Fh)	/	048(30h)	0	049(31h)	1	050(32h)	2	051(33h)	3
052(34h)	4	053(35h)	5	054(36h)	6	055(37h)	7	056(38h)	8
057(39h)	9	058(3Ah)	:	059(3Bh)	;	060(3Ch)	<	061(3Dh)	=
062(3Eh)	>	063(3Fh)	?	064(40h)	@	065(41h)	A	066(42h)	B
067(43h)	C	068(44h)	D	069(45h)	E	070(46h)	F	071(47h)	G
072(48h)	H	073(49h)	I	074(4Ah)	J	075(4Bh)	K	076(4Ch)	L
077(4Dh)	M	078(4Eh)	N	079(4Fh)	O	080(50h)	P	081(51h)	Q
082(52h)	R	083(53h)	S	084(54h)	T	085(55h)	U	086(56h)	V
087(57h)	W	088(58h)	X	089(59h)	Y	090(5Ah)	Z	091(5Bh)	[
092(5Ch)	\	093(5Dh)	]	094(5Eh)	^	095(5Fh)	_	096(60h)	`
097(61h)	a	098(62h)	b	099(63h)	c	100(64h)	d	101(65h)	e
102(66h)	f	103(67h)	g	104(68h)	h	105(69h)	i	106(6Ah)	j
107(6Bh)	k	108(6Ch)	l	109(6Dh)	m	110(6Eh)	n	111(6Fh)	o
112(70h)	p	113(71h)	q	114(72h)	r	115(73h)	s	116(74h)	t
117(75h)	u	118(76h)	v	119(77h)	w	120(78h)	x	121(79h)	y
122(7Ah)	z	123(7Bh)	{	124(7Ch)	|	125(7Dh)	}	126(7Eh)	~

192(C0h)	À	193(C1h)	Á	194(C2h)	Â	195(C3h)	Ã	199(C7h)	Ç
200(C8h)	È	201(C9h)	É	202(CAh)	Ê	205(CDh)	Í	209(D1h)	Ñ
211(D3h)	Ó	212(D4h)	Ô	213(D5h)	Õ	218(DAh)	Ú	220(DCh)	Ü
224(E0h)	à	225(E1h)	á	226(E2h)	â	227(E3h)	ã	231(E7h)	ç
232(E8h)	è	233(E9h)	é	234(EAh)	ê	237(EDh)	í	241(F1h)	ñ
243(F3h)	ó	244(F4h)	ô	245(F5h)	õ	250(FAh)	ú	252(FCh)	ü

⚠ If the pinpad does not support this codepage, it must “translate” the messages before the presentation on the display, to remove accents and cedilla.

