ETSI TS 102 225 V16.0.1 (2020-12)

Smart Cards;
Secured packet structure for UICC based applications
(Release 16)

Reference
RTS/SCP-T0284VG01

Keywords
security, smart card

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Smart Card Platform (SCP).

It is based on work originally done in the 3GPP in TSG-terminals WG3 and ETSI SMG.

The contents of the present document are subject to continuing work within TC SCP and may change following formal TC SCP approval. If TC SCP modifies the contents of the present document, it will then be republished by ETSI with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

- 0 early working draft;
- 1 presented to TC SCP for information;
- 2 presented to TC SCP for approval;
- 3 or greater indicates TC SCP approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document specifies the structure of Secured Packets for different transport and security mechanisms.

It is applicable to the exchange of secured packets between an entity in a network and an entity in the UICC.

Secured Packets contain application messages to which certain mechanisms according to ETSI TS 102 224 [1] have been applied. Application messages are commands or data exchanged between an application resident in or behind the network and on the UICC. The Sending/Receiving Entity in the network and the UICC are responsible for applying the security mechanisms to the application messages and thus turning them into Secured Packets.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

- In the case of a reference to a TC SCP document, a non-specific reference implicitly refers to the latest version of that document in the same Release as the present document.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 102 224: "Smart Cards; Security mechanisms for UICC based Applications - Functional requirements".

[2] Void.

[3] ISO 16609:2012: "Financial services - Requirements for message authentication using symmetric techniques".

[4] Void.

[5] ETSI TS 131 115: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Secured packet structure for (Universal) Subscriber Identity Module (U)SIM Toolkit applications (3GPP TS 31.115)".

[6] GlobalPlatform: "GlobalPlatform Technology Card Specification", Version 2.3.1.

NOTE: Available at https://globalplatform.org/specs-library/.

[7] Applied Cryptography: "Protocols, Algorithms, and Source Code in C", 2nd Edition, Bruce Schneier, John Wiley & Sons.

[8] ETSI TS 101 220: "Smart Cards; ETSI numbering system for telecommunication application providers".

[9] ETSI TS 102 226: "Smart Cards; Remote APDU structure for UICC based applications".

[10] ETSI TS 102 127: "Smart Cards; Transport protocol for CAT applications; Stage 2".

[11] ISO/IEC 13239:2002: "Information technology - Telecommunications and information exchange between systems - High-level data link control (HDLC) procedures".

[12] NIST Special Publication FIPS-197 (2001): "Advanced Encryption Standard (AES)".

NOTE: Available at http://csrc.nist.gov/publications/fips/index.html.

[13] NIST Special Publication 800-38A (2001): "Recommendation for Block Cipher Modes of Operation - Methods and Techniques".

NOTE: Available at http://csrc.nist.gov/publications/nistpubs/.

[14] NIST Special Publication 800-38B (2005): "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication".

NOTE: Available at http://csrc.nist.gov/publications/nistpubs/.

[15] GlobalPlatform: "GlobalPlatform Card UICC Configuration", Version 1.2.0.

[16] ETSI TS 102 484: "Smart Cards; Secure channel between a UICC and an end-point terminal".

[17] ETSI TS 102 483: "Smart cards; UICC-Terminal interface; Internet Protocol connectivity between UICC and terminal".

[18] ETSI TS 102 223: "Smart Cards; Card Application Toolkit (CAT)".

[19] GlobalPlatform: "Remote Application Management over HTTP, Card Specification v2.2 - Amendment B", Version 1.1.3.

NOTE: Available at https://globalplatform.org/specs-library/.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

- In the case of a reference to a TC SCP document, a non-specific reference implicitly refers to the latest version of that document in the same Release as the present document.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TR 102 216: "Smart Cards; Vocabulary for Smart Card Platform specifications".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI TR 102 216 [i.1] and the following apply:

NOTE: A term defined in the present document takes precedence over the definition of the same term, if any, in ETSI TR 102 216 [i.1].

Advanced Encryption Standard (AES): standard cryptographic algorithm specified in FIPS-197 [12]

application layer: layer above the Transport Layer on which the Application Messages are exchanged between the sending and receiving applications

application message: package of commands or data sent from the Sending Application to the Receiving Application, or vice versa, independently of the transport mechanism

NOTE: An Application Message is transformed with respect to a chosen Transport Layer and chosen level of security into one or more secured packets.

card manager: generic term for the 3 card management entities of a GlobalPlatform card i.e. the OPEN, Issuer Security Domain and the Cardholder Verification Method Services provider as defined in the GlobalPlatform Card Specification [6]

command header: security header of a command packet

NOTE: It includes all fields except the Secured Data.

command packet: secured packet transmitted by the Sending Entity to the Receiving Entity, containing a secured Application Message

counter: mechanism or data field used for keeping track of a message sequence

NOTE: This could be realized as a sequence oriented or time stamp derived value, maintaining a level of synchronization between the Sending Entity and the Receiving Entity.

cryptographic checksum: string of bits derived from some secret information, (e.g. a secret key), part or all of the Application Message, and possible further information (e.g. part of the Security Header)

NOTE: The secret key is known to the Sending Entity and to the Receiving Entity. The Cryptographic Checksum is often referred to as Message Authentication Code.

Data Encryption Key (DEK): key identifier for ciphering keys as defined in ETSI TS 102 226 [9]

Data Encryption Standard (DES): standard cryptographic algorithm specified as DEA in ISO 16609 [3]

digital signature: string of bits derived from some secret information, (e.g. a secret key), the complete Application Message, and possible further information (e.g. part of the Security Header)

NOTE: The secret information is known only to the Sending Entity. Although the authenticity of the Digital Signature can be proved by the Receiving Entity, the Receiving Entity is not able to reproduce the Digital Signature without knowledge of the secret information owned by the Sending Entity.

issuer security domain: on-card entity providing support for the control, security, and communication requirements of the Card Issuer as defined in the GlobalPlatform Card Specification [6]

receiving application: entity to which the Application Message is destined

receiving entity: entity where the Secured Packet is received (e.g. SMS-SC, UICC, USSD entry point, or dedicated Toolkit Server) and where the security mechanisms are utilized

NOTE: The Receiving Entity processes the Secured Packets.

redundancy check: string of bits derived from the Application Message and possible further information for the purpose of detecting accidental changes to the message, without the use of any secret information

response header: security header of a response packet

response packet: secured packet transmitted by the Receiving Entity to the Sending Entity, containing a secured response and possibly application data

secured data: this field contains the secured application message and possibly padding octets

secured packet: information flow on top of which the level of required security has been applied

NOTE: An Application Message is transformed with respect to a chosen Transport Layer and chosen level of security into one or more Secured Packets.

security domain: As defined in the GlobalPlatform Card Specification [6].

security header: that part of the secured packet which consists of all security information (e.g. counter, key identification, indication of security level, checksum or Digital Signature)

sender identification: simple verification of the identity of the Sending Entity by the Receiving Entity comparing the sender identity with an a priori stored identity of the sender at the Receiving Entity

sending application: entity generating an Application Message to be sent

sending entity: entity from which the Secured Packet originates (e.g. SMS-SC, UICC, USSD entry point, or dedicated Toolkit Server) and where the security mechanisms are invoked

NOTE: The Sending Entity generates the Secured Packets to be sent.

status code: indication that a message has been received (correctly or incorrectly, indicating reason for failure)

transport layer: layer responsible for transporting Secured Packets through the network

NOTE: The transport layer implements one or more transport mechanisms, (e.g. SMS or USSD).

unsecured acknowledgement: status code included in a response message

3.2 Symbols

Void.

3.3 Abbreviations

For the purpose of the present document, the following abbreviations apply:

AES	Advanced Encryption Standard
AID	Application IDentifier
API	Application Programming Interface
ARD	Additional Response Data
BER	Basic Encoding Rules
BIP	Bearer Independent Protocol
CAT_TP	Card Application Toolkit Transport Protocol
CBC	Cipher Block Chaining
CC	Cryptographic Checksum
CHI	Command Header Identifier
CHL	Command Header Length
CMAC	Cipher-based Message Authentication Code
CNTR	CouNTeR
CPI	Command Packet Identifier
CPL	Command Packet Length
CRC	Cyclic Redundancy Check
DEA	Data Encryption Algorithm
DEK	Data Encryption Key
DES	Data Encryption Standard
DS	Digital Signature
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IP	Internet Protocol
ISO	International Organization for Standardization
KIc	Key and algorithm Identifier for ciphering
KID	Key and algorithm IDentifier for RC/CC/DS
NIST	National Institute of Standards and Technology
PCNTR	Padding CouNTeR
PoR	Proof of Receipt
RAM	Remote Application Management
RC	Redundancy Check
RE	Receiving Entity
RFM	Remote File Management

RHI	Response Header Identifier
RHL	Response Header Length
RPI	Response Packet Identifier
RPL	Response Packet Length
RSC	Response Status Code
SE	Sending Entity
SMG	Special Mobile Group
SMS	Short Message Service
SMS-CB	Short Message Service - Cell Broadcast
SMS-SC	Short Message Service - Service Centre
SP	Special Publication
SPI	Security Parameters Indication
TAR	Toolkit Application Reference
TC	Technical Committee
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TLV	Tag/Length/Value (data structure)
USSD	Unstructured Supplementary Services Data

4 Overview of security system

4.0 Overview

An overview of the secure communication related to the Card Application Toolkit together with the required security mechanisms is given in ETSI TS 102 224 [1] (see figure 1). Standardized applications for remote management of the UICC, which make use of the secure communication defined in the present document, are specified in ETSI TS 102 226 [9].

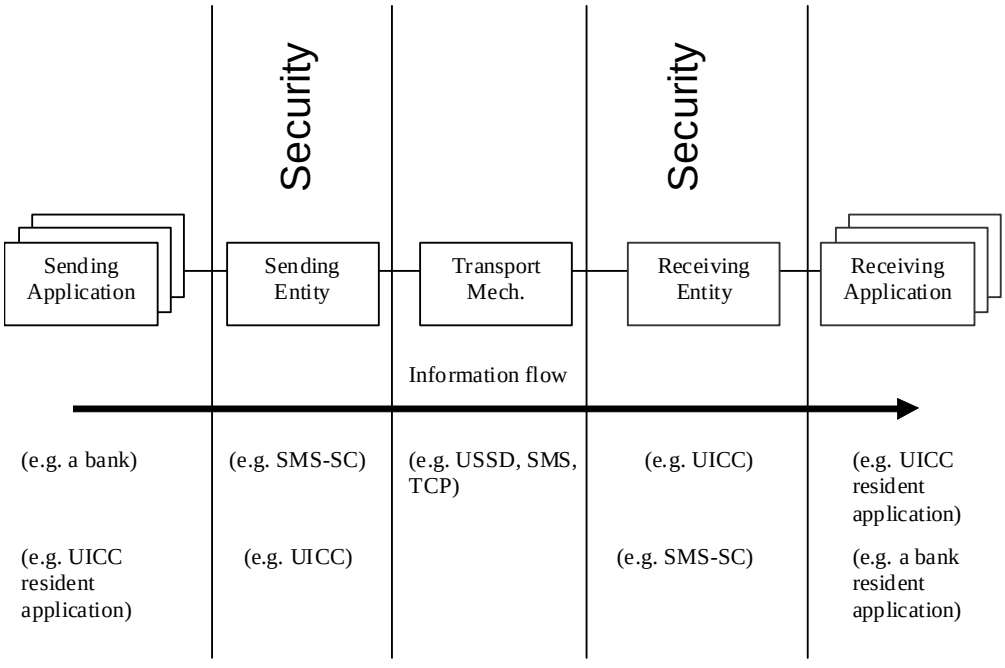


Figure 1: System overview

4.1 Protocol for generalized secured packets

This clause applies if messages are protected using an implementation of the generalized secured packet format.

The Sending Application prepares an Application Message and forwards it to the Sending Entity, with an indication of the security to be applied to the message.

The Sending Entity prepends a Security Header (the Command Header) to the Application Message. It then applies the requested security to part of the Command Header and all of the Application Message, including any padding octets. The resulting structure is here referred to as the (Secured) Command Packet.

Under normal circumstances the Receiving Entity receives the Command Packet and unpacks it according to the security parameters indicated in the Command Header. Additional security conditions may apply (e.g. a Minimum Security Level as defined in ETSI TS 102 226 [9]) before unpacking it. The Receiving Entity subsequently forwards the Application Message to the Receiving Application indicating to the Receiving Application the security that was applied. The interface between the Sending Application and Sending Entity and the interface between the Receiving Entity and Receiving Application are proprietary and therefore outside the scope of the present document.

If so indicated in the Command Header, the Receiving Entity shall create a (Secured) Response Packet. The Response Packet consists of a Security Header (the Response Header) and optionally, application specific data supplied by the Receiving Application. Both the Response Header and the application specific data are secured using the security mechanisms indicated in the received Command Packet. The Response Packet will be returned to the Sending Entity, subject to constraints in the transport layer (e.g. timing).

Although in some cases there might be no direct acknowledgement mechanism (i.e. for SMS-CB) the Sending Application may have requested a response. In this case a (Secured) Response Packet could be sent using a different bearer by the Receiving Application.

In some circumstances a security related error may be detected at the Receiving Entity. In such circumstances the Receiving Entity shall react according to the following rules:

1) nothing shall be forwarded to the Receiving Application. i.e. no part of the Application Message, and no indication of the error;

2) if the Sending Entity does not request a response (in the Command Header) the Receiving Entity discards the Command Packet and no further action is taken;

3) if the Sending Entity requests a response and the Receiving Entity can authenticate the Sending Entity, the Receiving Entity shall create a Response Packet indicating the error cause. This Response Packet shall be secured according to the security indicated in the received Command Packet;

4) if the Sending Entity requests a response and the Receiving Entity cannot authenticate the Sending Entity, the Receiving Entity shall:

   - either send a Response Packet indicating the error cause without any security being applied to the Response Packet and the Counter (CNTR) field set to zero; or

   - not send any Response Packet and discard the Command Packet with no further action being taken;

NOTE: The option to be adopted may depend on the bearer and the security policy of the UICC issuer.

5) if the Receiving Entity receives an unrecognizable Command Header (e.g. an inconsistency in the Command Header), the Command Packet shall be discarded and no further action taken.

4.2 Protocol for secured messages based on HTTPS

The security for data exchange over TCP is provided by TLS. The HTTP protocol is used on top of TLS to provide encapsulation of the data and information about the receiving entity.

The processing rules for messages that are protected using HTTPS are specified in Amendment B of the Global Platform Card Specification v2.2 [19].

TCP/IP transport is provided by the Bearer Independent Protocol of ETSI TS 102 223 [18] or a direct IP connection as specified in ETSI TS 102 483 [17].

If a TLS connection with the receiving entity is not already established, the sending entity shall send a triggering message as specified in Amendment B of the Global Platform Card Specification v2.2 [19] to the security domain handling the TLS connection for itself or for an associated application.

5 Generalized secured packet structure

5.0 Packet structure

Command and response packets have the same overall structure consisting of a variable length security header within a variable length shell. To model this, use is made of a double TLV -tag, length, value- structure.

5.1 Command packet structure

5.1.0 Overview

The Command Header precedes the Secured Data in the Command Packet, and is of variable length.

The Command Packet shall be structured according to table 1.

Table 1: Structure of the command packet

Element	Length	Comment
Command Packet Identifier (CPI)	1 octet	Identifies that this data block is the secured Command Packet.
Command Packet Length (CPL)	variable	This shall indicate the number of octets from and including the Command Header Identifier to the end of the Secured Data, including any padding octets required for ciphering.
Command Header Identifier (CHI)	1 octet	Identifies the Command Header.
Command Header Length (CHL)	variable	This shall indicate the number of octets from and including the SPI to the end of the RC/CC/DS.
Security Parameter Indicator (SPI)	2 octets	See detailed coding in clause 5.1.1.
Ciphering Key Identifier (KIc)	1 octet	Key and algorithm Identifier for ciphering.
Key Identifier (KID)	1 octet	Key and algorithm Identifier for RC/CC/DS.
Toolkit Application Reference (TAR)	3 octets	Coding is application dependent as defined in ETSI TS 101 220 [8].
Counter (CNTR)	5 octets	Replay detection and Sequence Integrity counter.
Padding Counter (PCNTR)	1 octet	This indicates the number of padding octets used for ciphering at the end of the secured data.
Redundancy Check (RC), Cryptographic Checksum (CC) or Digital Signature (DS)	variable	Length depends on the algorithm. A typical value is 8 octets if used, and for a DS could be 48 or more octets; the minimum should be 4 octets.
Secured data	variable	Contains the Secured Application Message and possibly padding octets used for ciphering.

Unless indicated otherwise, the CPL and the CHL shall be coded according to BER-TV's coding of length in ETSI TS 101 220 [8].

Table 2: Linear representation of command packet

CPI	CPL	CHI	CHL	SPI	KIc	KID	TAR	CNTR	PCNTR	RC/CC/DS	Secured data with padding
								Note 1	Note 1	Note 1	Note 1
Note 3	Note 3	Note 3	Note 3	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2		Note 2

NOTE 1: These fields are included in the data to be ciphered if ciphering is indicated in the Security Header.
NOTE 2: These fields are included in the calculation of the RC/CC/DS.
NOTE 3: Part or all of these fields may also be included in the calculation of the RC/CC/DS, depending on implementation (e.g. SMS).

If ciphering is indicated, first the RC/CC/DS shall be calculated as indicated in note 2 of table 2, and then ciphering shall be applied, as indicated in note 1 of table 2.

If the SPI indicates that a specific field is unused, the Sending Entity shall set the contents of this field to zero, and the Receiving Entity shall ignore the contents.

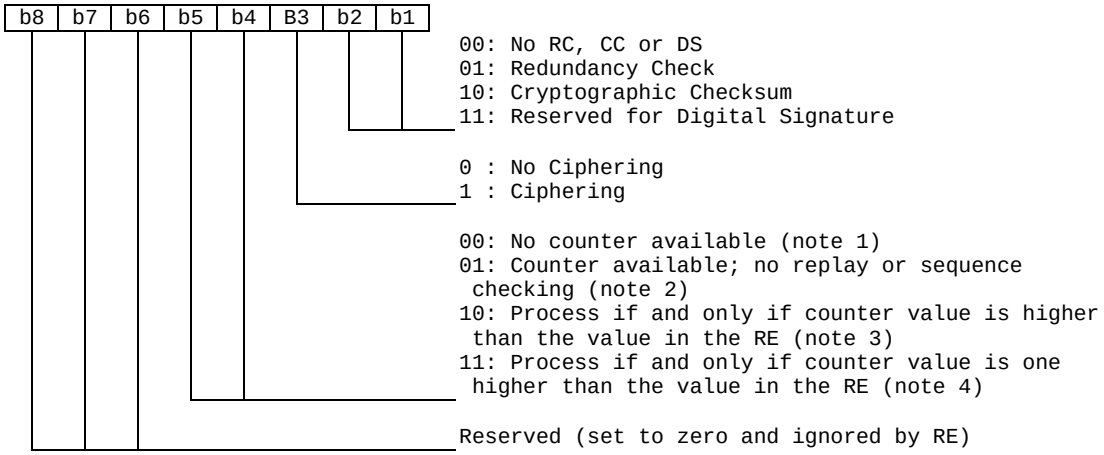
If the SPI indicates that no RC, CC or DS is present in the Command Header, the RC/CC/DS field shall be of zero length.

Padding octets may consist of any plaintext value. If the Padding Counter content is zero, this shall indicate no padding octets, or no padding is necessary.

5.1.1 Coding of the SPI

The SPI is coded as below.

First Octet:



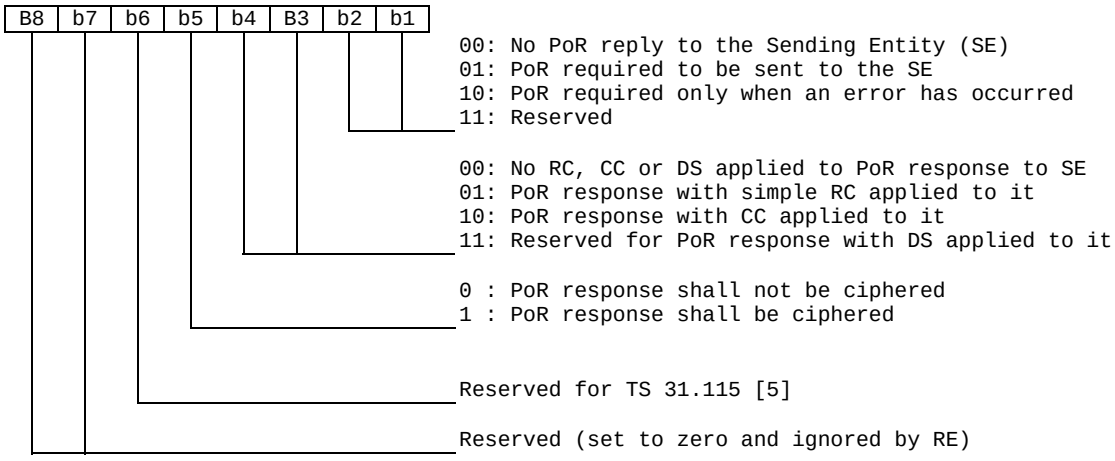
NOTE 1: In this case the counter field is present in the message.

NOTE 2: In this case the counter value is used for information purposes only, (e.g. date or time stamp). If the Command Packet was successfully unpacked, the counter value can be forwarded from the Receiving Entity to the Receiving Application. This depends on proprietary implementations and happens in an application dependent way.

NOTE 3: The counter value is compared with the counter value of the last received Command Packet. This is tolerant to failures on the transport level (i.e. losses of Command Packets). A possible scenario is a global update.

NOTE 4: This provides strict control in addition to security indicated in note 3.

Second Octet:



The following rules shall apply for the security settings of Response Packets:

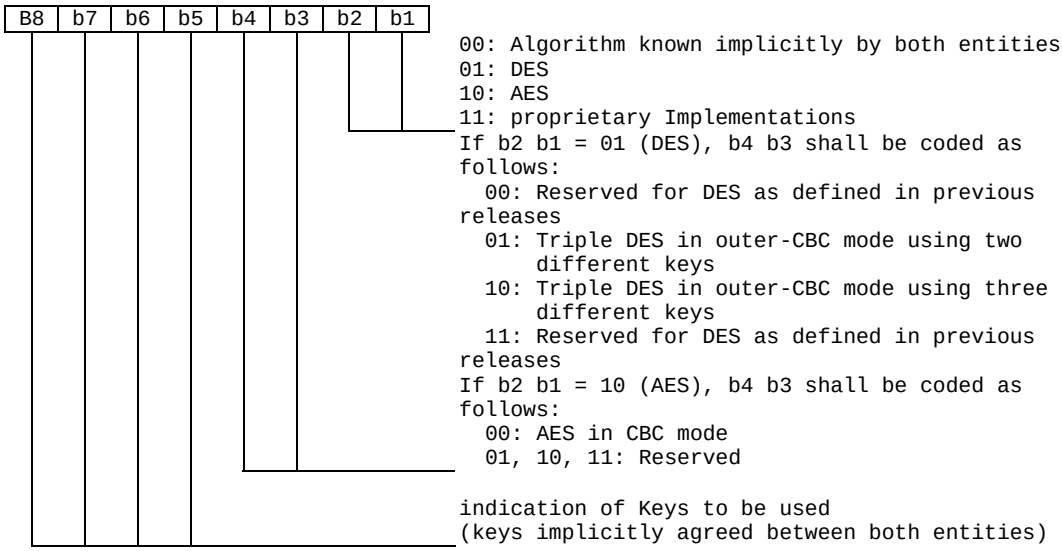
1) If SPI2.b4b3 is not set to 00, then the RC or CC requested for the Response Packet, i.e. SPI2.b4b3, shall be the same as the Command Packet, i.e. SPI1.b2b1.

2) Ciphering of the Response Packet, i.e. SPI2.b5 set to '1', is only allowed if the Command Packet has been successfully authenticated (e.g. using CC) and if ciphering was applied to the Command Packet, i.e. SPI1.b3 is set to '1'.

If any violation of these rules for the security settings of Response Packets is detected at the Receiving Entity, the response shall be processed as defined for a security error where the Receiving Entity cannot authenticate the Sending Entity (see clause 4.1, item 4)). In case a response is sent, the error cause shall be set to "Unidentified security error".

5.1.2 Coding of the KIc

The KIc is coded as below.



DES is the algorithm specified as DEA in ISO 16609 [3]. Triple DES in outer-CBC mode is described in clause 15.2 of [7].

The use of (single) DES is deprecated. However the coding is reserved for backwards compatibility with pre Release 8 versions of the present document.

AES is the block cipher algorithm specified in FIPS-197 [12]. AES in CBC mode is defined in NIST SP 800-38A [13]. The key length shall be known implicitly by both entities and shall be 128, 192 or 256 bits.

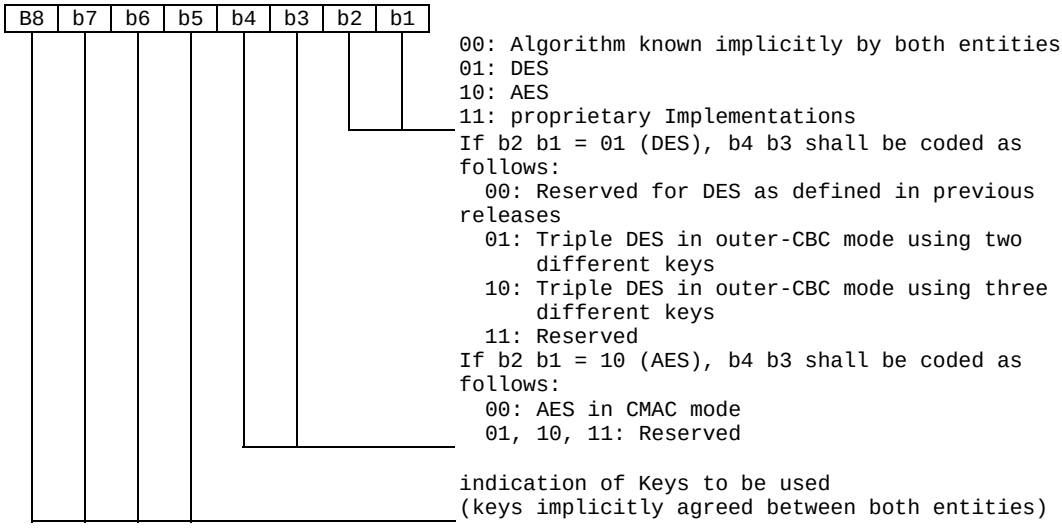
The initial chaining value for CBC modes shall be zero. AES shall be used together with counter settings (b5 and b4 of the first octet of SPI) 10 or 11.

For GlobalPlatform security architecture compliant cards see annex A.

5.1.3 Coding of the KID

5.1.3.1 Coding of the KID for Cryptographic Checksum

If b2b1= '10' (Cryptographic Checksum) in the first byte of SPI, KID shall be coded as following:



DES is the algorithm specified as DEA in ISO 16609 [3]. Triple DES in outer-CBC mode is described in clause 15.2 of [7].

In case of DES, the initial chaining value for CBC modes shall be zero. If padding is required, the padding octets shall be coded hexadecimal '00'. These octets shall not be included in the secured data.

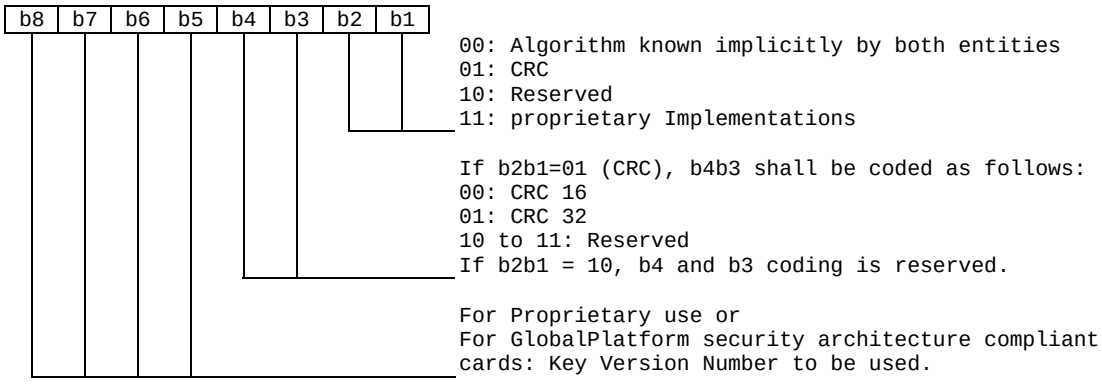
The use of (single) DES is deprecated. However the coding is reserved for backwards compatibility with pre Release 8 versions of the present document.

AES is the block cipher algorithm specified in FIPS-197 [12]. AES in CMAC mode is defined in NIST SP 800-38B [14]. The key length shall be known implicitly by both entities and shall be 128, 192 or 256 bits. AES shall be used together with counter settings (b5 and b4 of the first octet of SPI) 10 or 11. Each CMAC key shall be configured to accept and generate only cryptographic checksums with one length as defined in ETSI TS 102 226 [9] for PUT KEY for AES; the length of the cryptographic checksum shall be 32 or 64 bits.

For GlobalPlatform security architecture compliant cards see annex A.

5.1.3.2 Coding of the KID for Redundancy Check

If b2b1= '01' (Redundancy Check) in the first byte of SPI, KID shall be coded as follows:



CRC algorithm is specified in ISO/IEC 13239 [11].

The generator polynomial used for CRC 16 shall be X^16 + X^12 + X^5 + 1.

The generator polynomial used for CRC 32 shall be
X^32 + X^26 + X^23 + X^22 + X^16 + X^12 + X^11 + X^10 + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1.

The least significant bit of the first byte to be included in the checksum shall represent the most significant term of the input polynomial.

The least significant term of the output polynomial shall represent the most significant bit of the first byte of the RC/CC/DS field.

The initial value of the register shall be 'FFFF' for CRC 16 and 'FFFFFFFF' for CRC 32.

The CRC result is obtained after an XOR operation of the final register value with 'FFFFFFFF' for CRC 32 or 'FFFF' for CRC 16.

For GlobalPlatform security architecture compliant cards see annex A.

5.1.4 Counter Management

If in the first SPI byte b4b5 = 00 (No counter available) the counter field shall be ignored by the RE and the RE shall not update the counter.

If b5 of the first SPI byte is equal to 1 then the following rules shall apply to counter management, with the goal of preventing replay and synchronization attacks:

- The SE sets the counter value. It shall only be incremented.
- The RE shall update the counter to its next value upon receipt of a Command Packet after the corresponding security checks (i.e. RC/CC/DS and CNTR verification) have been passed successfully.
- The next counter value is the one received in the incoming message.
- When the counter value reaches its maximum value the counter is blocked.

If there is more than one SE, care has to be taken to ensure that the counter values remain synchronized between the SEs to what the RE is expecting, irrespective of the transport mechanism employed.

The level of security is indicated via the proprietary interface between the Sending/Receiving Application and Sending/Receiving Entity. Application designers should be aware that if the Sending Application requests "No RC/CC/DS" or "Redundancy Check" and "No Counter Available" from the SE, no security is applied to the Application Message and therefore there is an increased threat of malicious attack.

For GlobalPlatform security architecture compliant cards see annex A.

5.2 Response Packet structure

Table 3: Structure of the response packet

Element	Length	Comment
Response Packet Identifier (RPI)	1 octet	Identifies a Response Packet.
Response Packet Length (RPL)	variable	Indicates the number of octets from and including RHI to the end of Additional Response data, including any padding octets.
Response Header Identifier (RHI)	1 octet	Identifies the Response Header.
Response Header Length (RHL)	variable	Indicates the number of octets from and including TAR to the end of the RC/CC/DS.
Toolkit Application Reference (TAR)	3 octets	This shall be a copy of the contents of the TAR in the Command Packet.
Counter (CNTR)	5 octets	This shall be a copy of the contents of the CNTR in the Command Packet.
Padding counter (PCNTR)	1 octet	This indicates the number of padding octets at the end of the Additional Response Data.
Response Status Code Octet	1 octet	Codings defined in table 5.
Redundancy Check (RC), Cryptographic Checksum (CC) or Digital Signature (DS)	variable	Length depending on the algorithm indicated in the Command Header in the incoming message. A typical value is 4 octets to 8 octets, or zero if no RC/CC/DS is requested.
Additional Response Data	variable	Application Specific Response Data, including possible padding octets. The presence, length and coding of this field is defined by the application. This shall be empty for standardized response status codes different from "00".

Unless indicated otherwise, the RPL and RHL shall be coded according to BER-TV's coding of length in ETSI TS 101 220 [8].

Table 4: Linear representation of response packet

RPI	RPL	RHI	RHL	TAR	CNTR	PCNTR	Status Code	RC/CC/DS	Additional response data with padding
					Note 1	Note 1	Note 1	Note 1	Note 1
Note 3	Note 3	Note 3	Note 3	Note 2	Note 2	Note 2	Note 2		Note 2

NOTE 1: If ciphering is indicated in the Command Packet SPI then these fields shall be ciphered.
NOTE 2: These fields shall be included in the calculation of the RC/CC/DS.
NOTE 3: Part or all of these fields may also be included in the calculation of the RC/CC/DS, depending on implementation (e.g. SMS).

If ciphering is indicated, first the RC/CC/DS shall be calculated as indicated in note 2 of table 4, and then ciphering shall be applied, as indicated in note 1 of table 4.

If the SPI indicates that a specific field is unused, than its contents shall be set to zero, and ignored by the recipient of the Response Packet.

If the SPI in the Command Packet indicates that no RC, CC or DS is present in the Command Header, this field shall be of zero length.

Padding octets may consist of any plaintext value. If the Padding Counter content is zero, this shall indicate no padding octets are present, or no padding is necessary.

Table 5: Response status codes

Status Code (hexadecimal)	Meaning
'00'	PoR OK.
'01'	RC/CC/DS failed.
'02'	CNTR low.
'03'	CNTR high.
'04'	CNTR Blocked.
'05'	Ciphering error.
'06'	Unidentified security error. This code is for the case where the Receiving Entity cannot correctly interpret the Command Header and the Response Packet is sent unciphered with no RC/CC/DS.
'07'	Insufficient memory to process incoming message.
'08'	This status code "more time" should be used if the Receiving Entity/Application needs more time to process the Command Packet due to timing constraints. In this case a later Response Packet should be returned to the Sending Entity once processing has been completed.
'09'	TAR Unknown.
'0A'	Insufficient security level.
'0B'	Reserved for 3GPP (see ETSI TS 131 115 [5]).
'0C'	Reserved for 3GPP (see ETSI TS 131 115 [5]).
'0D' to 'BF'	Reserved for future use.
'C0' to 'FE'	Reserved for proprietary use.
'FF'	Reserved for future use.

6 Implementation for CAT_TP

The generic secured Command Packet and secured Response Packet are contained in the CAT_TP Packet Data as defined in ETSI TS 102 127 [10].

In the Command Packet, the Command Packet Identifier (CPI) value is '01' and the Command Header Identifier (CHI) is a Null field. CPI, CPL and CHL shall be included in the calculation of the RC/CC/DS.

In the Response Packet, the Response Packet Identifier (RPI) value is '02' and the Response Header Identifier (RHI) is a Null field. RPI, RPL and RHL shall be included in the calculation of the RC/CC/DS.

Packet Identifier values '00' to 'BF' and 'FF' are reserved for use in the present document. Values 'C0' to 'FE' are reserved for proprietary implementations.

The CAT_TP ports shall be used to address the applications running on top of CAT_TP, except when the CAT_TP link was opened by a PUSH command according to ETSI TS 102 226 [9] that was sent to Multiplexing application identified by its TAR defined in ETSI TS 101 220 [8]. In that case, incoming packets on that CAT_TP link shall be forwarded by TAR addressing.

The counter may be used also by an application to correlate Command Packets with Response Packets.

7 Implementation for TCP/IP

Before sending secured packets, the sending entity shall open a TCP/IP connection using the push mechanism defined in ETSI TS 102 226 [9]. The mechanisms to achieve this are provided by the Bearer Independent Protocol of ETSI TS 102 223 [18] or a direct IP connection as specified in ETSI TS 102 483 [17]. Optionally, this communication may be additionally secured using IPsec or TLS as detailed in ETSI TS 102 484 [16].

One or more generic secured command packets shall be transported from the sending entity as TCP application data. One or more generic secured response packets shall be transported from the receiving entity as TCP application data.

In the Command Packet, the Command Packet Identifier (CPI) value is '01' and the Command Header Identifier (CHI) is a Null field. CPI, CPL and CHL shall be included in the calculation of the RC/CC/DS.

In the Response Packet, the Response Packet Identifier (RPI) value is '02' and the Response Header Identifier (RHI) is a Null field. RPI, RPL and RHL shall be included in the calculation of the RC/CC/DS.

Packet Identifier values '00' to 'BF' and 'FF' are reserved for use in the present document. Values 'C0' to 'FE' are reserved for proprietary implementations.

Incoming secured packets shall be forwarded to the applications based on the TAR.

The counter may be used also by an application to correlate Command Packets with Response Packets.

In addition, an identification packet is defined that is used as defined in ETSI TS 102 226 [9].

Table 6: Format of the identification packet

Element	Length	Comment
Identification Packet Identifier	1 octet	Identifies that this data block is an Identification Packet. Value: '03'
Identification Packet Length	variable	This shall indicate the number of octets of the following fields. Coded as defined for BER TLVs in ETSI TS 101 220 [8].
Identification data tag	1 octet	Identification data as defined in ETSI TS 102 226 [9], coded as TLV object like Channel data specified in ETSI TS 102 223 [18].
Length of identification data string	1 octet	
Identification data string	N octets	

8 Secured message structure for HTTPS

Secure messages shall be sent in the body part of HTTP requests and responses as specified in Amendment B of the Global Platform Card Specification v2.2 [19] with the modifications given below.

Unless defined otherwise for an application, the following header fields shall take the values below:

- The Content-Type header field of the HTTP POST request shall be set to:

  application/vnd.etsi.scp.response-data;version=1.0

- The Content-Type header field of the HTTP POST response shall be set to:

  application/vnd.etsi.scp.command-data;version=1.0

- The X-Admin-Targeted-Application header field of the HTTP POST response shall contain the AID of the targeted application formatted as defined in Amendment B of the Global Platform Card Specification v2.2 [19] for the Security Domain AID.

For each command message, a corresponding response message shall be exchanged.

Switching to a different targeted application or to/from other receiving entities on the card shall be supported as specified in Amendment B of the Global Platform Card Specification v2.2 [19].

Annex A (normative): Relation between security layer and GlobalPlatform security architecture

A.0 Overview

This annex only applies to cards implementing the security architecture defined in the GlobalPlatform Card Specification [6].

At least one Security Domain shall be present on the card: the Issuer Security Domain.

The security of Application Messages (i.e. RC/CC/DS, ciphering/deciphering, counter management) shall be managed by a Security Domain as specified in the UICC Configuration [15].

A.1 Key version - counter association within a Security Domain

For each Security Domain, a dedicated counter shall be associated to each key version as described in table A.1.

Table A.1

	Key Version Number '00'	**Key Version Number '01'**	**...**	**Key Version Number n (maximum '0F')**
	Reserved	**Counter 1**		**Counter n**
Key Identifier 1	Reserved	KIc 1		KIc n
Key Identifier 2	Reserved	KID 1		KID n
Key Identifier 3	Reserved	DEK 1		DEK n
NOTE 1: The Key Version Number is defined in GlobalPlatform Card Specification [6]. The range from '01' to '0F' is used in the present document. NOTE 2: The Key Identifier is defined in GlobalPlatform Card Specification [6]. The range from '01' to '03' is used in the present document.				

A.2 Security keys KIc, KID

The indication of the key to be used in the KIc and KID (bits b8..b5) fields shall refer to a GlobalPlatform key version number.

The algorithm to be used with the key shall be the algorithm associated with the key (as described in the GlobalPlatform Card Specification [6] as detailed in the UICC Configuration [15]).

The key version number indicated in the KIc and KID (bits b8..b5) fields shall be identical when different from 0. If the key version numbers are different (and both different from 0) then the message shall be rejected with the "Unidentified security error" Response Status Code.

If no ciphering is applied to the command packet (i.e. SPI1.b3 = 0), '00' may also be used as a valid value for the KIc field.

If no authentication is applied to the command packet (i.e. SPI1.b2b1 = 00 or 01), '00' may also be used as a valid value for the KID field.

Annex B (informative): Example for CRC computation

Example C code for calculating the CRC32:

```
#include <stdio.h>

typedef unsigned char byte;

/* note that bit 0 from the CRC and the polynomial is the MSB of the implementation */
void CRC32(byte in[], int length, byte out[4]){
  unsigned long crc;
  int bit, byte, carry;
  crc = 0xFFFFFFFF; /* initialization */
  for (byte=0; byte<length; byte++){
        for (bit=0; bit<8; bit++){
                carry = crc & 1 ;
                crc >>=1;
                if (carry ^((in[byte] >> bit) & 1))
                    crc ^= 0xedb88320 ; /*polynomial, bit X^32 is handled by the carry */
        }
  }
  crc = ~crc; /* invert CRC */
  out[0]=crc>>24;
  out[1]=crc>>16;
  out[2]=crc>>8;
  out[3]=crc;
}



int main(int argc, char* argv[])
{
    byte in[]={1,2,3,4,5};
    byte c[4];
    CRC32(in, sizeof(in), c);
    printf("crc = 0x%2.2X%2.2X%2.2X%2.2X\n",c[0],c[1],c[2],c[3]);
    return 0;
}
```

EXAMPLE 1: If an input message is '01 02 03 04 05' where '01' is the first byte and '05' the last byte used for the computation, then the result of CRC 32 computation applied to the input message is '47 0B 99 F4', where '47' would represent the first byte and 'F4' the last byte of the RC/CC/DS field.

EXAMPLE 2: If an input message is '01 02 03 04 05' where '01' is the first byte and '05' the last byte used for the computation, then the result of CRC 16 computation applied to the input message is '22 EC', where '22' would represent the first byte and 'EC' the last byte of the RC/CC/DS field.

Annex C (informative): Change history

This annex lists all Changes Requests (CR) applied to the present document.

History Table								
Date	**Meeting**	**Tdoc**	**CR**	**Rv**	**Cat**	**Changes**	**Old**	**New**
2001-10	T3 API #9	T3a010195				Initial version is based on 3GPP TS 23.048 V5.1.0.	-	0.0.0
2001-11	T3#21/ SCP#8	T3-010669/ SCP-010345				Submitted to 3GPP T3#21 - TC SCP#8. Editorial changes.	0.0.0	0.0.1
2001-11	SCP#8	SCP-010376				Editorial and formatting enhancements. Version number raised to 1.0.0 in line with decision at TC SCP #8.	0.0.1	1.0.0
2002-01	SCP2#2	SCP2-020018				Updated to include the results of TSG-T#14 and editorial changes.	1.0.0	1.1.0
2002-03	SCP#9	SCP-020048				Updated to include the results of TSG-T#15. Submitted to SCP#9 for approval.	1.1.0	2.0.0
2002-03	SCP#9	SCP-020056				Editorial changes after discussion at SCP#9. This version has been sent to the ETSI secretariat for publication in March 2002 as ETSI TS 102 225 V6.0.0. No technical changes compared to v2.0.0.	2.0.0	6.0.0
2003-01	SCP#12	SCP-030021	001		A	Clarification on the RC/CC/DS coding in SPI2.	6.0.0	6.1.0
2003-05	SCP#13	SCP-030166	002		F	Clarification of the description/usage of b3 b4 of KIc and KID.	6.1.0	6.2.0
		SCP-030123	003		F	TAR coding clarification.		
2003-12	SCP#15	SCP2-030254	005	2	B	Update of ETSI TS 102 225 to GlobalPlatform Card Specification V2.1.1.	6.2.0	6.3.0
		SCP2-030262	006		B	Implementation of Secure Packet on CAT_TP.	6.2.0	6.3.0
		SCP2-030264	007		C	Allow CPI, CHI, RPI and RHI to be included in Checksum Calculations.	6.2.0	6.3.0
2004-02	SCP#16	SCP2-040014	008		B	Reservation of a new RSC for 3GPP.	6.3.0	6.4.0
2004-02	SCP#16	SCP-040093	009	1	B	Redundancy Check algorithms.	6.3.0	6.4.0
2004-09	SCP#18	SCP-040323	010		F	Correction to the range of Key Version Number and Key Identifier according to GlobalPlatform Card specification.	6.4.0	6.5.0
			011		F	Correction of the coding of KID in case of Redundancy Check for GlobalPlatform compliant cards.		
			012		B	Introduction of a proprietary range of packet identifiers for CAT_TP.		
2004-11	SCP#19	SCP-040417	013		F	Clarification for non-specific references.	6.5.0	6.6.0
2005-01	SCP#20	SCP-050059	014	1	B	Reserve a proprietary range of response status codes.	6.6.0	7.0.0
2005-10	SCP#22	SCP-050230	015		D	Addition of examples of result of CRC 16 computation.	7.0.0	7.1.0
2005-12	SCP#23	SCP-050439	016		B	Reservation of Response Status Code value "0C" for 3GPP.	7.1.0	7.2.0
2006-03	SCP#25	SCP-060131	018		A	Correction to ISO standard references.	7.2.0	7.3.0
			022		A	Coding of RPL,RHL,CHL,CPL.		
		SCP-060156	020	1	A	Clarification of presence of ARD.		
2007-10	SCP#33	SCP-070439	023	1	B	Addition of the capability to multiplex RAM and RFM sessions over a single CAT_TP link in a BIP session.	7.3.0	8.0.0
2008-10	SCP#39	SCP-080427	024		B	Addition of AES for encryption/decryption and cryptographic checksum.	8.0.0	8.1.0
2008-10	SCP#39	SCP-080427	025		C	Deprecate the use of single DES.	8.0.0	8.1.0
2009-01	SCP#40	SCP-090023	026		B	Addition of data download over IP.	8.1.0	8.2.0
2009-01	SCP#40	SCP-090053	027		C	Update to GlobalPlatform Card Specification v2.2 (CR renumbered to 027).	8.1.0	8.2.0
2009-10	SCP#43	SCP-090325	028		C	Precision of packet handling for TCP.	8.2.0	9.0.0
2009-10	SCP#43	SCP-090352	029		B	Addition of HTTPS.	8.2.0	9.0.0
2010-04	SCP#44	SCP(10)0039	032	-	A	Correction of duplicate status code allocation due to CR implementation error.	8.2.0	9.0.0
2010-04	SCP#44	SCP(10)0039	033	-	F	Clarification of network mechanisms for TCP.	8.2.0	9.0.0
2011-05	SCP#49	SCP(11)0181	034		A	Correction of reference to NIST SP 800-38B.	9.0.0	9.1.0
2011-12	SCP#53	SCP(11)0371	036		A	Clarification key configuration mechanism for AES keys	9.1.0	9.2.0
2012-02						Rel-10 of the specification is created as a step in creation of Rel-11. No changes in the technical content compared to V9.2.0.	9.2.0	10.0.0
2011-12	SCP#53	SCP(11)0372r1	037	1	C	Update of references to GlobalPlatform specifications.	10.0.0	11.0.0
2013-10	SCP#61	SCP(13)000232	039	-	C	Modification of response handling.	11.0.0	12.0.0
2013-10	SCP#61	SCP(13)000234	040	-	C	Update of reference to GlobalPlatform Amendment B.	11.0.0	12.0.0
2014-04	SCP#63	SCP(14)000101r1	041	2	C	Coding of the SPI to match security of PoR to the Command Packet.	11.0.0	12.0.0
2014-06	SCP#64	SCP(14)000171r1	042	1	C	Coding of the SPI - Byte 2.	12.0.0	12.1.0
2014-12	SCP#67	SCP(15)000044	043		F	Correction of implementation of CR025.	12.1.0	12.2.0
2015-04	SCP#68	SCP(15)000121r1	044		C	Clarification on KIc and KID fields values.	12.2.0	13.0.0

History Table								
Date	**Meeting**	**Tdoc**	**CR**	**Rv**	**Cat**	**Changes**	**Old**	**New**
2015-06	SCP#69	SCP(15)000171	045		D	Clarification of the Additional Response Data for Remote Management Applications.	12.2.0	13.0.0
2015-04	SCP#73	SCP(16)000085	046		F	Update of references to GP specifications after publication of Card Specification V2.3.	12.2.0	13.0.0
2020-12						Automatic Upgrade	13.0.0	14.0.0
2020-12						Automatic Upgrade	14.0.0	15.0.0
2020-06	SCP#91	SCP(19)000232r1	047		D	Alignment of definitions and abbreviations with ETSI TR 102 216.	13.0.0	16.0.0
2020-12						Minor editorial correction	16.0.0	16.0.1

History

Document history		
V16.0.0	June 2020	Publication
V16.0.1	December 2020	Publication