International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 7 July 2022, pp: 412-418 www.ijaem.net ISSN: 2395-5252

Overview on Secured Packet Structure for Universal Integrated Circuit Card Based Applications

Sushma S Ankad

Submitted: 01-07-2022 Revised: 07-07-2022 Accepted: 10-07-2022

ABSTRACT
Secured Packets contain application messages to which certain mechanismshavebeen applied. Application messages are commands or data exchanged between an application resident in or behind thenetwork and on the UICC.The Sending/Receiving Entity in the network and the UICC are responsible for applying thesecurity mechanisms to the application messages and thus turning them into Secured Packets.It is applicable to the exchange of secured packets between an entity in a network and an entity in the UICC.
The question of security arises when the applications on UICC manages to get encryption keys to the correct security domain without compromising on the keys used. So, security of the packet should be needed.
Keywords: Secured Packet, UICC

I. INTRODUCTION

The Sending Application prepares an Application Message and forwards it to the Sending Entity, with an indication ofthe security to be applied to the message.The Sending Entity prepends a Security Header (the Command Header) to the Application Message. It then applies therequested security to part of the Command Header and all of the Application Message [1], 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 [6]) before unpacking it. The Receiving Entity subsequently forwards theApplication Message to the Receiving Application indicating to the Receiving Application the security that was applied.

If so indicated in the Command Header, the Receiving Entity shall create a (Secured) Response Packet. The ResponsePacket consists of a Security Header (the Response Header) and optionally, application specific data supplied by theReceiving Application. Both the Response Header and the application specific data are secured using the securitymechanisms 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 from receiving entity (i.e. for Short Message Service – Cell Broadcast) the SendingApplication may have requested a response. In this case a (Secured) Response Packet could be sent using a differentbearer by the Receiving Application. The figure 1 describes the overview of security system.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 7 July 2022, pp: 412-418 www.ijaem.net ISSN: 2395-5252

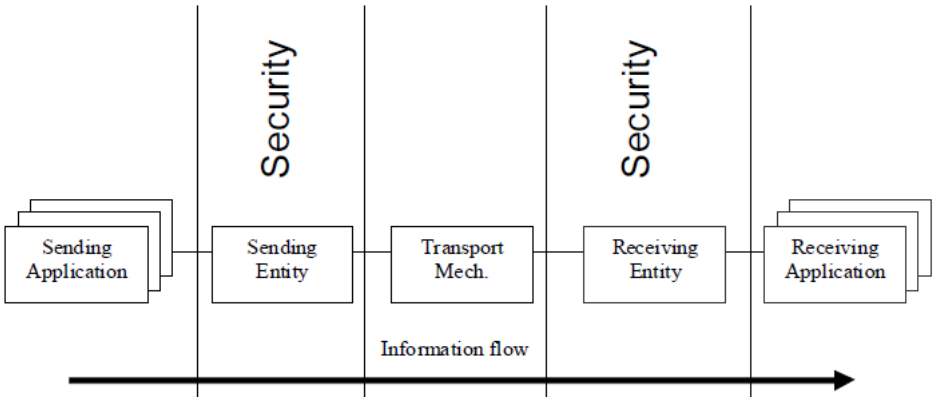

Figure 1: Security System

II. GENERALIZED SECURED PACKET STRUCTURE

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

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. The Command Packet Identifier (CPI) identifies that this data block is the secured Command Packet, Command Packet Length (CPL) is variable in size. This shall indicate the number of octets from and including theCommand Header Identifier to the end of the Secured Data,including any padding octets required for ciphering. Command Header Identifier (CHI) identifies the command header. Command Header Length (CHL) shall indicate the number of octets from and including the SPIto the end of the RC/CC/DS.

Table 1: Structure of the command packet

Element	Length
Command Packet Identifier (CPI)	1 octet
Command Packet Length (CPL)	variable
Command Header Identifier (CHI)	1 octet
Command Header Length (CHL)	variable
Security Parameter Indicator (SPI)	2 octets
Ciphering Key Identifier (KIc)	1 octet
Key Identifier (KID)	1 octet
Toolkit Application Reference (TAR)	3 octets
Counter (CNTR)	5 octets
Padding Counter (PCNTR)	1 octet
Redundancy Check (RC), Cryptographic Checksum (CC) or Digital Signature (DS)	variable
Secured data	variable

Table 2: Linear representation of command packet

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 7 July 2022, pp: 412-418 www.ijaem.net ISSN: 2395-5252

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 cipheringshall 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 theReceiving 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 zerolength.

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

III. CODING OF ELEMENTS IN SECURED PACKET STRUCTURE

a) **Coding of SPI Byte:** The security parameter indicator is coded in two byes described the first octet of SPI byte in Figure 2 and Second octet of SPI byte in Figure 3.

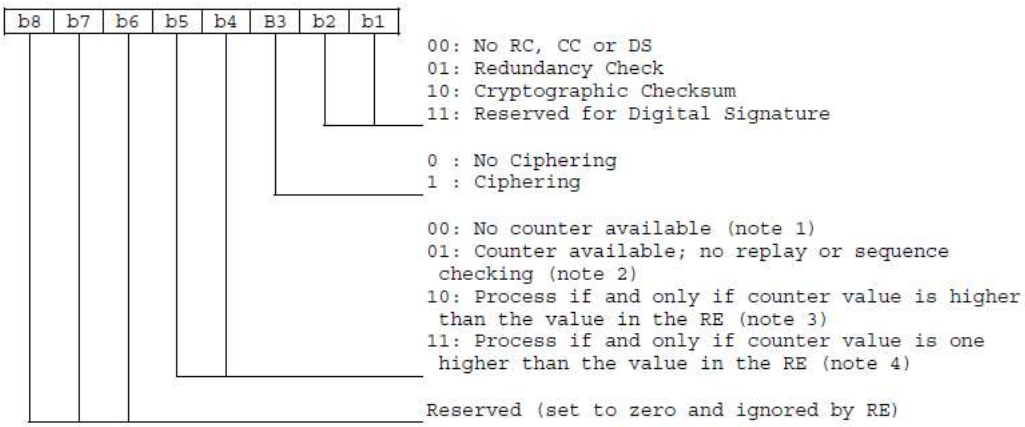

Figure 2: First Octet of SPI Byte

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 theCommand Packet was successfully unpacked, the counter value can be forwarded from the ReceivingEntity to the Receiving Application. This depends on proprietary implementations and happens in anapplication dependent way.
NOTE 3: The counter value is compared with the counter value of the last received Command Packet. This istolerant to failures on the transport level (i.e. losses of Command Packets). A possible scenario is a globalupdate.
NOTE 4: This provides strict control in addition to security indicated in note 3.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 7 July 2022, pp: 412-418 www.ijaem.net ISSN: 2395-5252

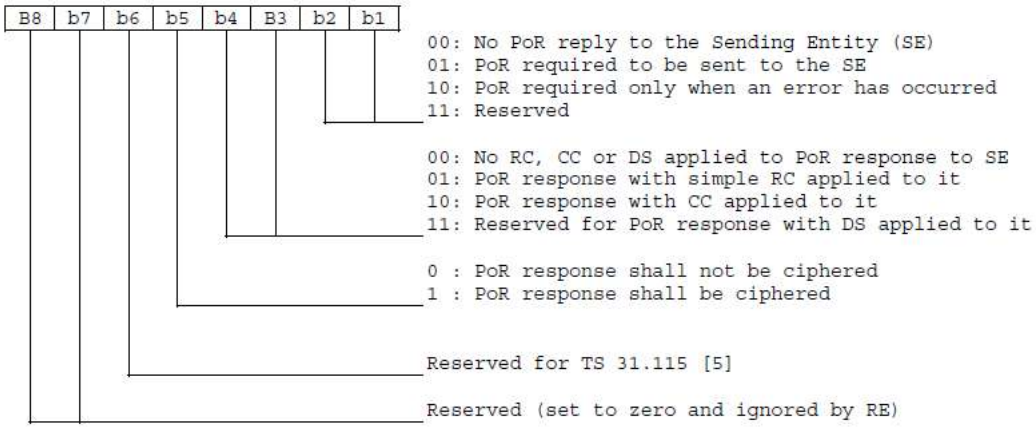

Figure 3: Second Octet of SPI Byte

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 thesame 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 beensuccessfully authenticated (e.g. using CC) and if ciphering was applied to the Command Packet, i.e. SPI1.b3 isset to '1'.

b) **Coding of KIC (Key and algorithm Identifier for ciphering) Byte**

KIC byte in an element in secured packet structure, which describes about the ciphering key identifier, coding of KIC is shown in Figure 4.

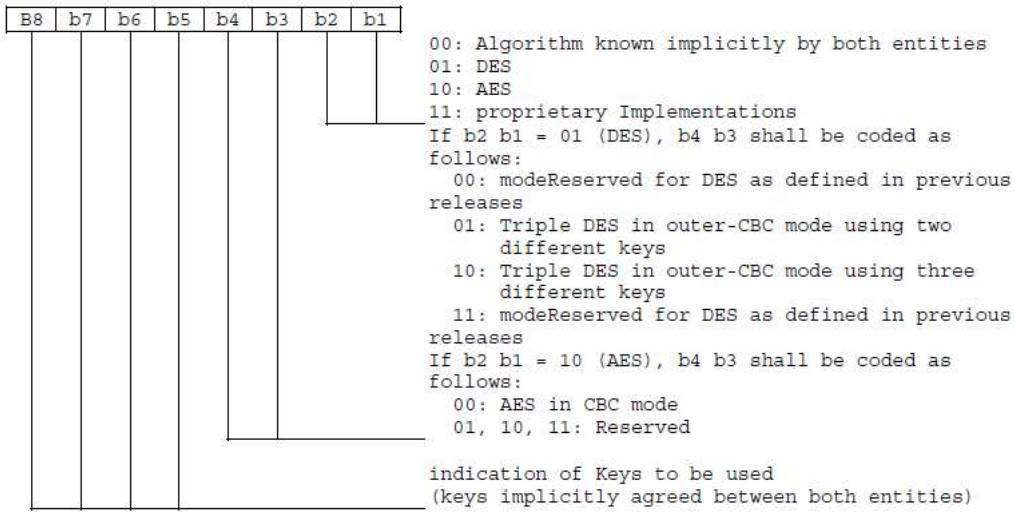

Figure 4: Coding of KIC Byte

c) **Coding of the KID for Redundancy Check**

The KID identifies the Key and algorithm Identifier for RC/CC/DS. The coding of KID for Redundancy Check is defined in Figure 5.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 7 July 2022, pp: 412-418 www.ijaem.net ISSN: 2395-5252

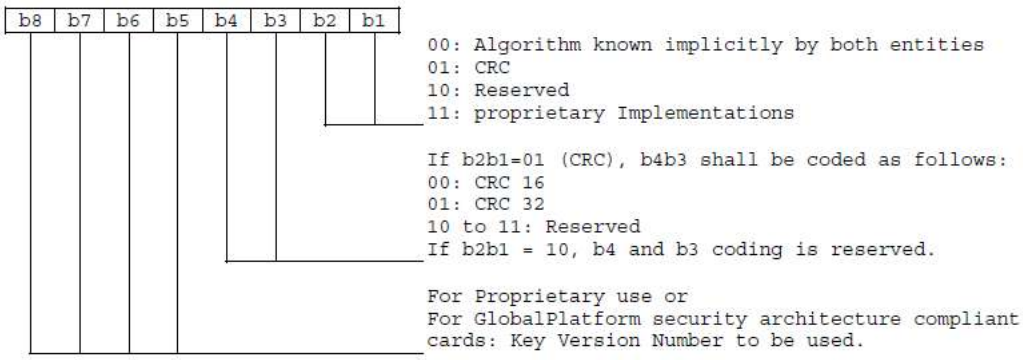

Figure 5: Coding of the KID for Redundancy Check

d) **Coding of Counter Management**

This counter management is defined with 5 bytes.If in the first SPI byte b4b5 = 00 (No counter available) the counter field shall be ignored by the RE and the RE shallnot update the counter. The range of values for CNTR is from '0000000000' to 'FFFFFFFFFF'.

IV. RESPONSE PACKET STRUCTURE

When the UICC has received the command header, a response containing a procedure byte, or a status byte shall be sentto the terminal. Both the terminal and the UICC shall be able to keep track of the direction of the data flow and who hasthe access to the I/O-line. The response packet structure is described in table 3 and linear representation is shown in table 4.The response status codes are defined in table 5.

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".

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 7 July 2022, pp: 412-418 www.ijaem.net ISSN: 2395-5252

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 cipheringshall 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 ofthe 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 beof zero length.

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

V. CONCLUSION

The command structure is onus for securing the communication between the sending entity and the receiving entity**.** So, major problems regarding the security of packet and command header processing will be avoided if the command packet header is correctly coded with the help of coding of each element like SPI, KIC, KID, Counter Management. This ensues the successful processing of the command header as well command data field.

REFERENCES

[1]. Jesani, Navin, Nishant Gupta, Somya Bhatt, Puja Singh, and Ankit Saxena. "Smart Card For Various Application In Institution." In 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), pp. 1-5. IEEE, 2020.

[2]. Li, Tian, Dazhi Sun, Peng Jing, and Kaixi Yang. "Smart card data mining of public transport destination: A literature review." Information 9, no. 1 (2018): 18.

[3]. Taherdoost, Hamed. "Appraising the smart card technology adoption; case of application in university environment." Procedia Engineering 181 (2017): 1049-1057.

[4]. Setyoko, Yoso Adi, and IGB Baskara Nugraha. "Multipurpose smart card system." In 2014 International Conference on ICT For Smart Society (ICISS), pp. 264-268. IEEE, 2014.

[5]. ETSI. 2008, ETSI TS 102 225 v8.0.0, Smart Cards; Secure packet structure for UICCbased applications.

[6]. ETSI. 2007, ETSI TS 102 226 v7.4.0, Smart Cards; Remote APDU structure for UICCbased applications.

[7]. ETSI. 2007, ETSI TS 102 222 v7.1.0, Integrated Circuit Cards (ICC); Administrativecommands for telecommunications applications.
[8]. ETSI. 2008, ETSI TS 102 241 v8.0.0, Smart Cards; UICC Application ProgrammingInterface (UICC API) for Java Card.
[9]. ETSI. 2008, ETSI TS 102 221 v8.0.0, Smart Cards; UICC - Terminal interface; Physicaland logical characteristics.