Watchdata

WD-CMC SAM Private Technical Manual v1.0

Table of Contents

Revision history

S/N	Date	Version	Description	Author
1	09-04-2013	v1.0	First version	Manik Biswas

Abbreviation

Definition	Description
3DES-CBC	3Key Triple DES operation in CBC mode
3DES-ECB	3Key Triple DES operation in ECB mode
CAN	Card Application Number
Card Rand	8 byte random number generated by the card

CMC	Common mobility card
CK2f	SFI of the EF containing the Credit Key #2
CK2n	Credit Key no 2 to be used by card in Credit transaction
CSN	Chip Serial Number
ADF	Application Directory file
EF	Elementary File
IV	Initialization vector to be used in Triple DES CBC operation
Pf	Purse File SFI
PTC	Purse Transaction Counter
Term-Date & Time	4-byte Date and Time value (in seconds) provided by the terminal
Term-Rand	8-byte random number generated by the terminal
TRP	Terminal Reference Parameter
SKf	SFI of the signature key file
SKn	Signing key number to be used by the card

Introduction

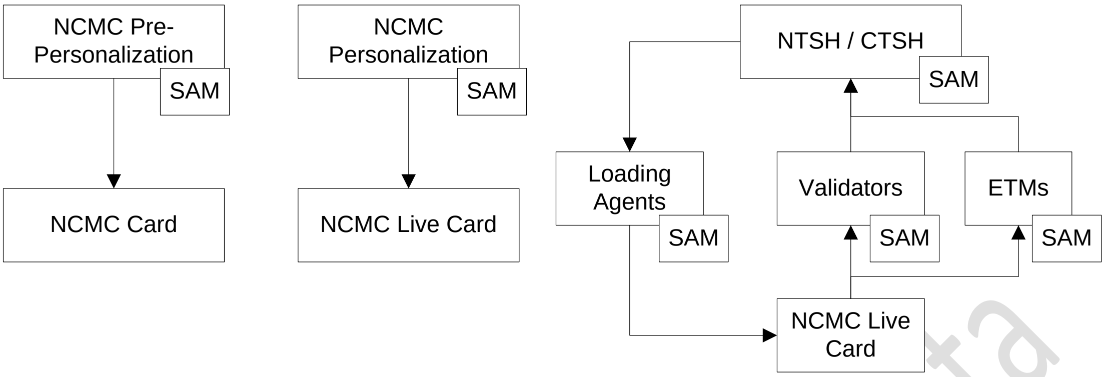
Purpose

This document shall describe the functionality of CMC SAM and all interfaces required to communicate with it. Also it shall specify the required command sets for personalizing TimeCOS CMC SAM card. This document prepared based on NCMC-SAM specification document of UTIITSL.

The intended audience for this document would be all personnel who want to work with Watchdata TimeCOS CMC SAM Cards.

SAM usage in an NCMC enabled system

NCMC SAM shall be used at different stages of a NCMC enabled system.



NCMC is a contactless payment card to be used for payment in the public transport system. The authentication and modification of this card is subject to the use of certain secret keys to ensure the authenticity of the card and the terminal and thereby forbidding the unauthorized usage. To make this process secure, SAM card shall be used as a container of all such secret keys and also for authenticating the NCMC card and terminals.

As depicted above the SAM card shall cater the following in an NCMC enabled system.

- It shall generate all the diversified keys which are to be personalized into the NCMC card.
- The Loading Agents, which are responsible for loading money / fare products into the NCMC card, shall use the SAM card to secure the transaction.
- The Validators / ETMs shall also use SAM card to secure the transaction with the NCMC card.

The functional behavior of SAM has been customized to suite the NCMC transaction. Apart from providing the secure key management the SAM shall also prepare and verify the cryptograms used in NCMC commands. Also, it shall authenticate the user once at each power up session.

The SAM cards are categorized based on the above arrangement. Each such SAM shall contain only the keys and PINs required for its functionality.

SAM	Purpose
Master SAM	This shall contain all the Super Master Keys and shall be used to generate other SAM master keys. It shall also be used for securely personalizing the Master Keys in other SAM cards.
Personalization SAM	This shall contain all the Master Keys required for deriving the Keys for NCMC card. It shall also be used for securely personalizing the NCMC keys into the NCMC cards.
Credit SAM	This shall contain all the Master keys which have been used to derive the Credit Keys for NCMC card. This SAM shall be present in the loading agents where the NCMC card shall be topped up.
Debit SAM	This shall contain all the Master Keys which can be used for Debit operation using an NCMC card. It shall be present in all the validators and other terminals which shall debit the NCMC card.
Combo SAM	A combination of different SAM Keys in the same SAM.

Product SAM	This shall contain all the Master Keys which can be used to update the Product File in the NCMC card.

Note: In the rest of the document the Master SAM shall be referred as Master SAM and the remaining 4 SAMs shall be referred as OtherSAM.

Master SAM

This SAM shall be used only for deriving Master Keys for Other SAMs. It shall not take part in normal transactions with NCMC card. Multiple Super Master Keys shall be personalized into this SAM which in turn can be used for deriving the Other SAM Master Keys. Master SAM shall be required in those devices which will personalize all Other SAMs.

Personalization SAM

This SAM shall hold at least 2keys for each key attribute. Personalization SAM shall be required in those devices which shall personalize the NCMC card. This SAM can’t be used for performing any transaction with NCMC card.

Credit SAM

This SAM shall only contain the Master Keys with Credit attribute and shall be used in those devices which shall perform the Credit Purse operation with NCMC card. It shall contain at least 2 Master Keys with Credit Key Derivation attribute so that if required keys can be switched instantly.

Debit SAM

This SAM shall only contain the Master Keys with Debit attribute and shall be used in those devices which shall perform the Debit Purse operation with NCMC card. It shall contain at least 2 Master Keys with Debit Key Derivation attribute so that if required keys can be switched instantly.

Combo SAM

Combo SAM is a special SAM which shall contain Master Keys with mixed attributes. This SAM would be required in situation where same device is required to perform multiple operations with the NCMC card, such as Credit and Debit together. This will hold 2 sets of Master Keys for each operation.

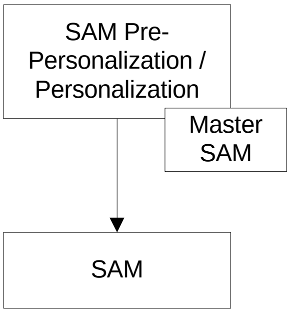
Product SAM

The product information file in NCMC card shall be bind with a particular key so that this EF can only be updated when that key has been used. This SAM shall contain the Master Key from which that particular key has been derived and shall be used in those devices which shall update the Product Information EF in NCMC card.

Based on the above SAM categories the supported command sets for each SAM shall change. Please refer to Supported Command Sets section for more details on this.

SAM management

As SAM shall store all the secret master keys so initialization and personalization of such cards should also have to be secure enough. To achieve this, a master SAM card shall be used. This card shall derive all other SAM master keys and securely personalize those. The keys inside the master SAM shall be personalized in a physically secured location.



File System

According to the SAM Specification there is no specific file system requirement for SAM cards. However, it must contain certain EFs for storing the SAM master keys and PIN. TimeCOS supports commands for creating such EFs in the card.

The format of Keys and PINs must be as shown in the following sections. It should be possible to refer to these keys and PINs by their respective numbers.

Key Storage

All SAM master keys shall be stored inside some record EFs. The SFI of this EF can be anything except 0x01. The format of each key record must be as shown below.

Key Header						
Key no	**Key Attribute**	**Counters**		**Diversification Identifier**	**Txn Amount Limit**	**Key Value**
		Retry Counter	**Max Retry Counter**			
1-byte	1-byte	1-byte		1-byte	3-bytes	24-bytes

Key No:

This is 1-byte value using which the key can be accessed by the transaction commands.

Key Attribute:

This is 1-byte value associated with each key. The meanings of each bit of this byte are:

BIT 7~6	**BIT 5**	**BIT 4**	**BIT 3**	**BIT 2**	**BIT 1**	**BIT 0**
RFU	SAM Master Key derivation	Auto-Topup Key derivation	Signature Key derivation	Debit Key derivation	Credit Key derivation	Key Validity

BIT value 0b: The related operation is allowed

BIT value 1b: The related operation is not allowed

Counters:

The high nibble of this byte shall represent the number of wrong retries using this key. And the lower nibble will represent the maximum allowed number of retries. The retry counter shall have to be incremented for each wrong usage of the key. For example, if the key attribute says only Credit key derivation allowed however, the key has been used to derive a Debit key then the retry counter shall have to be incremented.

Diversification Identifier

This byte shall identify the key diversification data from a diversification data pool. More information on this byte is available in section Key Diversification.

Txn Amount Limit

This 3-bytes shall contain the maximum credit / debit transaction amount for which this key can be used. It’s a signed number and shall be positive. During the processing of Compute Credit Crypt or Compute Debit Crypt this field shall be checked against the amount provided and appropriate action shall be taken.

PIN Storage

SAM card PINs shall be stored in a record EF. The SFI of this EF must be 0x01. The length of PIN must not be less than 0x04 bytes and must not be greater than 0x08 bytes. Each PIN record should have the format as shown below. Each SAM card can have 2 such PINs with PIN no 0x01 and 0x02.

PIN no	PIN Length	Counters		PIN Value
		Retry Counter	Max Retry Counter	
1 byte	1 byte	1 byte		4~8 bytes

PIN no

This 1 byte value can be used to identify the PIN inside the card.

PIN Length

This field shall contain the actual PIN length. It can range from 0x04 to 0x08.

Counters

The high nibble of this byte shall represent the number of wrong presentation of the PIN. And the lower nibble will represent the maximum allowed number of retries. The retry counter shall increment for each wrong presentation of the PIN and shall be reset to zero once the correct PIN value has been produced.

Key Management

The key points which have to be addressed by the SAM key management are:

- Secure storage of all keys and related parameters.
- CMC application can be issued by multiple issuers. So each issuer should be able to uniquely identify its key.
- The application can be acquired by multiple acquirers, so there should be one unique key for each such entity.
- Post issuance should be possible. Which means it should be possible to add any new acquirer at any point of time.

Key diversification

SAM shall implement two ways of diversifying the Master Keys to get the Unique Keys. The diversification data and the related algorithm shall be identified by the Diversification Identifier present in the each key header.

Diversification Data

Each SAM master key shall be associated with an 8-byte data block and this block shall be identified by the Diversification Data Identifier.

Diversification Data Identifier	Algorithm Identifier	Diversification Random Data
1 byte	1 byte	6 bytes

Diversification Data Identifier

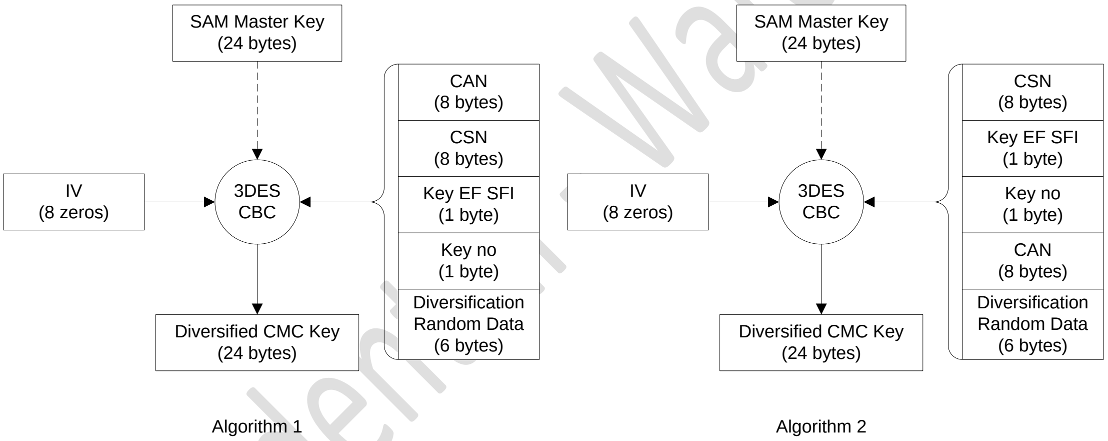
This shall identify the diversification algorithm and the random diversification data to be used for diversification.

Algorithm Identifier

This shall identify the algorithm to be used for the diversification process. The value of this byte can either be 0x01 or 0x02 to identify the Algorith1 or Algorithm2.

Diversification Random Data

These 6 bytes of random data shall be generated within the card and saved. This shall add an extra layer of security for the diversified keys.



Algorithm 1

Algorithm 2

Figure: Key diversification

These diversified keys shall be personalized into the CMC card.

Life Cycle Management

All SAM cards shall have the following life cycle states.

Life Cycle State	Description
MANUFACTURED	Cards which has just arrived from the factory. These cards shall have the Transport Key injected into it. The Transport Key shall be changed to Perso Key at this state.
PERSO	Cards from MANUFACTURED state shall be authenticated using the Perso Key and then all other Transaction related keys shall be personalized in to it. Also the SAM categorization shall happen at this stage and accordingly the Perso key shall be changed with Card Master Key.
USAGE	In this state the card shall behave according to its type.

BLOCKED	Due to fraudulent usage the card can block itself and come to this state.

Every SAM shall count its usage during the USAGE phase. Once this counter value overflows the SAM shall block itself and can't be used any further. This counter would be a 2 byte value and its maximum limit shall be decided during personalization.SAM Binding

The SAMs in the USAGE phase shall be attached to a particular terminal so that apart from the attached terminal that particular SAM can't be used anywhere else. This is achieved by storing the TRP of that particular terminal inside the SAM and checking for it during the Transaction related commands.

TimeCOS CMC SAM Security Architecture

TimeCOS CMC SAM security management scheme is based on TimeCOS security management system. Its security system includes the security state, security access conditions, security mechanism and cryptographic functions.

1. It will authenticate the card terminal devices before it can be accessed.
2. It first determines if the access to a particular file is allowed by comparing the value of the security state register with those required by the access condition.

Security State

Security State is referring to the security level at current directory. The MF and DF have 16 types of security level individually.

Two 4-bit register inside TimeCOS is used to indicate the current security state. Both registers can have any value between 0 to F. These two registers are:-

[1] MF security state register

It determines the security level for the global level.

[2] DF security state register

It is the security level at current DF level only.

MF security state register

1. This register will be reset to 0 after reset.

2. Moving from one application directories to another will not affect the value of the security state register.

3. Only successful PIN verification and external authentication will make the value of security state register at MF level be changed.

The security state register will be reset to 0 when the following happens:-

[1] After card reset.

[2] The PIN verification command or external authentication command return a failure code of 63CX.

When the current directory is MF, the security stare register at current directory will be equal to the security state register of MF.

DF security state register

The security state register will be reset to 0 when the following happens:-

[1] After card reset.

[2] Changes in current directory, such as, select father directory (which is not the MF), or select son DF.

[3] The PIN verification command or external authentication command return a failure code of 63CX.

Only PIN and external authentication of current DF will affect the security state register after a successful PIN verification and external authentication.

If current directory is MF, the security state register at current directory will be equal to the current security state register of this MF.

Security Access Condition

The access conditions must be fulfilled before the file can be accessed. This meant also the security state register must matches certain value before the operation can be executed.

Access condition can also be referred as individual access rights - i.e. read access right, write access right, create access right, add key access right and use access right. During file creation, each type of access right is represented by 1 byte.

As compared to other COS, TimeCOS uses a different method to control the access rights. It is using a predefined field to restrict others from accessing it illegally.

Assuming the value of security state register at current level is represented by V.

1. If the access condition of MF is "0Y", to access the files at this level, the security state register of MF must be equal or greater than 'Y'; i.e. for access condition = "0Y" V >= Y

2. If a particular file has a read access right of "05", which means that the MF security state register must be equal or greater than 5 before the file can be read; i.e. for read access right = "05" V >= 5

3. Let's say you are already access files at current level, if the access condition is 'XY' (in which X ≠ 0), which means the value in the security state register must fulfil both condition i.e. equal or greater than Y and also equal or smaller than X. For the case X>Y: - i.e. for access condition 'XY' where X > Y V >= Y V <= X

4. For the case X=Y then the security state register at current level must be equal to X.i.e. for access condition 'XY' where X=Y V = X = Y

5. For the case X<Y, it is a inhibit operation.

Example 1 : A file with write access right of 53, it means when writing to the file, the security state register must have a value of 3, 4 or 5.

Example 2 : A file with read access right of F0 and write access right of F1. It means the file can be read without any restriction. However, when comes to writing, the security state register must be equal or greater than 1.

Example: A file with read access right of FE, which means the security state register must match the value F or E before read access is granted.

Security mechanism

1. This referred to the security process involved in managing the transfer from one security state to the other state.

2. It uses the PIN verification and external authentication results to change the value of the security state register.

3. At MF level, the value of the security state register at MF and at DF level will be updated upon successful authentication. If not at MF level, upon successful verification, you can change the security state register at that level only.

4. When creating the PIN or external authentication key, the security state will indicates if the PIN verification is successful or external authentication is successful. The security state register will be set equal to the value of the current security state. e.g. If the security state of the PIN key is 1, which means the security state register will be equal to 1 after a successful PIN verification.

5. The current security state register will be set to 0 upon power-on-reset and when going from father-DF to son-DF or vice versa.

You may refer to the following example for more explanation on how the access management is being done. Refer below **Table.**

Assuming the card has a binary file which has been defined as :-

Read access right = F1 ;

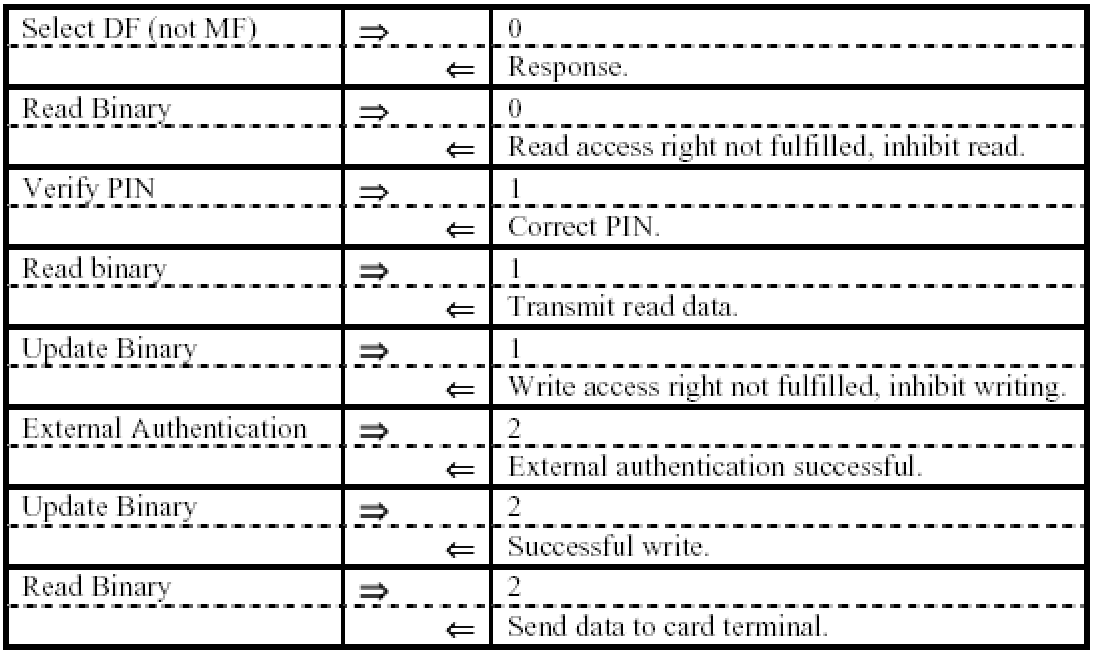
Write access right = F2 ;

DF has a PIN;

After successful PIN verification, the security state is 1;

Card has an external authentication key; use right is 11;

After successful external authentication, the security state is 2 ;



Select DF (not MF)	⇒	0
	⇐	Response.
Read Binary	⇒	0
	⇐	Read access right not fulfilled, inhibit read.
Verify PIN	⇒	1
	⇐	Correct PIN.
Read binary	⇒	1
	⇐	Transmit read data.
Update Binary	⇒	1
	⇐	Write access right not fulfilled, inhibit writing.
External Authentication	⇒	2
	⇐	External authentication successful.
Update Binary	⇒	2
	⇐	Successful write.
Read Binary	⇒	2
	⇐	Send data to card terminal.

Supported Command Sets

SAM Personalization Commands

Command	Description
Get Challenge	For generating an 8-byte random number
External Authentication (Perso)	For authenticating the card transport key.
Create File	For creating DF/EF
Write Key	Add or update card transport key

SAM Management Commands

Command	Description
Initialize Update	For initiating a secure channel.
Get Host Cryptogram	For getting the Host Cryptogram
External Authenticate	For initiating a secure channel.
Put Key	Command to personalize the SAM master keys.
Get Key	To get the Key payload from master SAM card
PIN Change / Un-Block	For changing or unblocking the SAM PIN
Lock / Un-Lock Key	For activating / de-activating a SAM Master key.
Generate Diversification Data	For generating the key diversification data.

Set Card Type / Application state	For setting the card type and changing the application state.
Get Card Info	For retrieving the SAM information.

NCMC Management Commands

Command	Description
Diversify Key	For getting the diversified key payload

Transaction Commands

Command	Description
Verify PIN	To verify the PIN
Get Challenge	For getting an 8-byte random number
Verify Read Purse	For verifying the encrypted Read Purse response.
Compute Credit Crypt	For computing the Credit Cryptogram
Compute Debit Crypt	For computing the Debit Cryptogram
Verify Credit Receipt Crypt	For verifying the Credit receipt cryptogram
Verify Debit Receipt Crypt	For verifying the Debit receipt cryptogram
Compute Atomic Update MAC	For computing MAC to be sent in Atomic Update
Verify Atomic Update MAC	For verifying the MAC received in Atomic Update

All the commands which are listed above are not available in all SAM categories and all Application States. Following table shall define the available commands in different SAM and their required application state.

Command	SAM Type	Allowed Application State	Destination Application State
Initialize Update	Personalization SAM Credit SAM Debit SAM Combo SAM Product SAM	IDLE AUTHENTICATED	IDLE
Get Host Cryptogram	Master SAM	IDLE AUTHENTICATED	AUTHENTICATED
External Authenticate	Personalization SAM Credit SAM Debit SAM Combo SAM Product SAM	IDLE	AUTHENTICATED
Put Key	Personalization SAM Credit SAM Debit SAM Combo SAM Product SAM	AUTHENTICATED	AUTHENTICATED
Get Key	Master SAM	AUTHENTICATED	AUTHENTICATED
PIN Change / Un-Block	Master SAM Personalization SAM Credit SAM Debit SAM Combo SAM	AUTHENTICATED PIN_VERIFIED	AUTHENTICATED PIN_VERIFIED

	Product SAM		
Lock / Un-Lock Key	Master SAM Personalization SAM Credit SAM Debit SAM Combo SAM Product SAM	AUTHENTICATED	AUTHENTICATED
Generate Diversification Data	Personalization SAM Credit SAM Debit SAM Combo SAM Product SAM	AUTHENTICATED	AUTHENTICATED
Set Card Type / Application state	Master SAM Personalization SAM Credit SAM Debit SAM Combo SAM Product SAM	IDLE AUTHENTICATED	IDLE AUTHENTICATED
Get Card Info	Master SAM Personalization SAM Credit SAM Debit SAM Combo SAM Product SAM	IDLE AUTHENTICATED	IDLE AUTHENTICATED
Diversify Key	Personalization SAM	AUTHENTICATED	AUTHENTICATED
Verify PIN	Master SAM Personalization SAM Credit SAM Debit SAM Combo SAM Product SAM	IDLE AUTHENTICATED	IDLE AUTHENTICATED
Get Challenge	Master SAM Personalization SAM Credit SAM Debit SAM Combo SAM Product SAM	IDLE AUTHENTICATED	IDLE
Verify Read Purse	Credit SAM Debit SAM Combo SAM Product SAM	PIN_VERIFIED	PIN_VERIFIED
Compute Credit Crypt	Credit SAM Combo SAM	PIN_VERIFIED	PIN_VERIFIED
Compute Debit Crypt	Debit SAM Combo SAM	PIN_VERIFIED	PIN_VERIFIED
Verify Credit Receipt Crypt	Credit SAM Combo SAM	PIN_VERIFIED	PIN_VERIFIED
Verify Debit Receipt Crypt	Debit SAM Combo SAM	PIN_VERIFIED	PIN_VERIFIED

Compute Atomic Update MAC	Product SAM Combo SAM	PIN_VERIFIED	PIN_VERIFIED
Verify Atomic Update MAC	Product SAM Combo SAM	PIN_VERIFIED	PIN_VERIFIED

If any of the commands fails then the destination state will be IDLE.

All the transaction commands will be available only in the USAGE phase of the card. Also the PIN1 must be verified to execute any of these functional commands. All commands mentioned above are described below in more detail.

External Authentication (Perso)

This command shall be used for authenticating the card transport key and thereby changing the corresponding DF security status. This command can only be executed when the access condition for using external authentication key is fulfilled and the key is not blocked.

Command Format

Parameter	Value
CLA	0x00
INS	0x82
P1	0x00
P2	Transport Key Identifier
Lc	0x08
Data	See below
Le	-

Data Sent

Encrypted 8 byte random data received from the SAM card.

External Authentication Process

External Authentication is the process that card authenticates the external terminal. The process is as follows:

Terminal	Direction	TimeCOS CMC SAM card
Get 8 bytes random number	⇒	Card generates the challenge RNDicc
	⇐	Send random number to terminal

Terminal encrypts the RNDicc using the Cardkey, which is the same as the external authentication key and get the encrypted D1. That is D1=3DES (Cardkey,RNDicc)		
Send D1 for external authentication	⇒ ⇐	Card uses the specific external authentication key to decrypt D1 and get D2.Compare D2 with RNDicc 1) D2=3DES-1(KID,D1) 2) D2? = RNDicc Send the comparison result (SW1SW2) to terminal. If comparison is successful, set the value of the security state register equal to the following status

Explanation:

1. The terminal gets the random number RNDicc
2. The terminal uses the specific key to encrypt RNDicc by 3DES and generates D1
3. The terminal sends the external authentication command to card and sends D1

00 82 00 kid 08 D1

4. After card receives D1, it uses the corresponding key to decrypt D1 by 3DES and generates 8 bytes D2. The card compares RNDicc and D2.

- If it's the same, then external authentication passes. Then the security status will be set to the following status and the reset the error counter.
- If it's not the same, then external authentication fails. The number of error counter decreases one and the security status remains the same.

5.2.8 Application Example

Conditions:

External authentication key ID = 01;

Use access right = 0xF0;

Change access right = 0xEF,

Attempts error counter = 0x33;

Following status = 01;

16 bytes secret key = 57415443484441544154696D65434F53

[Step 1]

Get 8 bytes random number

Command: 00 84 00 00 08

Response: D3 89 BF 67 45 B9 35 50 9000

[Step 2]

Card terminal uses the secret key 57415443484441544154696D65434F53 (which is the same as the external authentication key) to encrypt the random number and the result is C1 8A 5B 4B 13 40 25 21.

[Step 3]

Card terminal sends the encrypted random number to the card to do external authentication.

Command: 00 82 00 00 08 C1 8A 5B 4B 13 40 25 21

Explanation: C1 8A 5B 4B 13 40 25 21 is the encrypted data from [Step 2]

Response: 9000

Explanation: Since it's successful, it sets the security status to be the following status 01.

Response Data

There is no response data for this command.

Response SW

Status Words	Meaning
0x9000	Successful
0x63CX	X remaining attempts left
0x6700	Incorrect length
0x6981	Not an external authenticate key
0x6982	External authenticate key access right of use not fulfilled
0x6983	Authentication (external authentication key) was blocked
0x6A82	Key File not found
0x9302	Error detected during secure messaging
0x9403	Key not found

Create File

This command is used for creating DFs and EFs in TimeCOS SAM card.

Note

- DF and EF can only be created when the create Right is fulfilled for the current DF.
- There is only one Key File under each DF. This Key File must be created first before any other files.
- When the current DF is erased, file creation and accessing is free and not restricted by the Access Right. However, when DF is accessed again after leaving, it must follow the corresponding Access Right.
- Directory file cannot be auto-selected after creation (MF exclusive). Therefore, Select File command must be applied.

Command Format

Parameter	Value
CLA	0x80
INS	0xE0
P1	File ID high byte
P2	File ID Low Byte
Lc	Var
Data	File control information
Le	-

Data Sent

For MF

P1 P2 is set to 3F00

Data	File Type	Space	Create Right	Erase Right	8 byte Transportation Code

Length (byte)	1	2	1	1	8
Value (HEX)	38	FFFF	XX	XX	FFFFFFFFFFFFFFFF

Table FCI of MF

For DF

Data	File Type	Space	Create Right	Erase Right	RFU	DF Name
Length (byte)	1	2	1	1	3	5 - 16
Value (HEX)	38	XXXX	XX	XX	FFFFFF	DF name

Table FCI of DF

For EF

File Type	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**	**B7**
Binary File	28	File Space		Read Right	Write Right	FF	KID, refer to Note[2]
Fix Length Record File	2A	2<=Record no.<=254	Record length <= 178	Read Right	Write Right	FF	KID, refer to Note[2]
Cyclic Record File	2E	2<=Record no.<=254	Record length <= 178	Read Right	Write Right	FF	KID, refer to Note[2]
Variable-length Record File	2C	Space = all the record length +1 byte Checksum (calculated by COS)		Read Right	Write Right	FF	KID, refer to Note[2]
		Each Record length = Record length + 1 byte Checksum (calculated by COS)					

Key File	3F	Space = total key length + 5 reserved bytes For the calculation on each record, please refer to Note[4]	SFI for current DF. Refer to Note[4]	Create Right	FF	FF

Table FCI of EF

For EF When LC = 0Ch

Length	Data Command Coding	Remarks
2	Files ID	
1	Reserved ”0x00”	
1	Initialization Code	
1	File Type 8、2A、2C、2E	
2	File Spacing (Word*Length)	
1	SFI	Note 1
1	KID	Note 2
1	Read Access Rights	
1	Modify Access Rights	
1	Reserved ”0x00”	

If you want use secure message but file not is key file,please use LC = 0Ch

Note:

[1] For Binary file, Fix-length Record File, Variable-length Record File, Cyclic file (except Key File), secure messaging can be applied.

To enable the secure messaging, two MBS of file type are set during file creation.

Byte 1 (File Type) is set as follows:

B7	B6	B5	B4	B3	B2	B1	B0	Secure Messaging

0	0	File Type	None
1	0	File Type	MAC
1	1	File Type	DES & MAC

For example, File type will be changed from 28 to A8 for secure messaging.

[2] Note for KID

Byte 7 is defined as follows:

B7	File effectiveness 1 File effective 0 File not effective (usually not applied)
B6	File Write Position 1 EEPROM, that is the current 32K 0 expanded EEROM, that is the space out of 32K
B5	Atomic Protection 1 Yes 2 No
B4	Read Method 1 Plain Text 0 Encrypted Text
B3	Invert 2 bit for reading KID
B2	
B1	Invert 2 bit for writing KID
B0	

[4] KEY File

Note: SFI for Key file must be 0000

a. Each record length = 1byte TAG + 1byte length + 5 bytes key header + key length

T and L bytes are maintained by COS

Note: For Key file under MF,

Record length = 1 byte TAG + 1byte length + 1 byte Key type

T and L bytes are maintained by COS

b. SFI for DF

SFI for DF is illustrated as follows:

B7	B6	B5	B4	B3	B2	B1	B0	Description
0	0	0	x	x	x	x	x	If the current DF is DDF, the lowest 5 bits of LBS is the SFI
1	0	0	x	x	x	x	x	If the current DF is ADF, the lowest 5bits of LBS is the SFI for the issuer
1	1	0	x	x	x	x	x	It includes the SFI for A5 module of current DF
1	1	1	1	1	1	1	1	RFU

Table SFI for DF

Note: A5 is the record tag for File Control Information

Response Data

There is no response data for this command.

Response SW

Status Words	Meaning
0x9000	Successful
0x6982	Create Right not fulfilled
0x6700	Incorrect length
0x6A80	Record number is less than 2 or number of directory is greater than 3
0x6A84	Not Enough Space
0x6A86	File already exists

Write Key

This command shall be used for writing a new key or updating the existing key.

Note

- When the Append Right is fulfilled for the Key file under current DF, Write Key command can be used to write Key into Key File
- When the Modification Right is fulfilled, key value can be changed
- When the writing Key file , Key Type is Maintenance Key

Command Format

Parameter	Value
CLA	0x80/0x84
INS	0xD4

P1	0x01: For Key unload 0x3X: Key type for key renew
P2	Secret key identifier
Lc	Var
Data	Key header and Key value
Le	-

Data Sent

For Key Upload

Command Data Field = Key header (5 bytes) + key value

Key Type	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Key length**
Maintenance Key	36	Use Right	Modify Right	FF	Error Counter	8/16
Master Key	It's the external authentication key with KID = 00. Its command data field is the same as the external authentication key.					
External Authentication Key	39	Use Right	Modify Right	Following Status	Error Counter	16
Key under MF	3X	There is only one 1 byte of Key Type for this data field For the key installed by this method, its key type and content are the one corresponding to the key under MF				

Table Data Field for Key Upload by Write Key Command

Note: For the Key Version and Following Status, please refer to the Explanation [2]

Explanation:

[1] If there is only one Key of certain type under the Application directory, its KID is 00; else KID should starts from 01. Under one application:

- There is only one Master Key, and its KID must be 00.
- There are maximum 4 Maintenance Key and the KID is 00-03
- KID cannot be FF

[2] Explanation on Technical Terms:

- Use Right

It stands for the right that must be fulfilled before verification, authentication and computation.

For example: If the Use Right is 41, it means the Security Register value must be greater or equal than 1 and less or equal than 4 for using that key.

- Modify Right

It means the right to change the key by Write Key command. When the Modify Right is fulfilled, Write Key command can change the content of key. However, the value of error counter remains.

- Error Counter

The 4 highest bits stands for the maximum allowed consecutive unsuccessful trail. The 4 lowest bits stands for the number of remaining trails. If the number of consecutive unsuccessful trails is greater than the allowed value, the Key will be blocked.

For example: If the Error Counter is 33, it means the maximum unsuccessful key verification is 3. If it fails once, the counter will be 32 and further changed to 31 if it fails again. If the next verification or authentication is correct, the counter will change to 33. For a successful Unblock Key command, the 4 highest bits will be set to the same value as the 4 lowest bits. At the same time, the key value is changed. If unsuccessful, the number of allowed trails decreases by 1. The card will be permanently blocked if the unblock PIN and External Authentication Key is blocked.

- Following Status

After a successful verification or external authentication, the Security Register is set to the same value as the 4 lowest bits of the following status.

- Unblock KID

For a successful unblock Key command, the Key specified KID is unblocked.

- Key version and the Algo Tag are defined by users.

Key Modification

Command Data Field = New Key Value

- If the Modify Right fulfills, Write Key command can change the key value. However, the value of error counter remains.

- It does not applicable when the key is blocked.

Response Data

There is no response data for this command.

Response SW

Status Words	Meaning
0x9000	Successful
0x6982	Modify Right or Append right not fulfilled.
0x6700	Incorrect length
0x6983	Key is blocked
0x6A82	Key File not found
0x6A83	Key not found
0x6A84	Not enough space in the key EF
0x9302	Error in Secure Messaging when modify key

Initialize Update

This is the command which shall initialize a secure channel between the Master SAM and Other SAMs. This command shall only be accepted by the Other SAM cards. Once received this command shall clear the previous secure channel session if established before.

Command Format

Parameter	Value
CLA	0x80
INS	0x50
P1	0x00
P2	0x00
Lc	0x08
Data	Master SAM challenge
Le	0x00

Data Sent

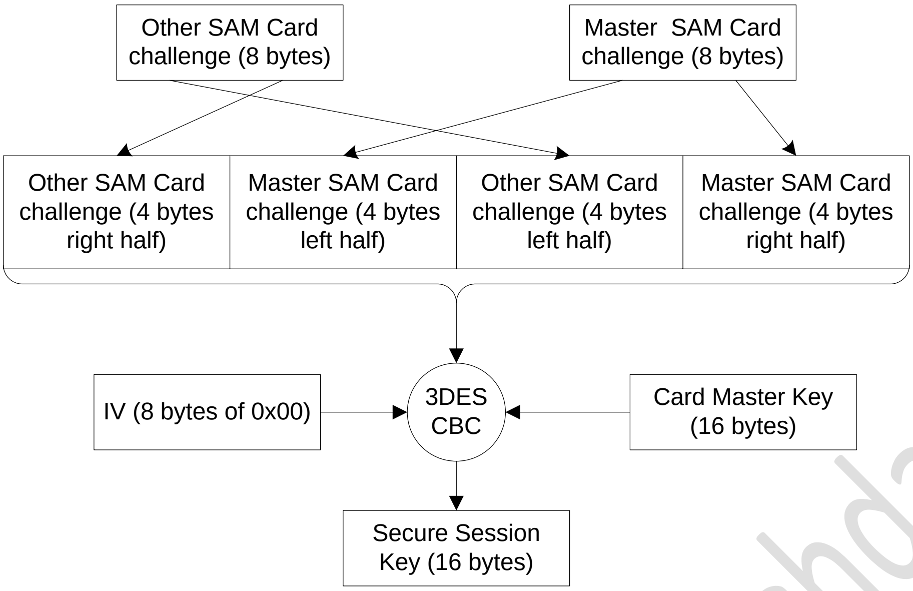
8 byte random data received from the Master SAM card.

Command Processing

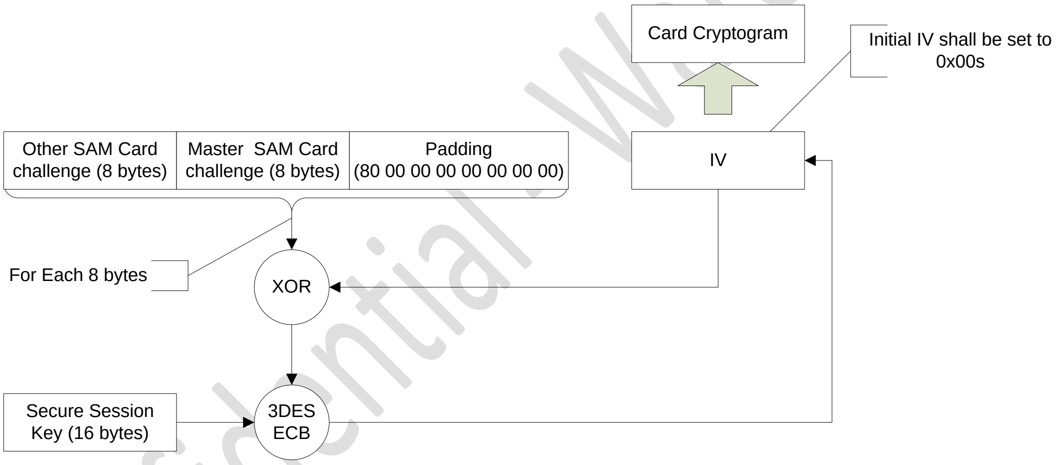
Following are the steps:

- If P1 P2 values are not in accordance with the above table then return error status 0x6A86.
- If Lc value is not 0x08 then return error status 0x6700.
- If Master SAM card then return error status 0x6985.
- If the card is in state PERSO then return error status 0x6985.
- Card shall generate the secure session key and card cryptogram as depicted below and shall return the same.

Session Key Derivation


Other SAM Card challenge (8 bytes)
Master SAM Card challenge (8 bytes)
Other SAM Card challenge (4 bytes right half)
Master SAM Card challenge (4 bytes left half)
Other SAM Card challenge (4 bytes left half)
Master SAM Card challenge (4 bytes right half)
IV (8 bytes of 0x00)
3DES CBC
Card Master Key (16 bytes)
Secure Session Key (16 bytes)

Card Cryptogram Calculation


Card Cryptogram
Initial IV shall be set to 0x00s
Other SAM Card challenge (8 bytes)
Master SAM Card challenge (8 bytes)
Padding (80 00 00 00 00 00 00 00)
IV
For Each 8 bytes
XOR
Secure Session Key (16 bytes)
3DES ECB

Response Data

Card shall return the following as response data.

Card Challenge	Card Cryptogram
8 bytes	8 bytes

Response SW

Status Words	Meaning
9000	Successful execution
6A86	Wrong P1 or P2
6700	Wrong Lc
6985	Wrong Card type Command not allowed
6983	Master key blocked

Get Host Cryptogram

This command shall verify the card cryptogram received in Initialize Update command and subsequently generate the host cryptogram. Only Master SAM card shall support this.

Command Format

Parameter	Value
CLA	0x80
INS	0x86
P1	0x00
P2	0x00
Lc	0x10
Data	See below
Le	-

Data Sent

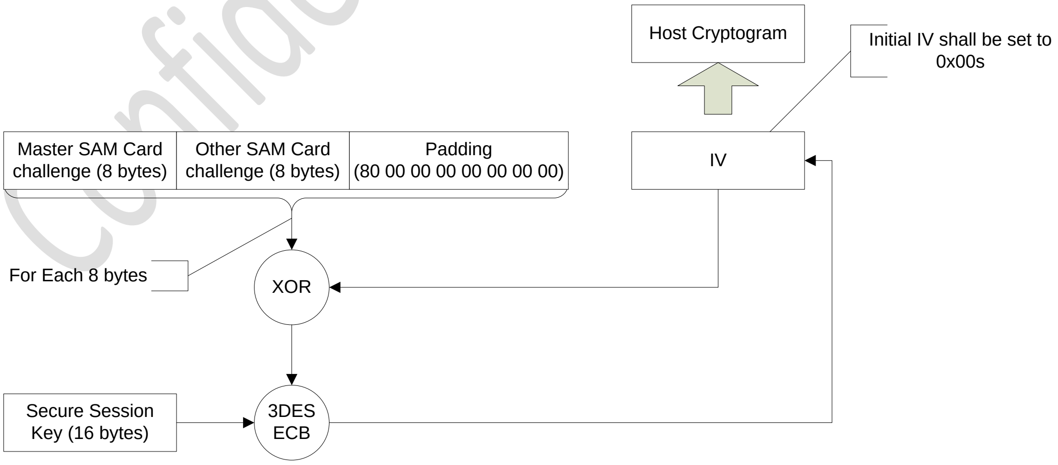
Response data received from the Initialize Update command.

Command Processing

Following are the steps:

- If P1 P2 are not in accordance to the above table then return error status 0x6A86.
- If Lc is not 0x10 then return error status 0x6700.
- Generate the Secure Session key as depicted in Session Key Derivation section.
- Calculate the Card Cryptogram as depicted in Card Cryptogram Calculation section.
- Compare the calculated card cryptogram with that received in the command data. If doesn't match then increment the retry counter of Master Key and return error status 0x63Cx.
- Calculate the Host Cryptogram as depicted below and return the same.

Host Cryptogram Calculation


Host Cryptogram
Initial IV shall be set to 0x00s
Master SAM Card challenge (8 bytes)
Other SAM Card challenge (8 bytes)
Padding (80 00 00 00 00 00 00 00)
IV
For Each 8 bytes
XOR
Secure Session Key (16 bytes)
3DES ECB

Response SW

Status Words	Meaning
9000	Successful execution
6A86	Wrong P1 or P2

6700	Wrong Lc
6985	Wrong Card type Command not allowed
6983	Master key blocked
63CX	Wrong cryptogram. x tried remaining for the key in use.

External Authenticate

External Authenticate command shall be used by Other SAM cards to verify the host cryptogram received from the Master SAM card.

Command Format

Parameter	Value
CLA	0x80
INS	0x82
P1	0x00
P2	0x00
Lc	0x08
Data	See below
Le	-

Data Sent

Host Cryptogram which has been received in response to Get Host Cryptogram command.

Command Processing

Following are the steps:

- If P1 P2 is not in accordance to the above table then return error status 0x6A86.
- If Lc is not 0x08 then return error status 0x6700.
- If no prior successful Initialize Update Command then return error status 0x6985.
- Calculate the host cryptogram as depicted in Host Cryptogram Calculation section.
- Compare the calculated host cryptogram with the received one. If doesn't match then increment the Master Key retry counter and return error status 0x63Cx.
- If the cryptograms match then return reset the master key retry counter and exit.

Response SW

Status Words	Meaning
9000	Successful execution
6A86	Wrong P1 or P2
6700	Wrong Lc
6985	Wrong Card type Command not allowed
6983	Master key blocked
63CX	Wrong cryptogram. x tried remaining for the key in use.

Put Key

This is a command to securely personalize the SAM master keys into all the SAM cards. This command shall only be available during the PERSO state of the card.

Command Format

Parameter	Value
CLA	0x80
INS	0x5C
P1	BITS 0~4: Key EF SFI BITS 5~6: 0b BIT 7: 0 for Other SAM 1 for Master SAM
P2	Key Number
Lc	0x24 / 0x1E
Data	See below
Le	-

Data Sent

Encrypted Key Data as received from the Master SAM card in response to Get Key command.

Command Processing

Command shall be processed as follows:

- If P1 value is not as per the above table then return error status 0x6A86.
- If Other SAM card and BIT 7 of P1 is 0x01b or if Master SAM card and BIT 7 of P1 is 0x00b then return error status 0x6985.
- If the SFI mentioned in P1 can't be found then return error status 0x6A82.
- If Lc value is not as per above table then return error status 0x6700.
- If Other SAM card and Lc value is 0x1E or if Master SAM card and Lc value is 0x24 then return error status 0x6A80.
- If Other SAM then verify the command data MAC using the algorithm shown in Command Data MAC Calculation. If doesn't match then return error status 0x6982.
- Decrypt the data received using the Secure Session key.
- Check the key header and if not correct return error status 0x6A80.
- Store the key and its related attributes.

Response Data

There is no response data for this command.

Response SW

Status Words	Meaning
9000	Successful execution
6A86	Wrong P1 or P2
6700	Wrong Lc
6985	Wrong Card type Session key not available

	Command not allowed

Get Key

Master SAM card shall derive and prepare the Key payloads for all other SAM cards. This is the command which shall be used to initiate the process and shall be available during the USAGE phase of the card.

Command Format

Parameter	Value
CLA	0x80
INS	0x5E
P1	Master Key EF SFI
P2	Master Key Number
Lc	0x00 / 0x03 / 0x18 / 0x1B
Data	See below
Le	0x20

Data Sent

Maximum Amount (Optional)	Key Diversification Data (Optional)
3-bytes	24-bytes

Key Diversification Data

If provided this data shall be used in the derivation of the key and if not provided then a set of 24 bytes of 0x00s shall be used as the derivation data.

Maximum Amount

If provided this data shall be added to the header portion of the derived key. And if not provided then the Maximum Amount data field of the Master Key shall be copied.

Command Processing

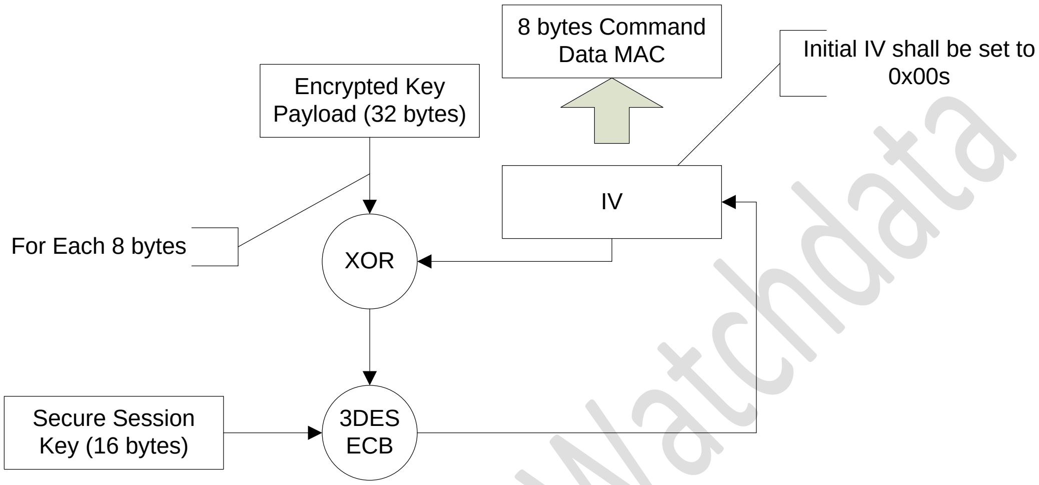
Command shall be processed as follows:

- If P1 is 0x00 or greater than 0x1F or if P2 is 0x00 or 0xFF then return error status 0x6A86.
- If Lc is anything other than 0x00, 0x03, 0x18, 0x1B then return error status 0x6700.
- If there is no Session Key available then return error status 0x6985.
- If the SFI provided in the P1 can't be found in the Master SAM card then return error status 0x6A82.
- If the Key Number provided in the P2 can't be found in the SFI specified then return error status 0x6A80.
- If Maximum Amount field has been provided and its not a positive value then return error status 0x6A80.
- Using the key referred in P1P2 encrypt the Key Diversification Data (If provided) or 24 bytes of 0x00s to generate the Key.
- Get the Key Header of the Master key (referred in P1 P2 bytes). Replace the Maximum Amount filed in this header if provided in the command data.

Concatenate this header with the Key derived above to form the key record. Pad this key record with 0x00 to make it multiple of 8 bytes and then encrypt using the Secure Session Key.

- Calculate MAC over the Encrypted Key payload using the algorithm as shown in Command data MAC Calculation. Concatenate 4MSB bytes of this MAC with the encrypted key payload and return.

Command Data MAC Calculation


8 bytes Command Data MAC
Initial IV shall be set to 0x00s
Encrypted Key Payload (32 bytes)
IV
For Each 8 bytes
XOR
Secure Session Key (16 bytes)
3DES ECB

Response Data

Encrypted Key Payload and MAC.

Response SW

Status Words	Meaning
9000	Successful execution
6A86	Wrong P1 or P2
6700	Wrong Lc
6985	Wrong Card type Session key not available Command not allowed
6A82	Master Key EF not found
6A80	Master Key not found.

PIN Change / Un-Block

Every SAM card shall have multiple PINs to be used for different purpose. Using this command the value of these PINs can be set, changed or can be unblocked if it’s already blocked. PIN set option shall only be available in PERSO life cycle state of the card and PIN Change and Unblock shall be available during card life cycle state USAGE.

Command Format

Parameter	Value
CLA	0x80
INS	0x62
P1	Unblock: 0x00

	Change: 0x01 Set: 0x02
P2	PIN Number
Lc	Var
Data	See below
Le	-

Data Sent

PIN block as shown below which shall be encrypted using the Card Transport Key. 4 MSB bytes of command data MAC shall also be appended with this encrypted PIN block. This MAC shall be calculated using the Card Transport Key following Command MAC Calculation algorithm.

PIN Length	Counters		PIN Value
	Retry Counter	Max Retry Counter	
1 byte	1 byte		4~8 bytes

Command Processing

Command shall be processed as follows:

- If P1 is anything other than as specified or P2 is 0xFF then return error status 0x6A86.
- If P1 is 0x00 and Lc is not 0x00 or if P1 is 0x01 or 0x02 and Lc is 0x00 then return error status 0x6A80.
- If P1 is not 0x02 and the PIN number provided in P2 can’t be found then return error status 0x6A80.
- If Mutual Authentication hasn’t been performed successfully prior to this command then return error status 0x6985.
- The specified PIN must be verified before if it’s a PIN change command. If not verified then return error status 0x6982.
- For PIN Change and Verify option the command data MAC shall be verified. If not correct an error status of 0x6982 shall be returned.
- Decrypt the received data using the Card Transport Key.
- If the PIN length in the decrypted data is not within the defined range then return error status 0x6A80.
- If it’s a PIN unblock command then set the retry counter to Max Retry Counter or if it’s a PIN change command then change the PIN record.

Response Data

There is no response data for this command.

Response SW

Status Words	Meaning
9000	Successful execution
6A86	Wrong P1 or P2
6700	Wrong Lc
6985	Wrong Card type

	Session key not available Command not allowed
6982	PIN not verified
6A80	PIN format not correct PIN not found

Lock / Un-Lock Key

The keys inside the SAM card might get blocked due to wrong usage. So, to unlock such keys this command shall be used. Also using this command any particular key can be blocked as well.

Command Format

Parameter	Value
CLA	0x80
INS	0x64
P1	Lock: 0x00 Unlock: 0x01
P2	0x00
Lc	0x02
Data	See below
Le	-

Data Sent

SAM Key EF SFI	SAM Key Number
1-byte	1-byte

Command Processing

Command shall be processed as follows:

- If P1 P2 values are anything other than the specified ones then return error status 0x6A86.
- If Lc value is not 0x02 then return error status 0x6700.
- If Mutual Authentication hasn't been performed successfully before this command then return error status 0x6985.
- If the specified Key EF can't be found then return error status 0x6A82 and if the specified Key number can't be found then return error status 0x6A80.
- If P1 is 0x00 and the specified Key is already Locked or if P1 is 0x01 and the specified key is already unlocked then return error status 0x6985.
- Lock or unlock the specified key according to the option and return.

Response Data

There is no response data for this command.

Response SW

Status Words	Meaning
9000	Successful execution
6A86	Wrong P1 or P2
6700	Wrong Lc

6985	Wrong Card type Session key not available Command not allowed
0x6A82	Key EF not found
0x6A80	Key not found

Set Diversification Data

Every SAM shall have a key diversification data associated with it. This data shall be used while deriving the NCMC keys. Every key in the SAM card shall have a pointer to point to any of this data element.

Command Format

Parameter	Value
CLA	0x80
INS	0x66
P1	0x00: Master SAM 0x01: Other SAM
P2	0x00
Lc	0x08
Data	See below
Le	-

Data Sent

In case of Master SAM following data shall be sent in plain. And in case of any other SAM following data shall be encrypted using the Secure Session Key.

Diversification Data Identifier	Algorithm Identifier	Diversification Data
1-byte	1-byte	6-bytes

Command Processing

Command shall be processed as follows:

- If P1 or P2 has any values other than 0x00 then return error status 0x6A86.
- If Lc value is not 0x08 then return error status 0x6700.
- If Mutual Authentication hasn't been performed successfully before then return error status 0x6985.If the Diversification Data Identifier provided in the data field already exist in the card then return error 0x6A80.
- If the algorithm identifier provided in the data field doesn't correspond to valid algorithm as specified in this document then return error status 0x6A80.
- If Master SAM Card then save the data as it is and if Other SAM card then decrypt the data using the secure session key and then store it.

Response Data

There is no response data for this command.

Response SW

Status Words	Meaning
9000	Successful execution

6A86	Wrong P1 or P2
6700	Wrong Lc
6985	Wrong Card type Session key not available Command not allowed
0x6A80	Wrong algorithm identifier

Get Diversification Data

This command shall be used to get the encrypted Key Diversification data payload from the MasterSAM card. Apart from the Master SAM card no other SAM card shall support this command.

Command Format

Parameter	Value
CLA	0x80
INS	0x6E
P1	SAM MasterKey EF SFI
P2	SAM Master Key No
Lc	-
Data	-
Le	0x08

Data Sent

No data shall be sent in this command

Command Processing

Command shall be processed as follows:

- If P1 is greater than 0x1F or P2 is 0xFF then return error status 0x6A86.
- If Le is not 0x08 then return error status 0x6700.
- If not Master SAM card then return error 0x6985.
- If the Card Life Cycle state is not USAGE then return error status 0x6985.
- If Mutual Authentication hasn’t been performed before this command then return error status 0x6985.
- If the Key EF referred in P1 can’t be found then return error status 0x6A82.
- If the Key number referred in P2 can’t be found then return error status 0x6A80.
- Get the Diversification Identifier from the referred key header and search for it. If not found then return error status 0x6A80.
- Encrypt the whole diversification data using the secure session key and return the encrypted block.

Response Data

Encrypted Diversification data block.

Response SW

Status Words	Meaning
9000	Successful execution
6A86	Wrong P1 or P2

6700	Wrong Lc
6985	Wrong Card type Session key not available Command not allowed
0x6A80	Wrong Key reference Wrong Diversification data indicator.

Set Card Type / Application state

This command shall set the different card types and application states. Card type can be set only once during the card life time.

Command Format

Parameter	Value
CLA	0x80
INS	0x68
P1	0x00 /0x01
P2	0x00
Lc	0x01
Data	See below
Le	-

Data Sent

1 byte data shall be sent. If P1 is 0x00 then the data field shall contain the card type and if it’s 0x01 then the data field shall contain the Application State.

Card type table

SAM	Value
Master SAM	0x01
Personalization SAM	0x02
Credit SAM	0x03
Debit SAM	0x04
Product SAM	0x05
Combo SAM	0x06

Application State table

Life Cycle State	Value
MANUFACTURED	0x01
PERSO	0x02
USAGE	0x03
BLOCKED	0x04

Application state values can only be changed upward. That means changing of state from PERSO to USAGE is allowed but from USAGE to PERSO is not allowed.

Command Processing

Once card receives this command it shall perform following steps:

- If P1 P2 is not as per the above table then return error status 0x6A86.
- If Lc is not 0x01 then return error status 0x6700.

- If P1 is 0x01 and the Master key hasn’t been authenticated then return error status 0x6985.
- If P1 is 0x00 and the card type has already been set before then return error status 0x6985.
- If application state change is from upper value to lower then return error status 0x6985.
- Set the value and return.

Response Data

There is no response data for this command.

Response SW

Status Words	Meaning
9000	Successful execution
6A86	Wrong P1 or P2
6700	Wrong Lc
6985	Wrong Card type Session key not available Command not allowed

Get Card Info

This command shall be used to read the SAM related information.

Command Format

Parameter	Value
CLA	0x80
INS	0x6A
P1	0x00
P2	0x00
Lc	-
Data	-
Le	0x00

Data Sent

No command data.

Command Processing

Once card receives this command it shall perform following steps:

- If P1 P2 is not 0x00 then return error status 0x6A86.
- If Le is not 0x00 then return error status 0x6700.
- If the application is not in USAGE phase then return error status 0x6985.
- Prepare the response data as shown below and return.

Response Data

SAM Type	Usage Counter	SAM Card TRP	Key Diversification Info	
			Diversification Data Identifier	Algorithm Identifier

1-byte	4-bytes	4-bytes	1-byte	1-byte

The SAM Card TRP and the Key Diversification information fields shall be present in response data only when PIN2 has been verified prior to this command. The SAM card TRP and the Transaction Usage counter shall be present only if this data elements has been personalized.

In the Key Diversification Info field all the available Diversification Data information shall be returned from the card.

Response SW

Status Words	Meaning
9000	Successful execution
6A86	Wrong P1 or P2
6700	Wrong Le
6985	Wrong Application state

Set Txn Counter / SAM Card TRP

Every SAM shall store a particular Terminal Reference Parameter. This TRP value shall be checked during the transaction commands. Also each SAM shall maintain a Transaction Counter which will count the number of transactions performed by this it. Using this command the initial values to these two parameters can be set. Once set none of these values can be changed again.

SAM Txn Counter and SAM Card TRP shall only be checked during transaction commands if set.

SAM Txn Counter is a down counter whose maximum value shall be set during personalization and shall be decremented by 1 in every transaction. If its value reaches 0x00 then no further transaction commands are allowed and the card shall be blocked.

The values which are set using this command shall be effective from the next power up session of the card.

Command Format

Parameter	Value
CLA	0x80
INS	0x6C
P1	0x00: Set Txn Counter 0x01: Set SAM Card TRP
P2	0x00
Lc	0x04
Data	See below
Le	-

Data Sent

If P1 is 0x00 then the data field shall contain the maximum value of SAM Txn Counter and if P1 is 0x01 then the data field shall contain the SAM Card TRP.

Command Processing

Once card receives this command it shall perform following steps:

- If P1 P2 is not as per table above then return error status 0x6A86.
- If Lc value is not 0x04 then return error status 0x6700.
- If Card Life Cycle State is PERSO and Mutual Authentication hasn't been performed before this command then return error status 0x6985.
- If Card Life Cycle State is USAGE and PIN2 hasn't been verified before this command then return error status 0x6985.
- If P1 is 0x00 and the Card Life Cycle state is not PERSO then return error status 0x6985.
- If P1 is 0x01 and the Card Life Cycle State is not PERSO or USAGE then return error status 0x6985.
- If P1 is 0x01 and the SAM Card TRP has already been set once then return error status 0x6985.
- If the data provided are all 0xFFs then return error status 0x6A80.
- Store the data and return.

Response Data

There shall be no response data for this command.

Response SW

Status Words	Meaning
9000	Successful execution
6A86	Wrong P1 or P2
6700	Wrong Le
6985	Wrong Application state SAM Card TRP already set No Mutual Authentication
6A80	Wrong command data

Verify PIN

Each SAM card shall have multiple PINs within it. This command shall be used to verify such PIN.

Command Format

Parameter	Value
CLA	0x80
INS	0x20
P1	0x00
P2	PIN Number
Lc	0x08
Data	See below
Le	-

Data Sent

The encrypted PIN block. This shall be obtained by encrypting the PIN using itself as a key. If the PIN is not multiple of 8 bytes then pad it with 0x00.

Command Processing

Once card receives this command it shall perform following steps:

- If P1 is not 0x00 or P2 is either 0x00 or 0xFF then return error status 0x6A86.
- If Lc is not 0x08 then return error status 0x6700.
- If the PIN number specified in P2 can't be found then return error status 0x6A80.
- If the PIN is blocked then return error status 0x6983.
- If the specified PIN is not 8 bytes long then pad it with 0x00s and then use this as a key to decrypt the incoming data.
- Compare the PIN value from the decrypted data with the reference PIN. If doesn't match then increment the retry counter and return error status 63Cx, where x signifies the remaining retry count.
- If the PIN value matches then set the status of the PIN as verified and exit. Also reset the retry counter.

Response Data

There is no response data for this command.

Response SW

Status Words	Meaning
9000	Successful execution
6A86	Wrong P1 or P2
6700	Wrong Lc
6985	Command not allowed
6A80	PIN not found
63Cx	Wrong PIN. X signifies the number of retries left.
6983	PIN blocked

Verify Read Purse

This command shall be used to verify the encrypted response received in Read Purse Secure command. Also this shall derive the session key which can be used later in Atomic update commands.

Command Format

Parameter	Value
CLA	0x80
INS	0x56
P1	SAM Master Key EF SFI
P2	SAM Master Key number
Lc	Var
Data	See below

Le	-

Data Sent

CMC Key EF SFI	CMC Key EF No	CMC Card Challenge	Read Purse Response
1-byte	1-byte	8-bytes	var

Command Processing

Once card receives this command it shall perform following steps:

- Extract the Master key no and select the appropriate key. If not found throw error 0x6A80.
- Check the validity of the Master key. If not valid throw error 0x6985.
- Check the retry counter of the Master key against the max retry counter. If exceeded throw error 0x6983.
- Check the Master Key attribute. If none of the Debit, Auto-Topup or Signature attribute is not set then
    - Increment the retry counter of the key. If the retry exceeds the max allowed times then invalidate the key.
    - Throw error 0x6985.
- Using the CAN, CSN, Key EF SFI and Key No from the command APDU derive the diversified key as shown in section Key Diversification.
- Check the availability of the challenge. If not available the throw error 0x6985.
- Derive the Read Purse session key by using the previous challenge, card random and CSN from command data. Also save this session key temporarily for later usage.
- Extract the encrypted “Last Transaction Signed Certificate” and “Counter Data” from the command data.
- Decrypt the encrypted data using the Read Purse Session key. Save the “Last Transaction Signed Certificate” and “Counter Data” from the decrypted data for later usage. Also save the “Last Transaction record” and “Debit Option”.
- Invalidate the challenge and return SW9000.

Response Data

There is no response data for this command.

Response SW

Status Words	Meaning
9000	Successful execution
6A80	Master Key not found
6985	Master Key not valid Key not valid for this operation Challenge not available
6983	Key retry exceeded

Compute Credit Crypt

This command shall be used to prepare the Credit Cryptogram to be sent to CMC card. SAM shall prepare the whole Credit APDU and return in response to this command.

Command Format

Parameter	Value
CLA	0x80
INS	0x58
P1	0x00
P2	0x00
Lc	0x29
Data	See below
Le	0x2A

Data Sent

SAM Master Keys						CMC Credit Keys				TRP	Pf	Sign Key		Txn Header	Txn User Data	CMC Card Challenge
SCKf1	SCKn1	SCKf2	SCKn2	SSKf	SSKn	CKf1	CKn1	CKf2	CKn2			SKf	SKn			
1-byte	1-byte	1-byte	1-byte	1-byte	1-byte	1-byte	1-byte	1-byte	1-byte	4-bytes	1-byte	1-byte	1-byte	8-bytes	8-bytes	8-bytes

Command Processing

Once the card receives this command it shall perform the following steps:

- Check P1P2. If not 0x00 throw error 0x6A86.
- Check Lc. If not equals to 0x29 throw error 0x6700.
- Check Le. If not equals to 0x2A throw error 0x6700. Not applicable if card supports only T0 protocol.
- Check if Verify Read Purse has been executed during this transaction session. If not then throw error 0x6985.
- Check the availability of SAM card challenge. If not available then throw error 0x6985.
- Find the SAM Master Key EF SCKf1. If not found throw error 0x6A82.
- Check validity of SCKn1. If out of range throw error 0x6A80.
- Check the master key attribute. If it doesn't support Credit then:
    - Increment the retry counter of the key. If the retry exceeds the max allowed times then invalidate the key.
    - Throw error 0x6985.
- If the Txn Amount provided in the Txn Header is greater than the Txn Amount Limit for SCKn1 then return 0x6A80.
- Find the SAM Master Key EF SCKf2. If not found throw error 0x6A82.
- Check validity of SCKn2. If out of range throw error 0x6A80.
- Check the master key attribute. If it doesn't support Credit then:
    - Increment the retry counter of the key. If the retry exceeds the max allowed times then invalidate the key.
    - Throw error 0x6985.
- Find the SAM Master Key EF SSKf. If not found throw error 0x6A82.

- Check validity of SSKn. If out of range throw error 0x6A80.
- Check the master key attribute. If it doesn’t support Signature then:
  - Increment the retry counter of the key. If the retry exceeds the max allowed times then invalidate the key.
  - Throw error 0x6985.
- Save this SSKf and SSKn for later usage.
- Calculate CRC16 over Credit Record Data which consists of TRP, Pf, SKf, SKn. Txn Header and Txn User data in that order. The CRC should be ISO / IEC 1444-3 CRC_B checksum.
- Derive the diversified Credit Key#2 from the SAM master key reference (SCKf2, SCKn2) as depicted in Key Diversification section.
- Use this diversified Credit Key #2 to encrypt the Credit record CRC, SKf, SKn, Txn Type and Txn Amount in that order. This encrypted data shall be called as Encrypted Credit Parameter Block.
- Derive the diversified Credit Key#1 from the SAM master key reference (SCKf1, SCKn1) as depicted in Key Diversification section.
- Derive the credit session key by encrypting the CMC Card Random, SAM Card Challenge and CMC CSN in that order using the Diversified Credit key#1.
- Using this Credit Session Key encrypt the data TRP, Encrypted Credit Parameter Block and Txn Date and Time in that order to generate the Credit Cryptogram.
- Store the TRP, Txn Header and Sign Key Reference from the command data for later usage.
- Invalidate the SAM challenge.
- Format the CMC Credit command data and return those as a response data.

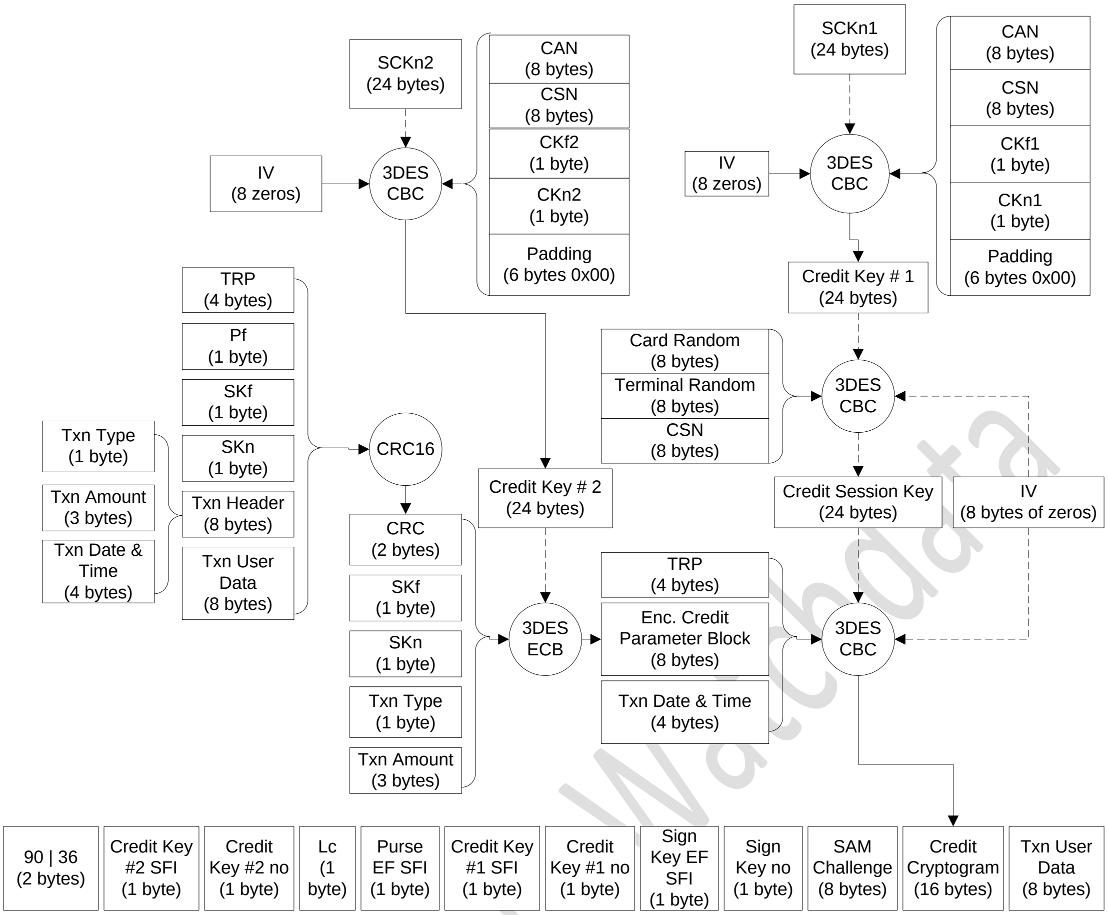


Figure: Compute Credit Crypt Command Flow.

Response Data

In response to this command the SAM card shall return the complete formatted credit APDU. The returned response from this command can be directly sent to the CMC card.

Response SW

Status Words	Meaning
9000	Successful execution
6A80	Master Key not found
6985	Master Key not valid Key not valid for this operation SAM Challenge not available No Preceding Read Purse
6983	Key retry exceeded
6A86	Invalid P1 P2
6700	Wrong Lc or Le
6A82	Master Key EF not found

Verify Credit Receipt Crypt

This command shall be used to verify the “Receipt Cryptogram” received from the CMC card as a response to the Credit command. If the receipt cryptogram is not valid then the transaction shall be aborted.

Command Format

Parameter	Value
CLA	0x80
INS	0x5A
P1	0x00 / 0x01
P2	0x00
Lc	0x16
Data	Credit Receipt Cryptogram
Le	-

Command Processing

After receiving this command the card shall perform the following steps:

- Check the P1P2. If anything other than 0x0000 or 0x0100 return error 0x6A86.
- Check the Lc. If not 0x16 return error 0x6700.
- Derive the Signature Key from the saved SSKf and SSKn by following the method described in Key Diversification.
- Compute the Purse Balance. Base Purse balance is the one received in Verify Read Purse command and after the modifications done on it by various transaction related commands during the session. Add the amount received in Compute Credit Crypt command if P1 is 0x00.
- Increment the Counter Values. Base values are the ones received in Verify Read Purse command.
    - Increment PTC by 1.
    - If P1 is 0x00 then increment the Add-Value Counter by 1.
    - If P1 is 0x01 then increment the Modify Purse Counter by 1.
- Decrypt the Credit Receipt Cryptogram using the Credit Session key which has been derived in Compute Credit Crypt command. Use the updated Counter data as Initial Vector for this decryption process.
- Verify the first 3-bytes of the decrypted data against the current Purse Balance. If doesn't match throw error 0x6982.
- Verify the last 8-bytes of the decrypted data against the updated Counter data. If doesn't match then throw error 0x6982.
- Derive the Signing session key by encrypting the Debit Options, Purse Balance, TRP, Counter data and CSN in that order using the diversified Signature Key.
- Decrypt the received Signed Certificate from the above the decrypted data using the Signing session key.
- Compare the decrypted data against the last saved Txn Header. If doesn't match then throw error 0x6982.
- Return 0x9000.

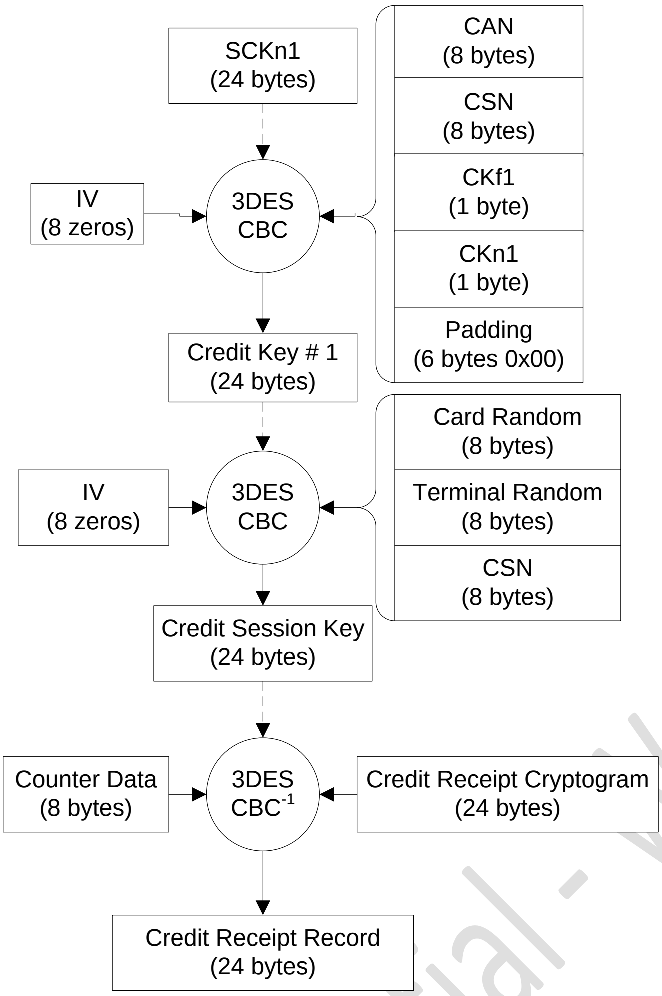


Figure: Verify Credit Receipt Cryptogram

Response Data

These will be no response data for this command.

Response SW

Status Words	Meaning
9000	Successful execution
6A80	Master Key not found
6985	Master Key not valid Key not valid for this operation SAM Challenge not available No Preceding Read Purse
6983	Key retry exceeded
6A86	Invalid P1 P2
6700	Wrong Lc or Le
6A82	Master Key EF not found
6982	Security conditions not satisfied

Compute Debit Crypt

This command shall be used to compute the Debit cryptogram as expected by the CMC card. The received response from this command can be directly sent to the CMC card for Debit Command processing.

Command Format

Parameter	Value
CLA	0x80
INS	0x5B
P1	0x00 / 0x01
P2	0x00
Lc	0x26
Data	See below
Le	0x2A

Data Sent

SAM Master Keys				CMC Keys									
SDKf	SDKn	SSKf	SSKn	DKf	DKn	TRP	Debit Options	Pf	SKf	SKn	Txn Header	Txn User data	CMC Challenge
1-byte	1-byte	1-byte	1-byte	1-byte	1-byte	4-bytes	1-byte	1-byte	1-byte	1-byte	8-bytes	8-bytes	8-bytes

Command Processing

After receiving the command the card shall perform the following steps:

- Check P1P2. If anything other than 0x0000 or 0x0100 return error 0x6A86.
- Check if Verify Read Purse has been executed during this transaction session. If not then throw error 0x6985.
- Check availability of SAM challenge. If not available then throw error 0x6985.
- Find the SAM Master Key EF SDKf. If not found throw error 0x6A82.
- Check validity of SDKn. If out of range throw error 0x6A80.
- Check the master key attribute. If P1 is 0x01 and the key doesn't support Auto-Topup key derivation or if P1 is 0x00 and it doesn't support Debit / Debit with Auto-Topup / Signature then:
    - Increment the retry counter of the key. If the retry exceeds the max allowed times then invalidate the key.
    - Throw error 0x6985.
- If the Txn Amount provided in the Txn Header is greater than the Txn Amount Limit using SDKn key then return error status 0x6A80.
- Find the SAM Master Key EF SSKf. If not found throw error 0x6A82.
- Check validity of SSKn. If out of range throw error 0x6A80.
- Check the master key attribute. If it doesn't support Signature then:

  - Increment the retry counter of the key. If the retry exceeds the max allowed times then invalidate the key.
  - Throw error 0x6985.
- Save this SSKf and SSKn for later usage.
- Calculate the CRC_16 over the TRP, Debit Options, Pf, SKf, SKn, Txn Header and Txn User Data in that order. The CRC here is ISO / IEC 1444-3 CRC_B checksum.
- Compute the derived Debit key from the SDKf and SDKn using the derivation method depicted in section Key Diversification.
- Calculate the Debit Session Key by encrypting the SAM challenge, CMC Card Challenge and CMC CSN in that order.
- Calculate the Debit Cryptogram by encrypting the TRP, CRC, SKf, SKn and Txn Header in that order by using the Debit Session key.
- Prepare the response data exactly in the form as expected in CMC Debit command.
- Invalidate the SAM challenge.
- Store the Counter Data and Txn Header for later usage.
- Return the response data and SW9000.

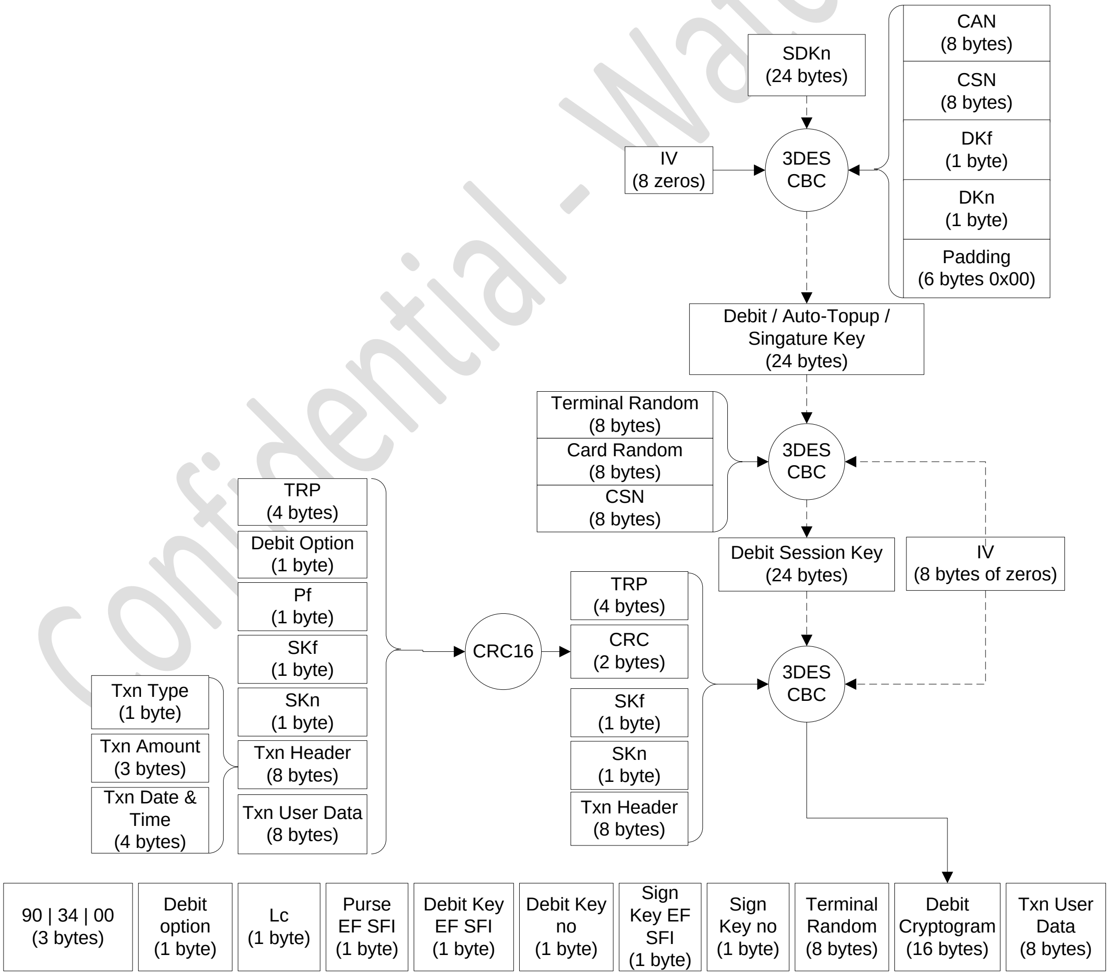


Figure: Compute Debit Crypt Processing

Response Data

As a response to this command the card shall reply the complete APDU block of Debit command which can be directly sent to the CMC card for processing.

Response SW

Status Words	Meaning
9000	Successful execution
6A80	Master Key not found
6985	Master Key not valid Key not valid for this operation SAM Challenge not available No Preceding Read Purse
6983	Key retry exceeded
6A86	Invalid P1 P2
6700	Wrong Lc or Le
6A82	Master Key EF not found

Verify Debit Receipt Crypt

This command shall be used to verify the Debit receipt Cryptogram received from CMC card as a response to CMC Debit command. If the Receipt Cryptogram can't be verified by this command then the transaction shall be aborted.

Command Format

Parameter	Value
CLA	0x80
INS	0x5D
P1	0x00 / 0x01
P2	0x00
Lc	0x16
Data	Debit Receipt Cryptogram
Le	-

Command Processing

After receiving this command the card shall perform the following steps:

- Check the P1P2. If anything other than 0x0000 or 0x0100 return error 0x6A86.
- Check the Lc. If not 0x16 return error 0x6700.
- Check the previous command. If not Compute Debit Crypt then return error 0x6985.
- Derive the Signature Key from the saved SSKf and SSKn by following the method described in Key Diversification.
- Compute the Purse Balance. Base Purse balance is the one received in Verify Read Purse command and after the modifications done on it by various transaction related commands during the session. Update the Purse Balance by performing signed addition between the existing Purse Balance and the Amount field.
- Increment the Counter Values. Base values are the ones received in Verify Read Purse command and after the modifications done in those during the current session.

  - Increment PTC by 1.
  - If Auto-Topup happened (P1==0x01) then increment the Add Value Counter by 1.
- Decrypt the Debit Receipt Cryptogram by using the Debit Session Key derived in earlier Compute Debit Crypt Command. Use the updated Counter data as the initial vector for this decryption process.
- Verify the latest Purse Balance with the first 3-bytes of the decrypted data. If doesn't match then return error 0x6982.
- Verify the last 8-bytes of the decrypted data against the latest counter data. If doesn't match then return error 0x6982.
- Derive the Signing session key by encrypting the Debit Options, Purse Balance, TRP, Counter data and CSN in that order using the diversified Signature Key.
- Decrypt the received Signed Certificate from the above decrypted data using the Signing session key.
- Compare the decrypted data against the last saved Txn Header. If doesn't match then throw error 0x6982.
- Return 0x9000.

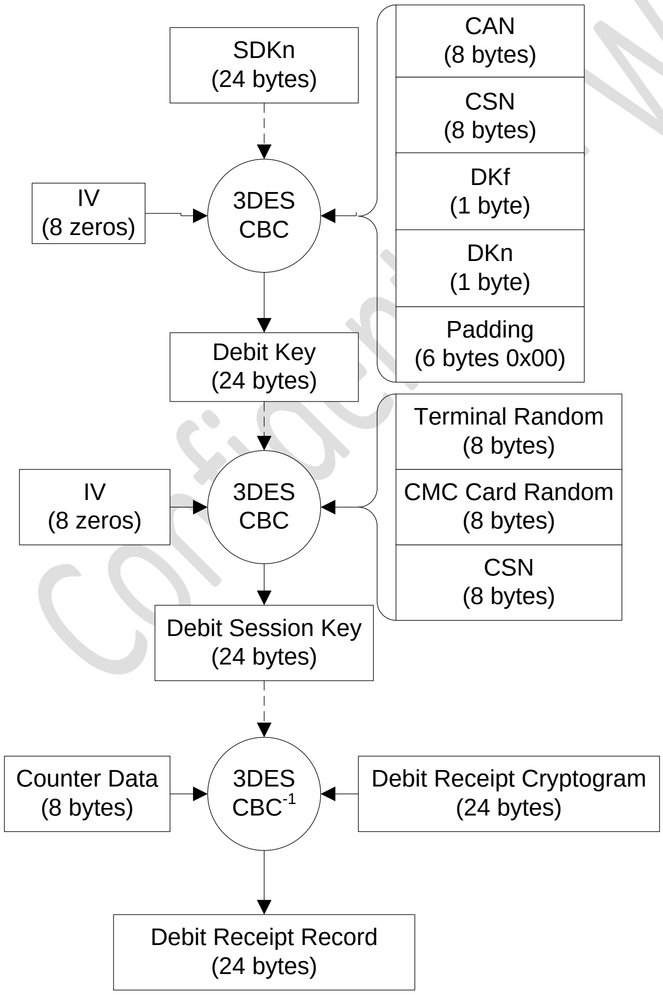


Figure: Debit Receipt Record Verification

Response Data

There will be no response data for this command.

Response SW

Status Words	Meaning
9000	Successful execution
6A80	Master Key not found
6985	Master Key not valid Key not valid for this operation SAM Challenge not available No Preceding Read Purse
6983	Key retry exceeded
6A86	Invalid P1 P2
6700	Wrong Lc or Le
6A82	Master Key EF not found
6982	Security conditions not satisfied

Compute Atomic Update MAC

This command shall be used to compute the MAC to be sent in CMC Atomic update command.

Command Format

Parameter	Value
CLA	0x80
INS	0x8C
P1	SFI of the EF
P2	Offset
Lc	Len
Data	Update Data
Le	0x00

Command Processing

SAM shall perform the following once it receives this command.

- Check the validity of the SFI. If not within the ISO range throw error 0x6A86.
- Check whether Verify Read Purse has been executed during this session or not. If not then return error 0x6985.
- Using the Read Purse Session key calculate MAC over the data received. The MAC should have to be calculated as shown in figure MAC Computation for Atomic Update.
- Format the data as required by the CMC Atomic Update command and append the 4 most significant bytes of the calculated MAC at the end.
- Temporarily save the 4 least significant bytes of the calculated MAC for later usage.
- Return the complete CMC Atomic Update apdu and 0x9000.

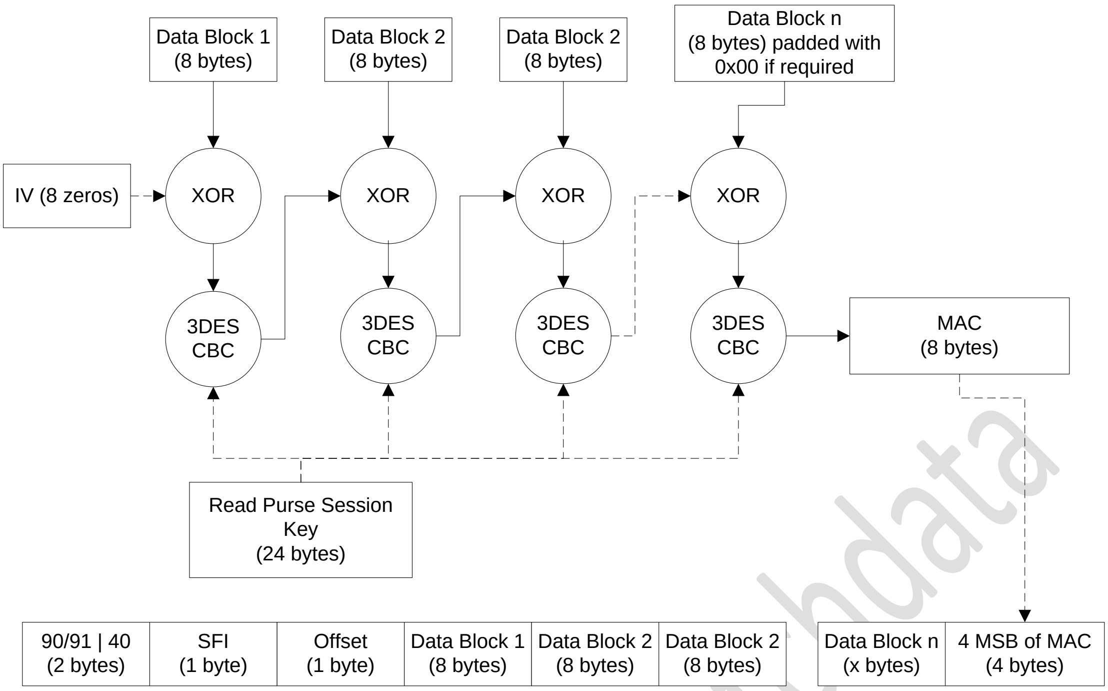


Figure: MAC Computation for Atomic Update

Response Data

The response data of this command shall be the complete CMC Atomic Update command apdu.

Response SW

Status Words	Meaning
9000	Successful execution
6985	No Preceding Read Purse
6A86	Invalid P1 P2
6700	Wrong Lc or Le

Verify Atomic Update MAC

This command shall be used to verify the MAC returned from the CMC as a response data to CMC Atomic Update command.

Command Format

Parameter	Value
CLA	0x80
INS	0x8E
P1	0x00
P2	0x00
Lc	0x04
Data	CMC Atomic Update MAC response
Le	-

Command Processing

After receiving this command the card shall perform the following steps:

- Check P1P2. If anything other than 0x0000 return error 0x6A86.
- Check Lc. If not 0x04 return 0x6700.
- Check the previous executed command. If not Compute Atomic Update MAC then returns error 0x6985.
- Compare the data sent in command data field with the 4 least significant bytes of the previous computed MAC in Compute Atomic Update MAC command. If doesn't match then return error 0x6982.
- Return 0x9000.

Response Data

There shall be no response data for this command.

Response SW

Status Words	Meaning
9000	Successful execution
6985	Previous command is not Compute Atomic Update MAC
6A86	Invalid P1 P2
6700	Wrong Lc
6982	MAC doesn't match

Diversify Key

This command shall be used to get the diversified key. These diversified keys shall be personalized in to the CMC card.

Command Format

Parameter	Value
CLA	0x80
INS	0x54
P1	SAM Master Key EF SFI
P2	SAM Master Key Number
Lc	0x12
Data	See below
Le	-

Data Sent

CMC CAN	CMC CSN	CMC Key EF SFI	CMC Key Number
8-bytes	8-bytes	1-byte	1-byte

Command Processing

After receiving this command the card shall perform the following steps:

- Check Lc. If not 0x12 return 0x6700.
- Find the SAM Master Key EF based on the SFI provided in P1. If not found return 0x6A82.

- Find the Key specified in the P2 parameter. If not found return 0x6A80.
- Check the Master key validity. If not valid return 0x6985.
- Check the retry counter status. If exceeded then return 0x6983.
- If the diversification data as indicated in the Master Key header can't be found then return error status 0x6985.
- Pad the data with diversification data which is associated with the referred Master key.
- Encrypt the above data with the master key.
- Return the encrypted result and SW9000.

Response Data

There shall be no response data for this command.

Response SW

Status Words	Meaning
9000	Successful execution
6985	Key not valid
6A86	Invalid P1 P2
6700	Wrong Lc
6983	Retry counter exceeded