AN12705

MIFARE SAM AV3 - X interface

Rev. 1.1 — 10 January 2020 Application note

521911 COMPANY PUBLIC

Document information

Information	Content
Keywords	MIFARE SAM AV3, TDEA, AES, RSA, MIFARE Plus, MIFARE DESFire EV1, X interface.
Abstract	This application note describes usages of MIFARE SAM AV3 in X interface.

Revision history

Rev	Date	Description
1.1	20200110	AN number changed, security status changed into “Company Public”.
1.0	20190115	Initial version

All information provided in this document is subject to legal disclaimers.

1 Introduction

MIFARE SAMs (**S**ecure **A**pplication **M**odule) have been designed to provide the secure storage of cryptographic keys and cryptographic functions for the terminals to access the MIFARE products securely and to enable secure communication between terminals and host (backend).

1.1 Scope

This application note presents examples of using MIFARE SAM AV3 (referred to SAM in this document, if not otherwise mentioned) in X-interface[1]. In this document, the SAM is in AV3 mode. There is a set of application note for MIFARE SAM AV3; each of them is addressing specific features. The list of application note is given in [4].

This application note is a supplement document for application development using MIFARE SAM AV3. Should there be any confusion please check MIFARE SAM AV3 datasheet [1]. Best use of this application note will be achieved by reading this specification [1] in advance.

Note: This application note does not replace any of the relevant data sheets, datasheets, application notes or design guides.

1.2 Abbreviation

Refer to Application note “MIFARE SAM AV3 – Quick Start up Guide” [4].

1.3 Examples presented in this document

The following symbols have been used to mention the operations in the examples:

= Preparation of data by SAM, PICC or host.

> Data sent by the host to SAM or PICC (if not mentioned, SAM).

< Data Response from SAM or PICC (if not mentioned, SAM).

Table 1. C-APDU:

CLA	INS	P1	P2	Lc	Data (nc)	Le

Table 2. R-APDU:

Response data	SW1	SW2

Please note, that the numerical data are used solely as examples. They appear in the text in order to clarify the commands and command data.

Any data, values, cryptograms are expressed as hex string format if not otherwise mentioned e.g. 0x563412 in hex string format represented as “123456”. Byte [0] = 0x12, Byte [1] = 0x34, Byte [2] = 0x56.

[1] MIFARE SAM AV3 is directly connected to reader IC [4].

1.4 X interface

The host is managing the communication to SAM only, and SAM is managing all the required communication to PICCs.

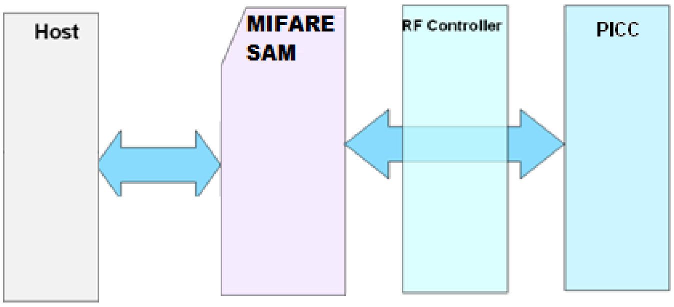


Figure 1. Architecture in X interface

RF controller can be RC52x, PN51x or RC66x. The X interface is explained in the following chapter.

2 X interface

MIFARE SAM AV3 has the FW for ISO/IEC 14443, MIFARE Classic, MIFARE DESFire (EV1, EV2 and light) and MIFARE Plus X, S, SE and EV1. The µC sends the command to SAM for specific task related to RF (PICC) and SAM performs that task fully independent of µC.

2.1 MIFARE SAM AV3, X interface

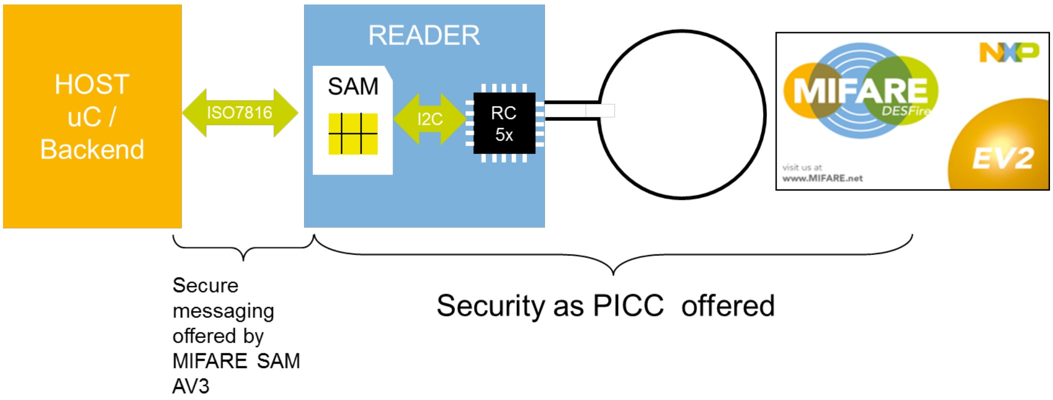


Figure 2. Reader with MIFARE SAM AV3

The I²C interface has to be implemented as described in [9]. The slave address of the MFRC52x/PN51x/RC66x is fixed in the SAM AV3.

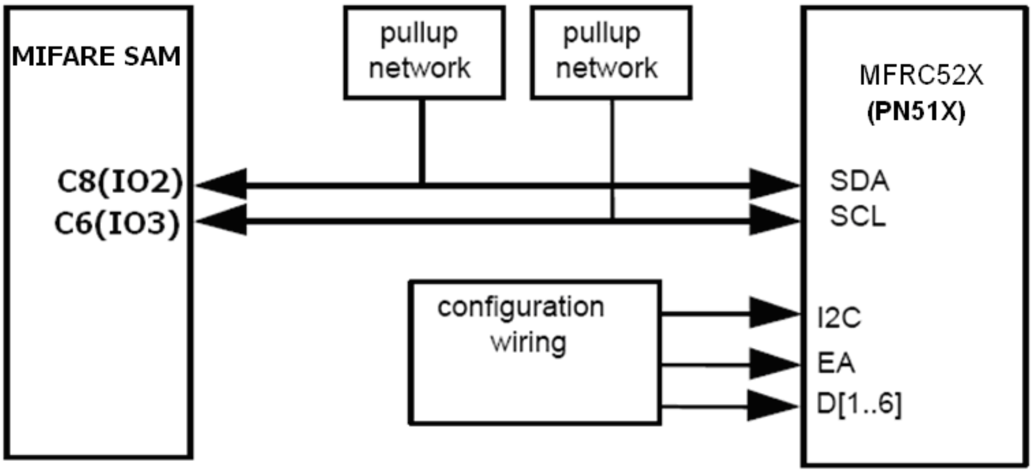


Figure 3. Detail I²C interface

2.2 Initializing the X interface

The chip must be initialized before using the X interface by executing the “RC_Init” command. The RC_Init establishes the I^{2}C communication between SAM and MFRC52X. The RF field must be turned on (if not done using the saved register setting) before any RF communication. One example flow diagram is shown in the following figure.

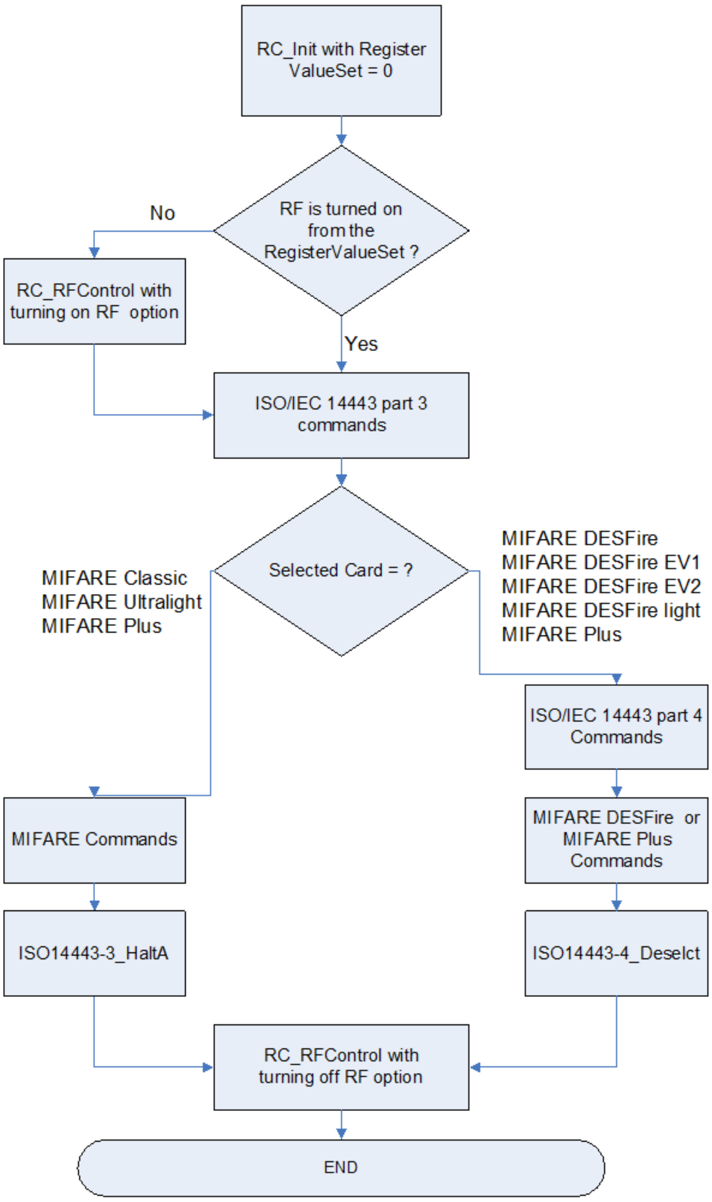

RC_Init with Register ValueSet = 0
RF is turned on from the RegisterValueSet ?
No
RC_RFControl with turning on RF option
Yes
ISO/IEC 14443 part 3 commands
Selected Card = ?
MIFARE Classic
MIFARE Ultralight
MIFARE Plus
MIFARE DESFire
MIFARE DESFire EV1
MIFARE DESFire EV2
MIFARE DESFire light
MIFARE Plus
ISO/IEC 14443 part 4 Commands
MIFARE Commands
MIFARE DESFire or MIFARE Plus Commands
ISO14443-3_HaltA
ISO14443-4_Deselct
RC_RFControl with turning off RF option
END

Figure 4. X interface command-sequence for MIFARE product family

3 X interface functions

The functions supported in X interface are also known as X functionalities. All the X-functionalities commands are listed in the following table. **Some of them are shown with examples in this application note.** For detail descriptions, refer to [2].

Table 3. All X functionalities commands

Command	CLA	INS	P1	P2	Lc	Data	Le	Purpose
MFRC52X Control commands								
RC_ReadRegister	8X	EE	00	00	xx	xx..xx	00	Reads the RC52X register.
RC_WriteRegister	8X	1E	00	00	xx	xx..xx	-	Writes to the RC52X register.
RC_RFControl	8X	CF	00	00	02	mSec	-	Turns on or off the RF field.
RC_Init	8X	E5	xx	00	-	-	-	Initializes the Interface between SAM and RC52X.
RC_LoadRegisterValueSets	8X	2E	xx	xx	xx	xx..xx	-	Loads the register values for initializing the RC52X.
ISO/IEC 14443, type A card activation command								
ISO14443-3_Request_WakeUp	8X	25	00	00	01	26 or 52	00	Sends the REQA or WUPA command to the RF.
ISO14443-3_Anticollision_Select	8X	93	00	00	xx	xx..xx	00	Sends anticollision and select commands for all cascade level.
ISO14443-3_ActivateIdle	8X	26	xx	xx	xx	xx..xx	00	Activates card(s) from Idle state.
ISO14443-3_ActivateWakeUp	8X	52	00	00	xx	xx..xx	-	Activates card from Halt state.
ISO14443-3_HaltA	8X	50	00	00	-	-	-	Halts the activated card.
ISO14443-3_TransparentExchange	8X	7E	xx	00	xx	xx..xx	00	Transceives any byte and bit to and from the PICC
MIFARE commands								
MF_Authenticate	8X	0C	00	00	xx	xx..xx	-	Authenticates MIFARE.
MF_Read	8X	30	00	00	xx	xx..xx	00	Reads MIFARE block(s).
MF_Write	8X	A0	xx	00	xx	xx..xx	-	Writes to MIFARE block(s).
MF_ValueWrite	8X	A2	00	00	xx	xx..xx	-	Prepares block(s) to value block(s).
MF_Increment	8X	C3	00	00	xx	xx..xx	-	Increments the value block(s).
MF_Decrement	8X	C0	00	00	xx	xx..xx	-	Decrements the value block(s).

Command	CLA	INS	P1	P2	Lc	Data	Le	Purpose
MF_Restore	8X	C2	00	00	xx	xx. .xx	-	Copies value block(s) to other value block(s)
MF_AuthenticateRead	8X	3A	00	00	xx	xx. .xx	00	Authenticates and reads MIFARE block(s).
MF_AuthenticateWrite	8X	AA	00	00	xx	xx. .xx	-	Authenticates and writes to MIFARE block(s).
MF_ChangeKey	8X	A1	xx	00	xx	xx. .xx	-	Changes (updates) MIFARE keys in the sector trailer.
MIFARE Ultralight commands								
UL_PwdAuthPICC	8X	2D	00	00	xx	xx xx	-	Performs the Password Authentication on the MIFARE Ultralight EV1 PICC
ISO14443-4 Type commands								
ISO14443-4_RATS_PPS	8X	E0	00	00	03	xx. .xx	00	Performs the RATS and PPS command
ISO14443-4_Init	8X	11	00	00	05	xx. .xx	-	Initializes PICC and reader for protocol data exchange, alternative command of ISO14443-4_RATS_PPS.
ISO14443-4_Exchange	8X	EC	xx	00	xx	xx. .xx	-/ 00	Transceives APDU to and from the PICC.
ISO14443-4_PresenceCheck	8X	4C	00	00	-	-	-	Tracks the PICC.
ISO14443-4_Deselcect	8X	D4	xx	00	-	-	-	Deselects the PICC and PICC goes to halt state.
ISO14443-4_FreeCID	8X	FC	00	00	xx	xx. .xx	-	Frees the CID used by the PCD.
MIFARE DESFire related commands								
DESFire_AuthenticatePICC	8X	DA	xx	xx	xx	xx. .xx	00	Performs complete 3-pass mutual authentication for DESFire.
DESFire_ChangeKeyPICC	8X	DE	xx	xx	xx	xx. .xx	00	Changes the keys in DESFire
DESFire_WriteX	8X	D3	xx	xx	xx	xx. .xx	00	Can be used for DESFire memory updated commands.
DESFire_ReadX	8X	D2	00	xx	xx	xx. .xx	00	Can be used for DESFire memory reading commands.
DESFire_CreateTMFilePICC	8X	D1	xx	xx	xx	xx xx	00	Creates a Transaction MAC File in the PICC
MIFARE Plus related command								
MFP_WritePerso	8X	A8	00	00	xx	xx. .xx	00	The data is transferred in plain, so perform the write_perso command in a secure site.

Command	CLA	INS	P1	P2	Lc	Data	Le	Purpose
MFP_Authenticate	8X	70	0x	00	xx	xx..xx	00	The same command is used in all security level (SL) of MIFARE Plus, P1 is used to distinguish the SL.
PCD_Authenticate	8X	73	0x	00	xx	xx xx	00	Performs the Post-Delivery configuration on the MIFARE Plus
MFP_CombinedRead	8X	31	00	00	04	xx..xx	00	The data field contains MIFARE Plus cmd+2-byte block nr + nr. of blocs to read
MFP_CombinedWrite	8X	32	00	00	xx	xx..xx	00	The data filed contains the plain command.
MFP_ChangeKey	8X	A5	0x	00	xx	xx..xx	00	Only one key can be changed at a time.
MFP_AuthSectorSwitch	8X	72	xx	00	xx	xx xx	00	Switches the security level of MIFARE Plus sectors
MIFARE Ultralight C Authentication command								
ULC_AuthenticatePICC	8X	2C	0x	00	xx	xx..xx	00	Only CMAC based key diversification is allowed.
MIFARE common								
TMRI_CommitReaderID	8X	37	00	00	xx	xx xx	00	Commits the ReaderID to the PICC
Programmable Logic								
SAM_PLExec	8X	BE	xx	00	xx	xx xx	00	Triggers the execution of the programmable logic
SAM_PLUpload	8X	BF	xx	xx	xx	xx xx	00	Updates the code in the programmable logic
Virtual Card Architecture								
VCA_ProximityCheck	8X	FB	0x	00	xx	xx..xx	00	Performs the proximity check.
VCA_Select	8X	45	0x	00	xx	xx..xx	00	Used for VC selection

X = 0, 1, 2, 3; the logical channel.

3.1 RF Controller IC Control commands

These commands are controlling, preparing and enabling the RC52x/PN51x/RC663 for further communication with PICC. As the reader IC can be always in one state, so the logical channel has no role in these commands.

3.1.1 RC_LoadRegisterValueSet

RC_LoadRegisterValueSet loads one full set of values (deleting complete set and loading the new value set) in a single command. In the SAM, 8 sets of register values can be stored. The default register values stored at register set 0 is given in the Table 4.

It is required to modify some of the register values to initialize the reader IC (RC52x) for ISO/IEC 14443 type A. The modified values are given also in Table 4.

Table 4. Default "Register Set 0" storage

RC52X register name	RC52X/PN51X register address	Default set Value	Modified Value to be reloaded
TModReg	2A	82	82
TPrescalerReg	2B	AA	AA
TxASKReg	15	40	40
RxThresholdReg	18	75	75
DemodReg	19	4D	4D
RFCfgReg	26	59	59
GsNReg	27	F4	F4
CWGsPReg	28	3F	3F
ModGsPReg	29	11	11
ControlReg	0C	10	10
CommandReg	01	-	00

As RC_LoadRegisterValueSet delete and store the complete set, it is required to load the full set (not only the modified one). Single register can be loaded using "RC_WriteRegister" command. This "RC_LoadRegisterValueSet" command can be executed (see table 4) once at SAM personalization and can be used through the SAM life as long the register set is not required to use for other purposes.

For other type of ISO/IEC 14443 standard, register setting can be defined with the help of register description given in 9; starting register values can be requested from local, ID FAEs as well. In the following example the register set 0x01 is loaded with the following values.

Table 5. Register Set for ISO/IEC 14443 Type A

RC52X register name	RC52X/PN51X register address	Value will be set to
TModReg	2A	82
TPrescalerReg	2B	AA
TxASKReg	15	40
RxThresholdReg	18	75
DemodReg	19	4D
RFCfgReg	26	59
GsNReg	27	F4
CWGsPReg	28	3F
ModGsPReg	29	11
ControlReg	0C	10
CommandReg	01	00

The above register setting is stored in the register set 0x00 in the following example.

Table 6. Example of RC_LoadRegisterValueSet

Step	Indication		Data/Message	Comments
1	C-APDU	>	802E0000162A822BAA154 01875194D265927F4283F2 9110C100100	Data field contains in pair [addr, value]
2	R-APDU	<	9000	Loading of register is successful.

The RC_Init command with the value P1 = 0x00 will initialize the RC52X/PN51X with the register settings stored in register set 0x00 in this example.

3.1.2 RC_Init

The RF controller IC (RC52X/PN51X) is initialized with the addressed set of values stored in the SAM memory. By default, the register value sets 0 contains ISO/IEC 14443 A type register settings of the RC52X and PN51X (RF is turned off).

Table 7. Example of RC_Init

Step	Indication		Data/Message	Comment
1	C-APDU	>	80E58000	Register set = 0, and higher speed in I2C.
2	R-APDU	<	9000	Status

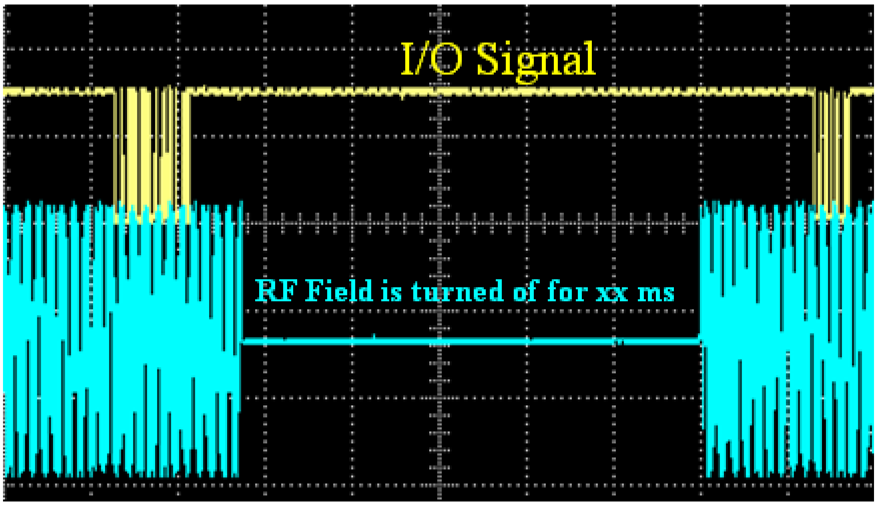
3.1.3 RC_RFControl

This command can be interpreted as the resetting of RF. The time (in ms) given in the data field is the time the RF remains turned off before turning on again. The time “0000” given in the data field turned off the RF.

Table 8. Example of RC_RFControl

Step	Indication		Data/Message	
1	C-APDU	>	80CF0000020500	(5ms is the RF turned off time)
2	R-APDU	<	9000	

In the following figure the RF field is shown while executing the above command.



The time provided in the data field and the real RF turned off time can be in tolerance up to 10%

Figure 5. RF behavior with RC_RFControl command

3.2 ISO14443-3 type A card activation commands

All the ISO/IEC 14443 part 3 type A commands are mapped in these APDU commands. Moreover, there are some compound commands which can activate A type card with minimum user interaction. It is also possible to activate the ISO/IEC 14443 B type card using the commands stated here.

3.2.1 ISO14443-3_ActivateIdle

This is a compound command, performs all ISO/IEC 14443 type A card activation sequences (ReqA – Anticollision - select). In the following example a DESFire card is activated.

Table 9. Example of ISO14443-3_ActivateIdle

Step	Indication		Data/Message	Comment
1	P1	=	05*	The application will activate up to 5 cards.
2	P2	=	03	The ATQA and SAK filter is applied
3	ATQA filter	=	FF44FF03	all bits of ATQA (4403 ATQA of DESFire) are considered
4	SAK filter	=	FF20	All bits of SAK is considered. For CL-2 and CL3 only the final SAK is considered.
5	ISO14443-3_ ActivateIdle C-APDU	>	8026050306FF44FF03FF 2000	
6	ISO14443-3_ ActivateIdle R-APDU	<	014403200704261419701 C809000;	One DESFire card has been found.

* All the activated card will go to halt state. To continue with a card, those cards need to wake up using ActivateWakeUp command. If P1= 01, then the card is in activated state.

3.2.2 ISO14443-3_TransparentExchange

Using this command every bits and bytes can be sent to the card. One example of using this command is to activate ISO/IEC 14443 B type card. In the following example the REQB command is shown.

Table 10. Example of ISO14443-3_TransparentExchange

Step	Indication		Data/Message	Comments
1	C-APDU	>	807E00000305000000	REQB command
2	R-APDU	<	50xxxxxxxxxxxxxxxxxxxxxx9000	ATQB response

Of course, before executing this command the RC523 registers have to be set to the correct values using RC_Init command. The register setting can be requested from Customer Application Support.

3.3 MIFARE Commands

These are the commands can be used to communicate with the MIFARE Classic (MIFARE Plus SL1) PICCs.

3.3.1 MF_Authenticate

Table 11. MF_Authenticate Example

Step	Indication		Data/Message	Comments
1	MIFARE UID	=	443898DE	In case of 7-byte UID, take last four byte.
2	SAM Key Entry No	=	02	The MIFARE Key entry is personalized in advance
3	Key version of the SAM Key Entry	=	01	
4	MIFARE Key Type A	=	0A	
5	MIFARE Block Nr	=	28	
6	Div constant	=	0A	Here the sector number.
7	C-APDU	>	800C000009443898DE02010A280A	
8	R-APDU	<	9000	

3.3.2 MF_Read

MF_Read command can read multiple numbers of blocks. In RF level the SAM is performing the read command for every block and providing the total data to the user in one step.

Table 12. MF_Read Example

Step	Indication		Data/Message	Comments
1	C-APDU	>	803000000304050600	Data field is the block numbers to be read.
2	R_APDU	<	0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000009000	Content of block 4, 5, 6.

In the above example, block number 04, 05 and 06 (sector 1) have been read. If any block has different access condition, the SAM will not return data from the read block(s) but only the NACK (90FX).

3.3.3 MF_Write

MF_Write command can read multiple numbers of blocks. In RF level the SAM is performing the read command for every block and providing the total data to the user in one step.

Table 13. MF_Write Example

Step	Indication		Data/Message	Comments
1	P1	=	00	16-byte data for writing each block
2	C-APDU	>	80A0000033**04**01020304050607080910111213141516**05**01020304050607080910111213141516**06**01020304050607080910111213141516	Data field contains [block nr,16-byte data; block nr, 16-byte data; …]
3	R_APDU	<	9000	Successful

In the above example, block number 04, 05 and 06 (sector 1) have been written. If the blocks access condition is different, the SAM will return NACK (90FX) but may be some blocks already updated. As example, in this example if block 6 has different write access condition than the current authentication state, SAM will return 90FX but already block number 4 and 5 are updated.

3.3.4 MF_ValueWrite

MF_ValueWrite can personalize one or several blocks to value block. In the following example block number 5 and block number 6 are personalized for 100 units.

Table 14. MF_ValueWrite Example

Step	Indication		Data/Message	Comment
1	Block Address of MIFARE	=	05	
2	Value	=	64000000	Value = 100
3	Address	=	FF00FF00	
4	Block Address of MIFARE	=	06	

Step	Indication		Data/Message	Comment
5	Value	=	64000000 (100 unit)	
6	Address	=	FF00FF00	
7	C-APDU	>	80A20000120564000000FF00FF000664000000FF00FF00	
8	R-APDU	<	9000	Successful

Please note, the address provided here is fully written in the value block (last 4 bytes of the 16-byte value block). If the blocks access condition is different, the SAM will return NACK (90FX) but some blocks may have already been updated.

3.3.5 MF_Increment

MF_Increment can increment the value block(s). In the following example the value of block 5 is incremented by 10 units and transferred to block number 6.

Table 15. MF_Increment Example

Step	Indication		Data/Message	Comments
1	Source Address	=	05	
2	Destination Address	=	06	
3	Value to be incremented by	=	0A000000	Value = 10
4	C-APDU	>	80C300000605060A000000	
5	R-APDU	<	9000	Successful

3.3.6 MF_Decrement

MF_Decrement can decrement the value block(s). In the following example the value of block 5 is decremented by 10 units and transferred to block number 6.

Table 16. MF_Decrement Example

Step	Indication		Data/Message	Comments
1	Source Address	=	05	
2	Destination Address	=	06	
3	Value to be incremented by	=	0A000000	Value = 10
4	C-APDU	>	80C000000605060A000000	
5	R-APDU	<	9000	Successful

3.3.7 MF_AuthenticateRead

This is a compound command consolidating Authentication and read, which can be very useful for optimizing performance transaction time of MIFARE Classic applications. In the following example, the sector number 10 is authenticated and blocks 40, 41 and 43 (3 user blocks of sector 10) will be read.

Table 17. MF_AuthenticateRead Example

Step	Indication		Data/Message	Comments
1	MIFARE UID	=	443898DE	Last 4-byte in case of 7-byte UID.
2	CmdSettings	=	02	key information is provided and diversifying key.
3	SAM Key Entry No	=	02	SAM key entry number.
4	Key version of the SAM Key Entry	=	01	
5	MIFARE Key Type A	=	0A	
6	MIFARE Block Nr to authenticate	=	28	
7	Div Constant	=	0A	Here the sector number
8	Number of blocks to be read	=	03	
9	MIFARE block numbers to read	=	28292A	3 blocks 40,41, 42
10	C-APDU	>	803A00000E443898DE0202010A280A0328292A00	
11	R-APDU	<	41627549736D61696C204341534E5850640000009BFFFFFF6400000000FF00FF640000009BFFFFFF6400000000FF00FF9000	3x16= 48 bytes data and SW1SW2.

Please note, if the block read accesses are different or required keys are different, then the information has to be provided in the data field. Please refer to [2]. If any block has different access condition, the SAM will not return data from the read block(s) but only the NACK (90FX).

3.3.8 MF_AuthenticateWrite

This is a compound command consolidating Authentication and write, which can be very useful for optimizing performance transaction time of MIFARE Classic applications. In the following example, the sector number 1 is authenticated and blocks 4, 5 and 6 (3 user blocks of sector 1) will be written.

Table 18. MF_AuthenticateWrite Example

Step	Indication		Data/Message	Comment
1	MIFARE UID	=	540B9ADE	Last 4-byte in case of 7-byte UID.
2	CmdSettings	=	02	key information is provided and diversifying key
3	SAM Key Entry No	=	01	
4	Key version of the SAM Key Entry	=	02	
5	MIFARE Key Type	=	0B	Key type B

Step	Indication		Data/Message	Comment
6	MIFARE Block Nr to authenticate	=	04	
7	Div Constant	=	01	
8	Number of blocks to be written	=	03	
9	MIFARE block numbers and data	=	**04**01020304050607080910111213141516**05**0102030405060708091011121314151 6**06**01020304050607080910111213141516	Block nr, data; block nr, data ….
10	C-APDU	>	80AA00003E540B9ADE0201010B040103040102030405060708091011121314151605010203040506070809101112131415160601020304050607080910111213141516	
11	R-APDU	<	9000	Successful

Please note, if the block write accesses are different or required keys are different, then the information has to be provided in the data field. Please refer to [2]. If the blocks access condition is different, the SAM will return NACK (90FX) but may be some blocks already updated.

3.3.9 MF_ChangeKey

This command can be used to personalize or roll the MIFARE keys in MIFARE Classic cards. MF_ChangeKey command at first generates the MIFARE diversified key and then writes it to the corresponding sector trailer.

Table 19. MF_ChangeKey Example

Step	Indication		Data/Message	Comments
1	KeyCompMeth (P1)	=	06	Both key A and key B have to be diversified), Please note bit 0 and other bits are RFU and has to be set 0.
2	SAM Key Entry No	=	02	Which is a MIFARE Key entry, personalized in advance.
3	Key version of the SAM Key Entry for MIFARE key A	=	01	
4	Key version of the SAM Key Entry for MIFARE key B	=	01	The version for Key A and Key B can be different. If different, the Key A is taken from one position (version) and Key B is taken from another position (version).

Step	Indication		Data/Message	Comments
5	MIFARE Block number where to store the key	=	2B	Sector trailer block number, here we are taking sector number 0A.
6	Access conditions	=	08778F69	3 bytes AC and GPB
7	MIFARE UID	=	443898DE	Last 4-byte in case of 7-byte UID.
8	Div Constant	=	0A	Here is the sector number.
9	C-APDU	>	80A106000D0201012B0877 8F69443898DE0A	
10	R-APDU	<	9000	Successful

3.4 Preparing the proximity chips for T=CL half duplex transmission

MIFARE SAM AV3 supports the “Exchange Transparent Data” state with up to 4 cards (according to ISO/IEC 14443-4, the number of cards in this state can be up to 15, CID 0 to CID 14). One logical channel can be assigned to one specific CID. In the following a flow diagram is shown:

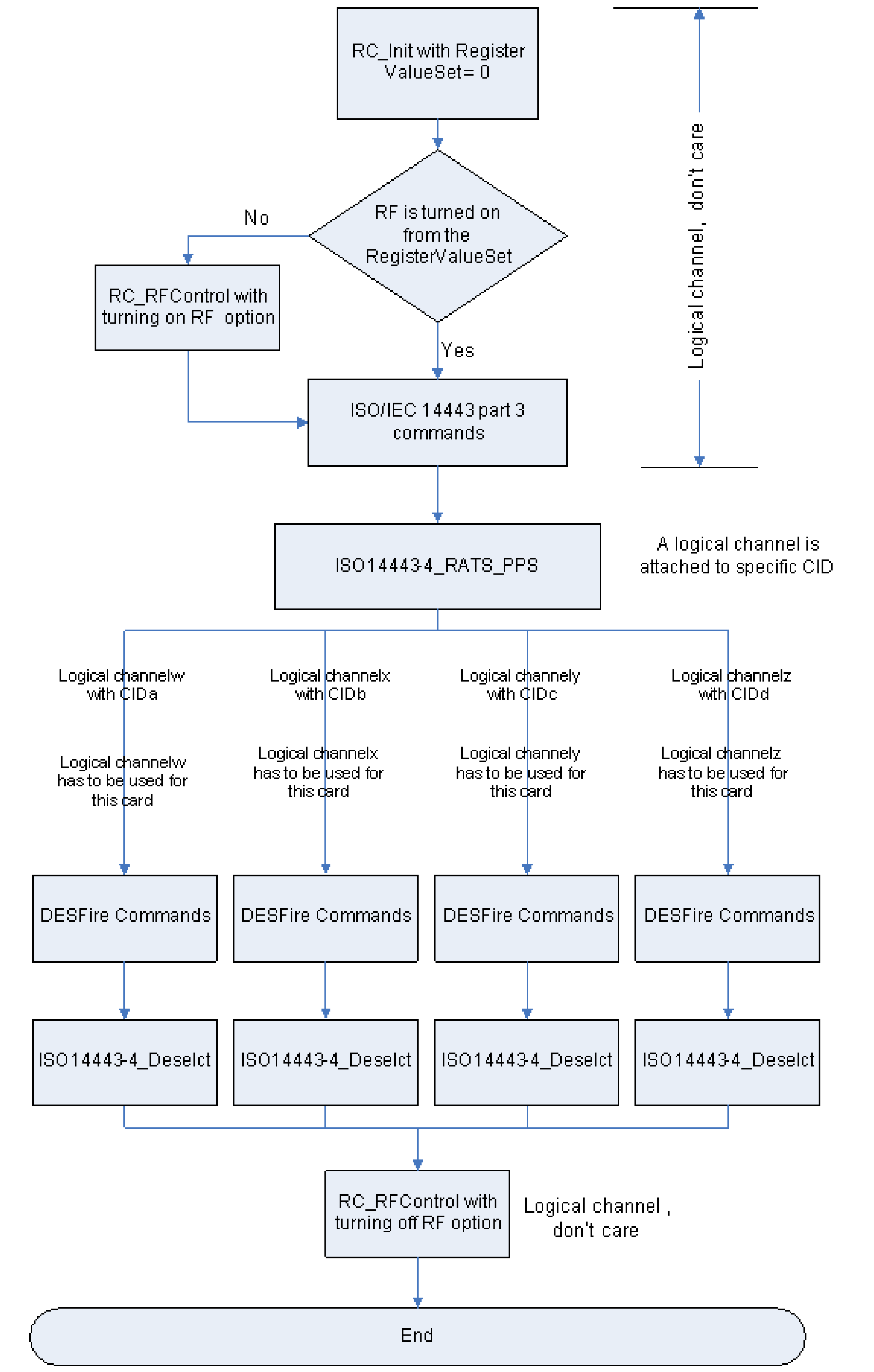

RC_Init with Register ValueSet= 0
RF is turned on from the RegisterValueSet
No
Yes
RC_RFControl with turning on RF option
ISO/IEC 14443 part 3 commands
Logical channel, don't care
ISO14443-4_RATS_PPS
A logical channel is attached to specific CID
Logical channelw with CIDa
Logical channelx with CIDb
Logical channely with CIDc
Logical channelz with CIDd
Logical channelw has to be used for this card
Logical channelx has to be used for this card
Logical channely has to be used for this card
Logical channelz has to be used for this card
DESFire Commands
DESFire Commands
DESFire Commands
DESFire Commands
ISO14443-4_Deselct
ISO14443-4_Deselct
ISO14443-4_Deselct
ISO14443-4_Deselct
RC_RFControl with turning off RF option
Logical channel , don't care
End

Figure 6. Specific logical channel is assigned in ISO/IEC14443-4

3.4.1 ISO14443-4_RATS_PPS

Table 20. RATS_PPS Example

Step	Indication		Data/Message	Comments
1	CID	=	01	
2	DRI	=	02	424 kbps (PCD to PICC)
3	DSI	=	02	424 kbps (PICC to PCD)
4	C-APDU	>	80E000000301020200	
5	R-APDU	<	0102020675778102809000	
Activating another card				
6	CID	=	02	
7	DRI	=	01	212 kbps (PCD to PICC)
8	DSI	=	01	212 kbps (PICC to PCD)
9	C-APDU	>	81E000000302010100	
10	R-APDU	<	0202020675778102809000	
Accessing the card with CID 01, ‘GetApplicationID’ command				
11	C-APDU	>	80EC0000016A00	Logical channel 0 is communicating with card with CID = 0.
12	R-APDU	<	004444449000	
Accessing the card with CID 02, ‘GetApplicationID’ command				
13	C-APDU	>	81EC0000016A00	Logical channel 1 is communicating with card with CID = 1.
14	R-APDU	<	002F8CF11111119000	

MIFARE SAM AV3 supports using different RF communication speeds with different cards at the same time.

3.4.2 ISO14443-4_PresenceCheck

For tracking a card, (if still the activated card is present) this command can be issued, facilitates the windows resource manager according to PC/SC. This command will not change any state of the card.

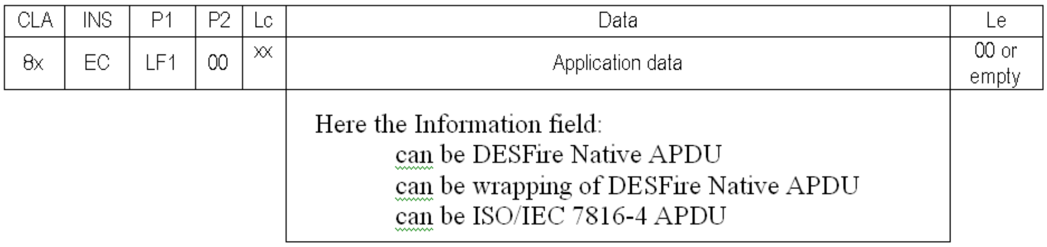
Table 21. ISO14443-4_PresenceCheck Example

Step	Indication		Data/Message	Comments
1	C-APDU	>	804C0000	
2	R-APDU	<	9000	Card is present.

In this example the presence of the card attached to logical channel 0 is checked.

3.5 Accessing MIFARE DESFire

The “ISO14443-4_Exchange” command can be used to access a MIFARE DESFire (EV1) or any ISO/IEC 14443 part 4 compliant PICCs. In this case, the data field contains the application data.



CLA	INS	P1	P2	Lc	Data	Le
8x	EC	LF1	00	xx	Application data	00 or empty

Here the Information field:
can be DESFire Native APDU
can be wrapping of DESFire Native APDU
can be ISO/IEC 7816-4 APDU

Figure 7. ISO14443-4_Exchange Command APDU for DESFire

3.5.1 Selecting MIFARE DESFire Application

MIFARE DESFire “Select Application” command in native mode is shown in the following table.

Table 22. Example of Select Application command

Step	Indication		Data/Message	Comment
1	Application ID	=	123456	3-byte DESFire application ID
2	DESFire Select application command	=	5A123456	Select application cmd and 3-byte AID
3	ISO14443-4_Exchange C-APDU	>	80EC0000045A12345600	DESFire select application command is packed in the data field of ISO14443-4_ Exchange command APDU.
4	ISO14443-4_Exchange R-APDU	<	009000	DESFire response is in the response data field and SW1SW2. Here ‘00’ is the DESFire status code.

3.5.2 MIFARE DESFire Read command

MIFARE DESFire “Read Data” command in native mode is shown in the following table.

Table 23. Example of MIFARE DESFire Read native APDU

Reading 70 bytes from a standard data file

Step	Indication		Data/Message	Comments
1	Read command	=	BD	
2	File no	=	02	
3	Offset	=	000000	
4	length	=	460000 (70 bytes)	
5	DESFire Native APDU	=	BD02000000460000	

Step	Indication		Data/Message	Comments
6	ISO14443-4_Exchange C-APDU	>	80EC000008BD0200000046000000	MIFARE DESFire native APDU command is packed in the data field of the ISO14443-4_Exchange C-APDU
7	ISO14443-4_Exchange R-APDU	<	AF01020304050607080910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455565758599000 (AF is the DESFire native status code [5] and 59 bytes data)	MIFARE DESFire EV1 response is packed in ISO14443-4_Exchange R-APDU
8	C-APDU to SAM for more data	>	80EC000001AF00	
9	R-APDU from SAM	<	0060616263646566676869709000 (00 is the DESFire native status code [5] and 11 bytes data)	
10	Application data read	=	0102030405060708091011121314151617181920212223242526272829303132333435363738394041424344454647484950515253545556575859606162636465666768696970	

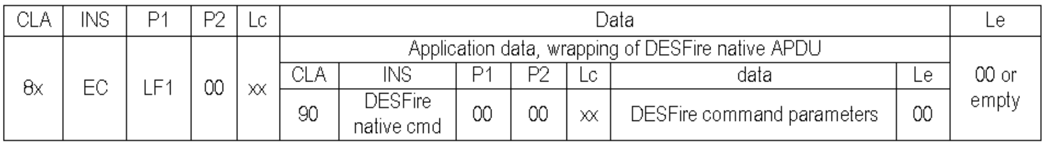
Table 24. Example of Wrapping of DESFire Native APDU

Reading 70 bytes from a standard data file

Step	Command		Data/Message
1	Read command	=	BD
2	File no	=	02
3	Offset	=	000000
4	length	=	460000 (70 bytes)
5	Wrapped APDU[5]	=	90BD0000070200000046000000
6	C-APDU to SAM	>	80EC00000D90BD000007020000004600000000
7	R-APDU from SAM	<	01020304050607080910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455565758599 1AF9000 (91AF is the SW1SW2 from wrapping and 59 bytes data)
8	C-APDU to SAM for more data	>	80EC00000590AF00000000
9	R-APDU from SAM	<	6061626364656667686970 91009000 (9100 is the SW1SW2 and 11 bytes data)

Step	Command		Data/Message
10	Application data read	=	0102030405060708091011121314151617181920212223 2425262728293031323334353637383940414243444546 4748495051525354555657585960616263646566676869 70

Important clarification: The complete APDU is made up of two APDUs. DESFire's APDU is transported/wrapped within the standard ISO14443 part IV APDU, as shown in the following figure.



<table>
<tr><td>CLA</td><td>INS</td><td>P1</td><td>P2</td><td>Lc</td><td colspan="7">Data</td><td>Le</td></tr>
<tr><td rowspan="3">8x</td><td rowspan="3">EC</td><td rowspan="3">LF1</td><td rowspan="3">00</td><td rowspan="3">xx</td><td colspan="7">Application data, wrapping of DESFire native APDU</td><td rowspan="3">00 or empty</td></tr>
<tr><td>CLA</td><td>INS</td><td>P1</td><td>P2</td><td>Lc</td><td>data</td><td>Le</td></tr>
<tr><td>90</td><td>DESFire native cmd</td><td>00</td><td>00</td><td>xx</td><td>DESFire command parameters</td><td>00</td></tr>
</table>

Figure 8. Wrapping of DESFire Native APDU in ISO14443-4_Exchange APDU

Please note, for ISO/IEC 7816-4 INS will have same structure like the above one.

These structures can be used for any DESFire commands. More over, some of the DESFire commands are supported by MIFARE SAM AV3 directly and these commands are named "DESFire related commands" in [1]. In the following some of them are discussed.

3.5.3 DESFire_AuthenticatePICC

This command is very straightforward. The SAM key entry has to be personalized prior to issue DESFire_AuthenticatePICC command. Please make sure, the key entry is in accordance.

Table 25. Example of MIFARE DESFire EV1 Authentication

Reading 70 bytes from a standard data file

Step	Indication		Data/Message	Comments
1	DESFire_ AuthenticatePICC C-APDU	>	80EC0000045A12345600	No key diversification is used.
2	DESFire_ AuthenticatePICC R-APDU	<	9000	Authentication is successful

3.5.4 DESFire_ChangeKeyPICC

This command changes the keys of the MIFARE DESFire EV1 and can be used in personalization or rolling of the keys. It supports the diversification mechanism as described in [1]. Please note the same diversification inputs have to be used for both new and current key, if they both are diversified.

Table 26. Example of DESFire_ChangeKeyPICC

Step	Indication		Data/Message	Comment
1	DESFire key number to be changed (one application key)	=	01	

Step	Indication		Data/Message	Comment
2	Current DESFire key belongs to SAM key entry nr.	=	01	
3	Current DESFire key version (version of the SAM key entry of 1)	=	00	
4	New DESFire key belongs to SAM key entry nr.	=	01	
5	New DESFire key version (version of the SAM key entry of 01)	=	01	
6	P1	=	00100010b (0x22)	b0 is set to 0, DESFire change key nr ≠ currently authenticated key nr. New key will be diversified but not the current one. Key diversification mode is CMAC based.
7	Diversification input	=	049137C9922680	UID of the card, as the CMAC based diversification is used the input length can be any value from 1 to 31.
8	C-APDU	>	80DE22010B010001010491 37C992268000	
9	R-APDU	<	9000	

3.5.5 DESFire_WriteX

“DESFire_WriteX” command is optimized for several memory update-type functions e.g. ChangeKeySettings, WriteData, Credit, Debit, LimitedCredit, WriteRecord for DESFire. Please note, the complete DESFire APDU (DESFire native, ISO 7816 wrapping or ISO7816-4 INS) is provided in the data field. Please check the following example.

Table 27. Example of DESFire_WriteX Command for writing to a data file

Step	Command		Data/Message
1	“Write Data” command for DESFire	=	3D
2	File no, where to write	=	01
3	Offset at which the write starts	=	000000
4	Length of data to be written	=	0A0000 (10 bytes)
5	Data to write	=	01020304050607080910
6	DESFire Native APDU, the application data.	=	3D010000000A000001020304050607080910 (will be the data field of DESFire_WriteX C-APDU)
7	Now mapped to DESFire_WriteX APDU		

Step	Command		Data/Message
8	P1	=	00, last frame
9	P2	=	38, (encrypted communication**, encryption starts from 8th byte as this is the starting of written data bytes)
10	Lc	=	12; (18 bytes from step 6)
11	C-APDU	>	80D30038123D010000000A00000102030405060708091000
12	R-APDU	<	9000

**Please note, “DESFire_WriteX” command cannot be used for plain communication. For plain communication, use the “ISO14443-4_Exchange” command.

DESFire_WriteX command does not support DESFire application chaining. To write longer length of data (does not fit in one write frame, please check [5]), user has to implement the chaining.

3.5.6 DESFire_ReadX

DESFire_ReadX command is optimized for accessing memory (ReadData, GetValue and ReadRecord) in fully encrypted or MACed (CAMCed) communication. The complete DESFire application protocol data unit (Native, ISO7816 wrapping or ISO7816-4 INS) is given in the data field. In the following one example with reading the data file is shown.

Table 28. Example of DESFire_ReadX Command for reading a data file

Step	Command		Data/Message
1	“Read Data” command for DESFire	=	BD
2	File no, to read	=	01
3	Offset at which the read starts	=	000000
4	Length of data to be read	=	0A0000 (10 bytes)
5	DESFire Native APDU, the application data.	=	BD010000000A0000 (will be the data field of DESFire_ReadX C-APDU)
6	Now mapped to DESFire_ReadX APDU		
7	P1	=	00
8	P2	=	30, (encrypted communication)
9	Lc	=	08; (8 bytes from step 6)
10	C-APDU	>	80D200300B0A0000BD010000000A000000 (The length of data “0A0000” to be read has to be added in front of the DESFire APDU as well)
11	R-APDU	<	0102030405060708091090000

**Please note, “DESFire_ReadX” command cannot be used for plain communication. For plain communication, use the “ISO14443-4_Exchange” command.

DESFire_ReadX command does not support DESFire application chaining. To read longer length of data (does not fit in one frame, please check [5]), user has to implement the chaining. Please see the next example.

Table 29. Example of DESFire_ReadX Command for reading a data file with chaining

Step	Command		Data/Message
1	“Read Data” command for DESFire	=	BD
2	File no, to read	=	01
3	Offset at which the read starts	=	000000
4	Length of data to be read	=	960000 (150 bytes)
5	DESFire Native APDU, the application data.	=	BD01000000960000 (will be the data field of DESFire_ ReadX C-APDU)
6	Now mapped to DESFire_ReadX APDU		
7	P1	=	00
8	P2	=	30, (encrypted communication)
9	Lc	=	08; (8 bytes from step 6)
10	C-APDU	>	80D200300B960000BD0100000096000000 (The length of data “960000” to be read has to be added in front of the DESFire APDU as well)
11	R-APDU	<	000102030405060708090A0B0C0D0E0F101112131415 161718191A1B1C1D1E1F202122232425262728292A2B 2C2D2E2F90AF (90AF means more data from the DESFire)
12	C-APDU (for more data, chaining)	>	80D2003001AF00
13	R-APDU	<	303132333435363738393A3B3C3D3E3F4041424344454 64748494A4B4C4D4E4F505152535455565758595A5B5C 5D5E5F606162636465666790AF
14	C-APDU (for more data, chaining)	>	80D2003001AF00
15	R-APDU	<	68696A6B6C6D6E6F707172737475767778797A7B7C7D 7E7F808182838485868788898A8B8C8D8E8F909192939 4959000
16	The complete 150 bytes data	=	000102030405060708090A0B0C0D0E0F1011121314151 61718191A1B1C1D1E1F202122232425262728292A2B2C 2D2E2F303132333435363738393A3B3C3D3E3F4041424 3444546474849 4A4B4C4D4E4F50515253545556575859 5A5B5C5D5E5F606162636465666768696A6B6C6D6E6F 707172737475767778797A7B7C7D7E7F8081828384858 68788898A8B8C8D8E8F909192939495

3.6 Accessing MIFARE Plus

All the MIFARE Plus commands can be executed in X interface of MIFARE SAM AV3.

3.6.1 MFP_WritePerso

MFP_WritePerso command requires the exact data/key to be written to MIFARE Plus card. The MIFARE Plus AES keys can be dumped from the SAM with “must diversified” option, if it is required.

Table 30. Example of MFP_WritePerso

Step	Indication		Data/Message	Comment
1	Activate the card up to ISO/IEC 14443-4 layer (e.g. ISO14443-3_ActivateIdle, ISO14443-4_RATS_PPSRATS)			
2	MFP_WritePerso C-APDU	>	80A800005A00904A5EBE086D7A4E353345614E9B88C87F0190D7B12348ABE1A58AFECC513C713C1BF302903F613B19AE782E989AA5CDA4073BE27B039067B2C4D72DF59C413F8BCDDE9795BE00049086EDB107245EC47045FF88FEB6DB363E00	Here the block numbers 0x9000 to 0x9004 have been written. The LSB of the block number comes first.
3	MFP_WritePerso R-APDU	<	909000	The status code of MIFARE Plus = ‘90’ means successful.

The Commit_Perso command can be issued by using the ISO14443-4_Exchange command.

As the data/keys are transferred in plain to the MIFARE Plus card, it is recommended to perform the “Write Perso” command in a secure site.

3.6.2 MFP_Authenticate

The same command is used for all type of AES authentication in all security level. Set bit number 2 and 3 accordingly for selection of different authentication. In the following, one example is given for authentication in security level 3.

Table 31. Example of MFP_Authenticate

Step	Indication		Data/Message	Comment
	The MIFARE Plus key is stored in SAM key entry number 07 and version 00			
1	MFP_Authenticate C-APDU	>	80700C0005070000400000	P1 = 0C means no diversification, authentication first and SL3 authentication.
2	MFP_Authenticate R-APDU	<	0000000000000000000000009000	PDCap2 (6 bytes) || PCDCap2 (6 bytes) and status 9000.

3.6.3 MFP_CombinedRead

This ‘combined read’ command can read MIFARE Plus block(s). If the access condition allows, the full card can be read in one command.

Table 32. Example of MFP_CombinedRead

Step	Indication		Data/Message	Comment
1	MFP_CombinedRead C-APDU	>	8031000004**31000004**00	The data field contains read command in plain (the MIFARE Plus read is encrypted and CMAC in both direction). Four blocks have to be read starting from block number ‘0000’.
2	MFP_CombinedRead R-APDU	<	9000050001020304184200140111002209A6FE56B361A6595A568401D3597D0A86097D1FA3C8BA056D70D2E9DF3E545502000102030405060708090A0B0C0D0E0F9000	The response contains MIFARE Plus status code (90) and the content of the blocks followed by the SAM status bytes(9000, success).

3.7 Use of Secure Messaging

The communication between SAM and the PICC is secured by the PICC’s security policy and the security between the SAM and the host is ensured by the SAC (Secure authenticated Channel [1]).

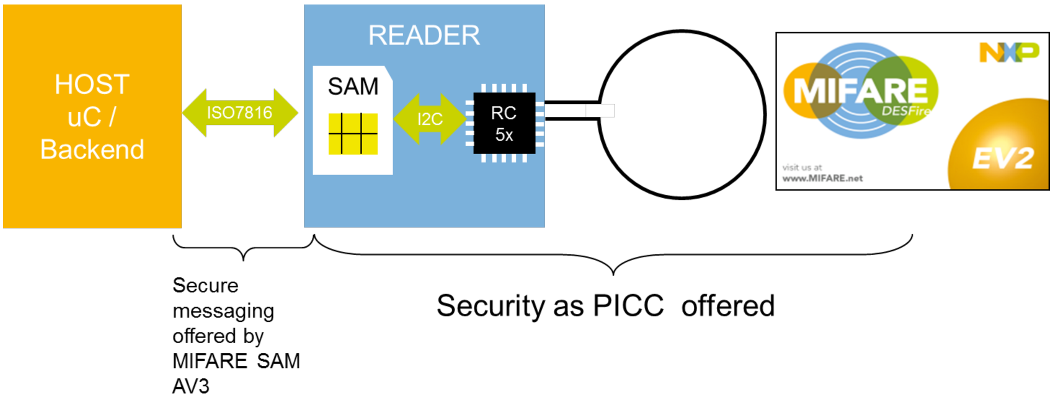


Figure 9. Secure messaging adds security in the communication between SAM and Host

3.7.1 Secure Messaging example for MIFARE DESFire EV1

The logical channel number 0 (CLA = 0x80) is used for this example.

Table 33. Example full protection Host communication for MIFARE DESFire EV1

Step	Indication		Data/Message	Comment
1	Initialize the reader IC and turn on the RF.			
2	Authenticate Host, using SAM_AuthenticateHost command take host mode = full protection. See §2 of [8] for detail calculation. In this example session key were as follows: Encryption session key = 3056A1804B24B44386F5E1032AA206A9 and CMAC session key = D03206A036FB41257A8093DB52A2DBC5			
3	ISO14443-3_ActivateIdle	=	8026010000	The command APDU in plain. It requires now calculation of secure messaging.
4	ISO14443-3_ActivateIdle C-APDU in full protection	>	802601000804FD77D0FAFF11E500	Data field contains CMAC. See §2 of [8] for detail calculation
5	ISO14443-3_ActivateIdle R-APDU in full protection	<	4FE359F6A562BC2E51BA95ED48C9E9F4432959D77D63B69A9000	The response is encrypted with a CAMC.
6	Plain response after CMAC verification and decryption	=	44032007049137C9922680	See §2 of [8] for detail calculation
7	ISO14443-4_RATS_PPS	=	80E000000301000000	The command APDU in plain. It requires now calculation of secure messaging.
8	ISO14443-4_RATS_PPS C-APDU in full protection	>	80E00000181917CFB3C9E585DFA822E3FEC496406247C842647935E3EF00	Data field contains encrypted data and CMAC. See §2 of [8] for detail calculation.
9	ISO14443-4_RATS_PPS R-APDU in full protection	<	983A7DF82021274B40FC3919E00F7269C330BD2316DAD8299000	Response data field contains encrypted data and CAMC
10	Plain response of the card after CMAC verification and decryption	=	010000067577810280	See §2 of [8] for detail calculation.
11	ISO14443-4_Exchange command for application selection	=	80EC0000045A12345600	The command APDU in plain. It requires now calculation of secure messaging.
11	ISO14443-4_Exchange C-APDU for application selection in full protection mode	>	80EC00000002000018B73D246612CF9FB04C61089DBD45DF3A00	Data field contains encrypted data and CMAC. See §2 of [8] for detail calculation.
12	ISO14443-4_Exchange R-APDU in full protection	<	80EC000018B73D246612CF9FB04C61089DBD45DF3A06FD8224F07FFF3800	Response data field contains encrypted data and CAMC

Step	Indication		Data/Message	Comment
13	Plain response of the card after CMAC verification and decryption	=	00	See §2 of [8] for detail calculation.
14	DESFire_AuthenticatePICC command	=	80DA00000303020400	The command APDU in plain. It requires now calculation of secure messaging.
15	DESFire_AuthenticatePICC C-APDU in full protection mode	>	80DA000018A352C73F5AEDBA175FBED58CA83F2500F3616AC0732A74E800	Data field contains encrypted data and CMAC. See §2 of [8] for detail calculation.
16	DESFire_AuthenticatePICC R-APDU in full protection mode	<	2B2972077BE6D0E79000	Only CMAC as, that command has no response data.
17	Verify the CMAC	=	2B2972077BE6D0E7	See §2 of [8] for detail calculation.
18	DESFire_WriteX command in plain	=	80D30038123D010000000A00000102030405060708091000	Writing 10 bytes (01020304050607080910) to file 01 and at offset 0.
19	DESFire_WriteX C-APDU in full protection mode.	>	80D3003828283BB2DBF563F405DDD0AA65E45863CF9C3ADD68667C06CED221652FCB601DF04518399BB15DF57500	See §2 of [8] for detail calculation.
20	DESFire_WriteX R-APDU in full protection mode.	<	0938B4429A7FCDA29000	Only CMAC as, that command has no response data.
21	Verify the CMAC	=	0938B4429A7FCDA2	See §2 of [8] for detail calculation.
22	DESFire_ReadX command in plain	=	80D200300B0A0000BD010000000A000000	Reading 10bytes from file 1 at offset 0.
23	DESFire_ReadX C-APDU in full protection mode.	>	80D20030188EAFB3DF0999FDF926255B661C2411BABA9788D8BB65B88F00	See §2 of [8] for detail calculation.
24	DESFire_ReadX R-APDU in full protection mode.	<	FEBE6CB3F57860A92DFFE7774913D303544C5BDB3B81B2C59000	Response data field = encrypted data and CAMC.
25	After verification of CMAC and decryption	=	01020304050607080910	The data read from the DESFire file.

4 References

1. **Data sheet –** MIFARE SAM AV3, document number DS3235xx.
2. **System guidance manual – MF4SAM30 (MIFARE SAM AV3)**, document number xx.
3. **Data sheet –** MIFARE Plus, document number 1637xx.
4. **Application note** – **AN12695 - MIFARE SAM AV3 –Quick Start up Guide**, document number 5210xx, https://www.nxp.com/docs/en/application-note/AN12695.pdf
5. **Application note** – **AN5212 - MIFARE SAM AV3- Key Management and Personalization**, document number 5212xx.
6. **Application note – Symmetric Key Diversifications**, document number 1653xx.
7. **Application note – AN5217 – MIFARE SAM AV3 for MIFARE Classic,** document number AN5217xx.
8. **Application note – AN12704 – MIFARE SAM AV3 Host communication,** document number 5213xx, https://www.nxp.com/docs/en/application-note/AN12704.pdf
9. **Data sheet –** MFRC523**,** Contactless Reader IC.

5 Legal information

5.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

5.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors. In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory. Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification. Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products. NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer. In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages. Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

5.3 Licenses

ICs with DPA Countermeasures functionality

NXP ICs containing functionality implementing countermeasures to Differential Power Analysis and Simple Power Analysis are produced and sold under applicable license from Cryptography Research, Inc.

5.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

MIFARE — is a trademark of NXP B.V.

DESFire — is a trademark of NXP B.V.

MIFARE Plus — is a trademark of NXP B.V.

MIFARE Ultralight — is a trademark of NXP B.V.

MIFARE Classic — is a trademark of NXP B.V.

Tables

Figures

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 10 January 2020
Document identifier: AN12705
Document number: 521911