AN12704

MIFARE SAM AV3 - Host Communication

Rev. 1.1 — 10 January 2020
521311

Application note
COMPANY PUBLIC

Document information

Information	Content
Keywords	MIFARE SAM AV3, TDEA, AES, RSA, Host communication.
Abstract	This application note addresses different types of communication between host (microcontroller) and MIFARE SAM AV3.

Revision history

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

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[1] securely and to enable secure communication between terminals and host (backend).

1.1 Scope

This application note describes different type of host communication and secure messaging features of MIFARE SAM AV3 (referred to SAM in this document, if not otherwise mentioned) with examples. 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 [3].

This application note is a supplement document for application development using MIFARE SAM AV3. Should there be any confusion please check MIFARE SAM AV3 data sheet [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, application notes or design guides.

1.2 Abbreviation

Refer to Application note “MIFARE SAM AV3 – Quick Start up Guide” [3].

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 received from SAM or PICC (if not mentioned, SAM).

C-APDU:

CLA	INS	P1	P2	Lc	Data (nc)	Le

Table 1. R-APDU:

Response data	SW1	SW2

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 Ultralight C, MIFARE Classic, MIFARE Plus, MIFARE DESFire, MIFARE DESFire EV1

2 Host Communication

MIFARE SAM AV3 can be used in X-interface or in non-X interface architecture. In both cases, a microcontroller (host) is communicating to the SAM. This communication also has an important role to provide the targeted end to end security.

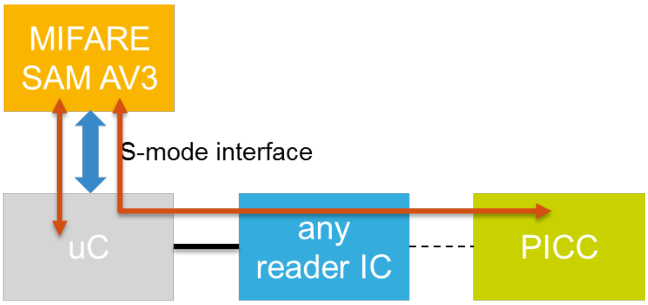

Figure 1. MIFARE SAM AV3 in S interface architecture

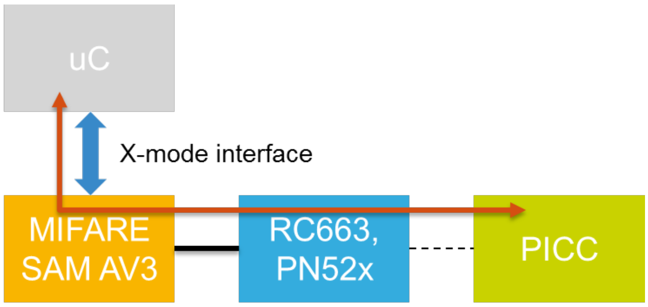

Figure 2. MIFARE SAM AV3 in X interface

The commands of MIFARE SAM AV3 are classified in different sets:

Initial command set (ICS): allowed before SAM activation

Minimal command set (MCS): available after SAM activation, no active host authentication needed, even if SAM is locked

General command set (GCS): These commands may need active authentication. Only available if unlocked

PL command set (PCS): subset of GCS, requires active Host authentication with PLKey.

Limited Command set (LCS): Commands that need an active Host authentication and Key Access Control permission.

Restricted command set (RCS): These commands need active host authentication in each logical channel.

See the detail in [1].

In the following diagram the usual host communication structure is shown:

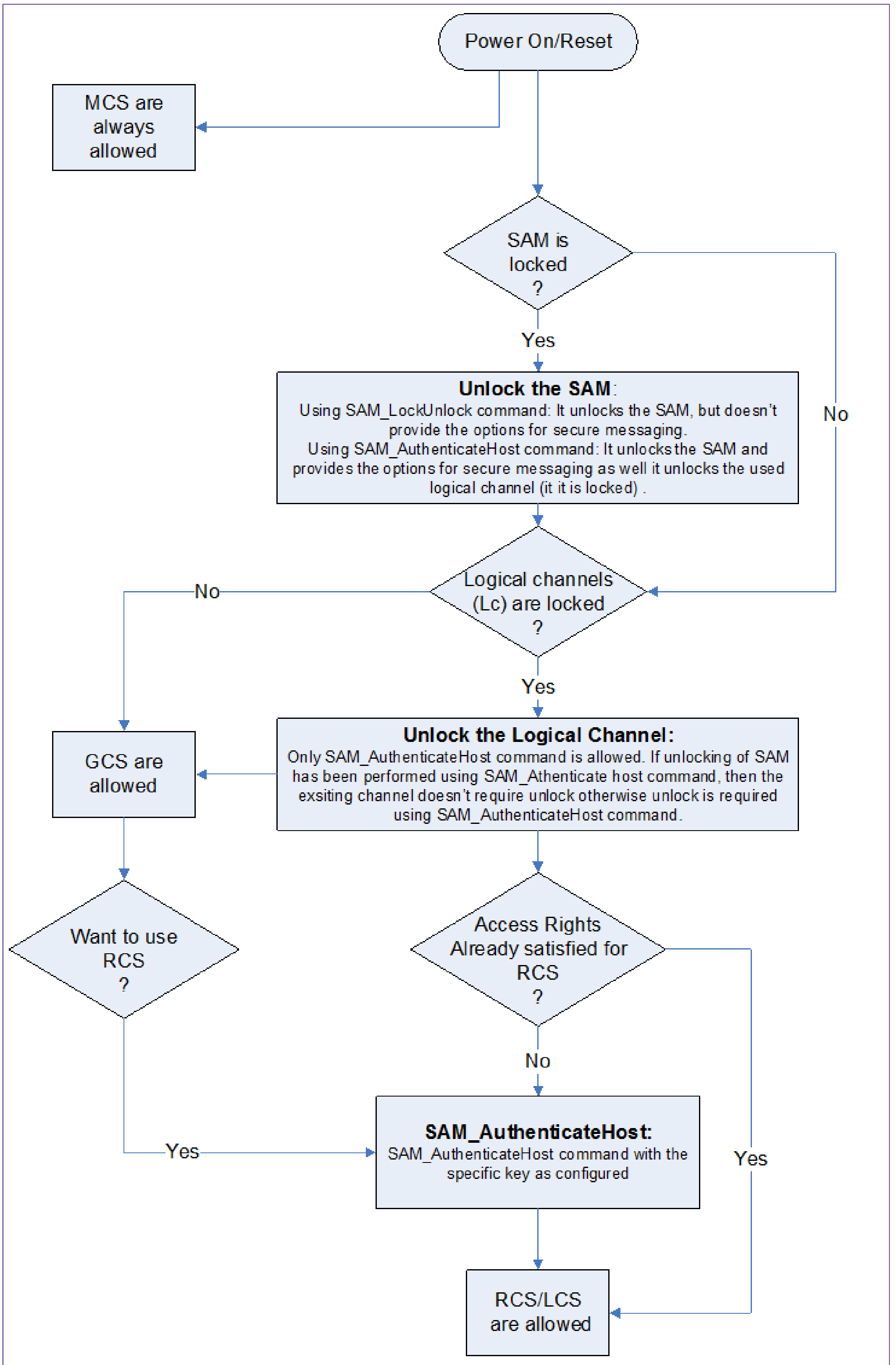

Figure 3. Usual Communication from the host

The host communication can be one of the three types:

1. Plain
2. MAC Protection
3. Full Protection (encrypted communication)

The type of communication at that channel is defined by previous host authentication command.

2.1 Host Authentication for full protection

This option will enable the encrypted communication between SAM and Host.

2.1.1 Example - SAM_AuthenticateHost Command

Secret key (Kx) = 000102030405060708090A0B0C0D0E0F.

Table 2. Example - SAM_AuthenticateHost for full protection mode

step	Indication		Data / Message	Comment
1	C-APDU, part 1	>	80A400000305010200	Key nr = 0x05; key version = 0x01; Host mode = Full protection.
2	R-APDU	<	2509C7B09F2DA8FF6D765 78B90AF	Rnd2 + SW
3	Rnd2	=	2509C7B09F2DA8FF6D765 78B	
4	CMAC load for part 2 C-APDU	=	2509C7B09F2DA8FF6D765 78B02000000	Rnd2 + HostMode + padding
5	CMAC	=	9D2231E7B99F0CFF	8-byte CMAC calculated on step 4 data using Secret key (Kx). (Every odd byte (start from 0) from 16-byte standard CMAC).
6	Rnd1	=	000102030405060708090A 0B	12-byte random 1 generated by reader. (Take real random).
7	C-APDU, part 2	>	80A40000149D2231E7B99 F0CFF00010203040506070 8090A0B00	Data field contains CMAC of step 5 + Rnd1
8	SV1	=	0708090A0BFF6D76578B2 508C5B39B91	Rnd1(byte 7-13) + Rnd2(byte 7-13) + (Rnd1(byte 0-4)Xor Rnd2(byte 0-4))+91.
9	Kxe	=	7FB7B598D7E5045743809 9994907A2F0	Encryption of SV1 using secret key (Kx).
10	R-APDU	<	**E89F438446F5177E** 03322788AE6DB98C963E12 C6DF1F401990AF	8-byte CMAC + encrypted RndB with Kxe (calculated in step 9)+ SW(90AF).
11	CMAC load in last R-APDU	=	000102030405060708090A 0B02000000	Rnd1 + HostMode+padding

step	Indication		Data / Message	Comment
12	CMAC	=	**E89F438446F5177E**	8-byte CMAC calculated on step 11 data using Secret key (Kx). (Every odd byte (start from 0) from 16-byte standard CMAC). This calculated CMAC must be equal to received CMAC in last R-APDU.
13	RndB	=	B4FFEAA4B4293B6D2077A172E095C819	Decrypt the RndB received in last R-APDU using Kxe.
14	RndA	=	000102030405060708090A0B0C0D0E0F	16-byte random generated by reader. (Take real random).
15	RndB´´	=	EAA4B4293B6D2077A172E095C819B4FF	Rotate left RndB by two bytes.
16	RndA+RndB´´	=	000102030405060708090A0B0C0D0E0FEAA4B4293B6D2077A172E095C819B4FF	Concatenate RndA and RndB´´.
17	Ek(Kxe, RndA +RndB´´)	=	9379F61F1D6EB335803343620CE9AD045C672F4E8A66666527384A4DB251F455	Encrypt RndA+RndB´´ using Kxe.
18	C-APDU, part 3	>	80A40000209379F61F1D6EB335803343620CE9AD045C672F4E8A66666527384A4DB251F45500	Data filed is Ek(Kxe, RndA+RndB´´)
19	RndA´´	=	**02030405060708090A0B0C0D0E0F0001**	Rotate RndA left by two bytes.
20	R-APDU	<	F261C8E49E275A46E210899B3EFD0D589000	Rotate RndA left by two bytes.
21	Dk(Kxe, Ek(Kxe, RndA´´))	=	**02030405060708090A0B0C0D0E0F0001**	Ek(Kxe, RndA´´) + SW. Status 9000 means “SAM confirms authenticity”. Now the reader must check if it is finally ok from his side.
22	Dk(Kxe, Ek(Kxe, RndA´´)) = RndA´´	=	Yes (step 19 value = step 21 value).	Decrypt Ek(Kxe, RndA´´) using Kxe.
23	SVKe	=	0B0C0D0E0F72E095C819B02C3D6A2881	RndA byte 11 to 15 + RndB byte 11 to 15 + ((RndA byte 4 to 8) XOR (RndB byte 4 to 8)) + 81;
24	SVKm	=	0708090A0B6D2077A172B4FEE8A7B082	RndA byte 7 to 11 + RndB byte 7 to 11 + ((RndA byte 0 to 4) XOR (RndB byte 0 to 4)) + 82;
25	Session key encryption (Ke)	=	F7B5D7E05FCDA9F12D6F106CB483B66A	Ek(Kx, SVKe).
26	Session key MAC (Km)	=	10CDA5E6BF15A309C4DA69C85B9AACBA	Ek(Kx, SVKm).

2.2 Full protection communication

Refer to [1] for all detail of the command and response message structure. Here one example for changing key in full protection mode is shown.

2.2.1 Example – SAM_ChangeKeyEntry Command in full protection

Key entry number 0x17 will be changed to AES -128. Current session keys are as follows:

Ke = 092D5F2AA78F5A22B5F5A01F931A83FB.

Km = 2CA7ADBD4969DD3F22BEC6B5C39952CA.

Table 3. Example - SAM_ChangeKeyEntry in full protection mode

step	Indication		Data / Message	Comment
1	New PosA key	=	0102030405060708091011 1213141516	16-byte key
2	New PosB key	=	00112233445566778899AA BBCCDDEEFF	16-byte key
3	New PosC key	=	ABCDEF012345678990817 263545E740F	16-byte key
4	DF_AID	=	000000	DESFire Application ID.
5	DF_KeyNo	=	00	DESFire key number.
6	KeyNoCEK	=	00	This key entry update will require SAM_AuthenticateHost with key entry number 00.
7	KeyVCEK	=	00	This key entry update will require SAM_AuthenticateHost with the key of entry number 00 which has version 01.
8	RefNoKUC	=	02	New key entry number 1 is linked to counter number 2. If no counter is used set this value to FF.
9	SET	=	2001	Key type AES-128, b8 must be set for host key (except Key entry number 0 if not individual LC locking is required).
10	Version of key PosA	=	00	Version number of key position A, can be any value from 00 to 0xFF.
11	Version of key PosB	=	01	Version number of key position B, can be any value from 00 to 0xFF.
12	Version of key PosC	=	02	Version number of key position C, can be any value from 00 to 0xFF.
13	ExtSET	=	00	Host key
14	New key entry data	=	0102030405060708091011 1213141516001122334455 66778899AABBCCDDEEFF ABCDEF012345678990817 263545E740F00000000000 102200100010200	Concatenate all from step 1 to 13.

step	Indication		Data / Message	Comment
15	C-APDU in plain mode	=	80C117FF3D01020304050 60708091011121314151600 0112233445566778899AAB BCCDDEEFFABCDEF0123 456789908172635445E740F 00000000000000220010001 0200	P1= 17, as key entry number 0x17 is changed.
16	IV Load for command	=	01010101000000000000000 0000000000	First 4-byte 0x01, 3 times 4-byte counter. Here the value is 00000000.
17	IV	=	E54FEC4863DD3F1BFD2D CD813799144D	Ek(Ke, IV load). Encryption of IV load using the encryption session key.
18	Data to encrypt	=	01020304050607080910111 121314151600112233445 5 66778899AABBCCDDEEFF ABCDEF01234567899081 7 263545E740F00000000000 00220010001020080000 0	Data, in this case from step 14 and padding (started with 80).
19	Encrypted data	=	4DC47E96DB150A861C93 2BC74010E5F9BE644C408 9E08F9AE05CE76E5FA8E B9BECA650452E1212FEB3 A3DD9A03EE8972A0D380 83DEA40C69834A2EEDEF A3E407	Data of step 18 is encrypted using the encryption key and IV (here from step 17).
20	CMAC load	=	80C10000000017FF484DC 47E96DB150A861C932BC 74010E5F9BE644C4089E0 8F9AE05CE76E5FA8EB9B ECA650452E1212FEB3A3 DD9A03EE8972A0D38083 DEA40C69834A2EEDEFA3 E407	CLA + INS + 4-byte counter + P1 + P2 + Lc (data length + 8 for CMAC) + encrypted data (step 19)
21	CMAC	=	47E9C5F61CF0D242	8-byte CMAC calculated on step 20 data using CMAC session key (Km). MIFARE Plus specific CMAC.
22	C-APDU	>	80C117FF48 DC47E96DB150A861C932 BC74010E5F9BE644C4089 E08F9AE05CE76E5FA8EB 9BECA650452E1212FEB3A 3DD9A03EE8972A0D38083 DEA40C69834A2EEDEFA3 E40747E9C5F61CF0D242	Fully protected command sent to the SAM.
23	R-APDU	<	AA60E01E86561A6F9000	Answer from the SAM. CMAC and success.
24	Calculate CMAC	=	AA60E01E86561A6F	CMAC load = 900000000001 (SW1SW2 and the return counter which is 1 more than the counter sent in the command).

step	Indication		Data / Message	Comment
25	CMAC? received CMAC	=	Yes from step 23 and 24	OK

2.2.2 Some examples of Secure Message calculation

In this example session key were as follows:

Encryption session key (Ke) = 3056A1804B24B44386F5E1032AA206A9 and

CMAC session key (Km) = D03206A036FB41257A8093DB52A2DBC5

Table 4. Example - Data field is absent in the C-APDU

step	Indication		Data / Message	Comment
1	C-APDU	=	8026010000	ISO14443-3_ActivateIdle command
2	IV load	=	01010101000000000000000000000000	
3	IV	=	EE861B62816BD21BFF2C5F66F77A1F02	En(Ke, IV load)
4	CMAC load	=	80260000000001000800	After inserting the current counter between INS and P1.
5	CMAC	=	04FD77D0FAFF11E5	MIFARE Plus specific CMAC.
6	C-APDU	>	802601000804FD77D0FAFF11E500	Secure, the data field contains CMAC.
7	R-APDU	<	4FE359F6A562BC2E51BA95ED48C9E9F4**432959D77D63B69A**9000	Encrypted data = 4FE359F6A562BC2E51BA95ED48C9E9F4 CMAC = 432959D77D63B69A
8	CMAC load	=	9000000000014FE359F6A562BC2E51BA95ED48C9E9F4	SW1SW2+Counter (cmd ctr +1) + encrypted data
9	Calculated CAMC	=	**432959D77D63B69A**	Calculated CMAC = received CMAC, so the data integrity is ok.
10	IV load for decrypting response data	=	02020202000000010000000100000001	Response IV load = 02020202+3 times (cmd ctr+1)
11	R-APDU	<	4FE359F6A562BC2E51BA95ED48C9E9F4**432959D77D63B69A**9000	Encrypted data = 4FE359F6A562BC2E51BA95ED48C9E9F4 CMAC = 432959D77D63B69A
12	CMAC load	=	9000000000014FE359F6A562BC2E51BA95ED48C9E9F4	SW1SW2+Counter (cmd ctr +1) + encrypted data
13	Calculated CAMC	=	**432959D77D63B69A**	Calculated CMAC = received CMAC, so the data integrity is ok.
14	IV load for decrypting response data	=	02020202000000010000000100000001	Response IV load = 02020202+3 times (cmd ctr+1)

step	Indication		Data / Message	Comment
15	IV	=	8BF04E917C4CE7883CD6 E5A0D609DC76	En(Ke, IV load).
16	Decryption of the encrypted data using the Ke and IV.	=	44032007049137C9922680 8000000000	Card response and padding

Table 5. Example - Data field is present

step	Indication		Data / Message	Comment
1	C-APDU	=	80E000000301000000	ISO14443-4_RATS_PPS
2	IV load	=	0101010100000001000000 0100000001	01010101+3 times cmd ctr
3	IV	=	0CB765AA23EC38690D797 148E0C882F7	En(Ke, IV load), IV = 00s
4	Encryption data load	=	0100008000000000000000 0000000000	Data field of step 1 and padding.
5	Encrypted data	=	1917CFB3C9E585DFA822E 3FEC4964062	En(Ke, encryption data load of step 4), using the IV of step 3.
6	CMAC load	=	80E0000000010000181917 CFB3C9E585DFA822E3FE C496406200	After inserting the current counter between INS and P1 + encrypted data + Le.
7	CMAC	=	47C842647935E3EF	MIFARE Plus specific CMAC.
8	C-APDU	>	80E00000181917CFB3C9E 585DFA822E3FEC4964062 47C842647935E3EF00	Secure, the encrypted data and CMAC.
9	R-APDU	<	983A7DF82021274B40FC3 919E00F7269**<u>C330BD2316 DAD829</u>**9000	Encrypted data = 983A7DF82021274B40FC3919E00 F7269 CMAC = C330BD2316DAD829
10	CMAC load	=	900000000002983A7DF820 21274B40FC3919E00F7269	SW1SW2+Counter (cmd ctr +1) + encrypted data
11	Calculated CAMC	=	**<u>C330BD2316DAD829</u>**	Calculated CMAC = received CMAC, so the data integrity is ok.
12	IV load for decrypting response data	=	0202020200000002000000 0200000002	Response IV load = 02020202+3 times (cmd ctr+1)
13	IV	=	ACC2959268A3CF105E8E7 9D0C9F208DF	En(Ke, IV load).
14	Decryption of the encrypted data using the Ke and IV.	=	0100000675778102808000 0000000000	Card response and padding

2.3 Secure Authenticated Channel

The secure authenticated logical Channel (SAC) can be used to protect the plain data (PICC or any other data) exchanged between Host and SAM. In this case, the channel is becoming a secure pipe-line.

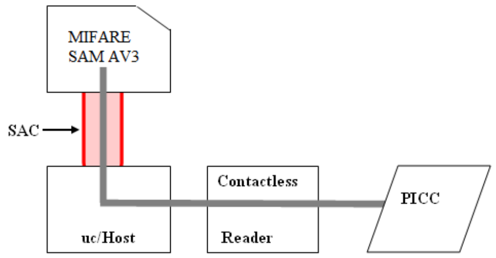

Figure 4. SAC in non-X interface

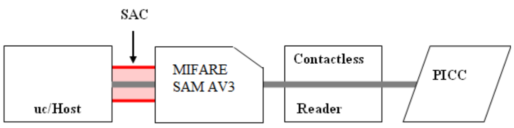

Figure 5. SAC in X-interface

In the same logical channel, there can be host authentication as well as authentication for PICC at the same. At the same time, only one host authentication per channel is allowed. The data exchanged between host and PICC (in non-X interface) and SAM and PICC (in X interface) is secure by the PICC's crypto mechanism. The plain data exchanged between host and SAM can be now protected using the secure messaging supported by SAC.

2.4 Host Communication via the I2C Slave interface

As an alternative to the UART-based ISO 7816 T=1 communications, the SAM AV3 will support an I2C Slave interface to replace the ISO 7816-3 low-level UART character transmission. The I2C Slave interface will support the existing ISO 7816 block interface. The I2C slave interface will be compatible with the I2C standards for transmission and reception of bits, bytes and blocks as detailed in the I2C-bus specification [2].The major difference between ISO7816 and I2C is that the ISO7816 interface is asynchronous and the SAM AV3 can send a response at any time up to the BWT after receiving a T=1 command from the Host. For I2C Slave the communication is synchronous, therefore the SAM AV3 can only return data as part of a READ request from the Host Master. The procedure for returning responses from the SAM AV3 in I2C Slave mode is detailed in section Section 2.4.4.

2.4.1 I2C Slave Address (SLAD)

The SAM AV3 will use the 5-bit NXP device bus address 01010b, with the 2 remaining bits set to 11b, to support an address byte of 0101011 | (R/W).

SLAD(R) = 0101011**1**

SLAD(W) = 0101011**0**

2.4.2 Transmission and Reception of APDUs with I2C

For commands, the I2C interface follows the T=1 interface closely, with the following bytes transmitted by the Host.

SLAD (W) | NAD | PCB | LEN | DATA (LEN) | LRC ->

With the SLAD(W) byte indicating an I2C WRITE operation.

The Host must transmit the SLAD byte to receive any data from the SAM AV3.

SLAD (R) ->

<- NAD | PCB | LEN | DATA (LEN) | LRC

With the SLAD(R) byte indicating a I2C READ operation

2.4.3 I2C Slave Control Bytes (T=1 PCB)

As there is a requirement to transmit control information as well and command and response data, a CTRL byte precedes any data sent by the I2C Host Master and returned by the SAM AV3. The CTRL bytes are defined below in Table 6. The CTRL are compatible with the T=1 Protocol (PCB) byte definition.

Table 6. I2C PCB Byte Definition

Label	Value	Use
I-Block	0nm00000b	Command / Response Data, (I-BLOCK) **n**…N(S), is the send sequence number. It is initialized as 0, and is to be toggled for every subsequent I_Block sent. **m**…more data bit. If set to 1, more data is available, and block chaining shall be used.
S-WTX	0xC3	WTX Request from the SAM for more processing time
S-IFS	0xC1	INF request to change the IFS size.

2.4.4 I2C Slave Response Polling

For Master Polling, the HOST will periodically poll the SAM AV3 with a SLAD (R) sequence and look for a valid response. If there is no ACK the HOST can assume the SAM AV3 is still processing. If the SAM AV3 sends a correct ACK, it will respond with the command response frame or a WTX frame requesting more processing time. The HOST should poll the SAM AV3 in a relatively short time interval to minimize the time between SAM AV3 processing completion and returning the command response. As the I2C interface is not interrupt-driven, there will be no processing overhead for the SAM for each poll request from the HOST.

Clock Stretching is not used for the MIFARE SAM AV3.

2.4.5 I2C WTX Requests

Like ISO7816 T=1, the MIFARE SAM AV3 will support a WTX mechanism to request additional processing time from the HOST. In this case the MIFARE SAM AV3 will respond to a poll with a 7816 T1 WTX frame.

For a successful WTX request, the following data will be exchanged.

SLAD (R) ->

<-NAD | 0x83 | 01 | WTX | LRC - WTX request from the SAM AV3

SLAD (WR) NAD | 0xC3 | 01 | WTX | LRC – WTX response from the host

This implies that the HOST must poll the slave by performing an I2C start and device select. An ACK indicates that the slave has stopped internal processing and is ready to perform a communication sequence. A NACK indicates that the slave is busy performing internal processing.

2.4.6 ATR

The SAM AV3 ATR is returned after Reset. For I2C mode, the ATR availability is indicated by an ACK being detected following a device reset.

If the ATR is not yet available:

SLAD (R) ->

NACK

If the ATR is available:

SLAD (R) ->

ACK

<- ATR

2.4.7 Information Field Size (INF)

Like T=1, by default the maximum size of the information that can be received by the SAM AV3 is set to 32 bytes following a reset. This is the default value used to initialize IFSC and IFSD. These values can be increased using S(IFS request) and S(IFS response) command frames.

To increase the value of IFSD the following command frame can be used.

SLAD(W) | NAD |S(IFS request) | 01 | INF | LRC ->

SLAD(R) ->

<- NAD |S(IFS response) | 00 | LRC

The maximum size of INF supported is 251, as the I2C Slave library code used in the SAM AV3 development is limited to 255 bytes.

3 References

1. **Data sheet –** Data sheet of MIFARE SAM AV3, document number DS3235xx.
2. **The I2C Specification and User Manual –** UM10204, http://www.nxp.com/documents/user_manual/UM10204.pdf
3. **Application note – AN12695 MIFARE SAM AV3 – Quick Start up Guide**, document number 5210xx, https://www.nxp.com/docs/en/application-note/AN12695.pdf
4. **Application note – Symmetric Key Diversifications**, document number 1653xx.
5. **System guidance manual – MF4SAM30 (MIFARE SAM AV3)**, document number xx.
6. **Application note – MIFARE SAM AV3 for MIFARE Plus EV1,** document number 1825xx
7. **Application note – MIFARE SAM AV3 for MIFARE DESFire EV2,** document number 1826xx

4 Legal information

4.1 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.

4.2 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.

4.3 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.

4.4 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

Date of release: 10 January 2020
Document identifier: AN12704
Document number: 521311