AN10787

MIFARE Application Directory (MAD)

Rev. 07 — 7 July 2010 **Application note**

Document information

Info	Content
Keywords	MIFARE Application Directory (MAD), multi-application, function cluster code, application code, General Purpose Byte (GPB), CRC.
Abstract	Presenting the proposed MIFARE Application Directory, its rule and structure together with examples, which opens the possibility to combine different applications in one card with certain interoperability.

Revision history

Rev	Date	Description
07	20100707	Modifications: • Section 3.12 “MAD and MIFARE Plus”: added • Table 16 “Function cluster codes”: updated • Section 4.5 “MIFARE standardization group and registration authority”: web link updated • Section 10.2 “Disclaimers”: updated
06	20091204	Modifications: • Table 13 “MIFARE DESFire AID”: updated • Section 3.10.1 “Example”: updated
05	20091013	Modifications: • Section 3.10.1 “Example”: section added • Section 10 “Legal information”: updated
04	20090305	Fourth release (supersedes AN MAD, MIFARE application directory, Rev. 03.00, 4 May 2007)

Contact information

For more information, please visit: **http://www.nxp.com**

For sales office addresses, please send an email to: **salesaddresses@nxp.com**

1. Introduction

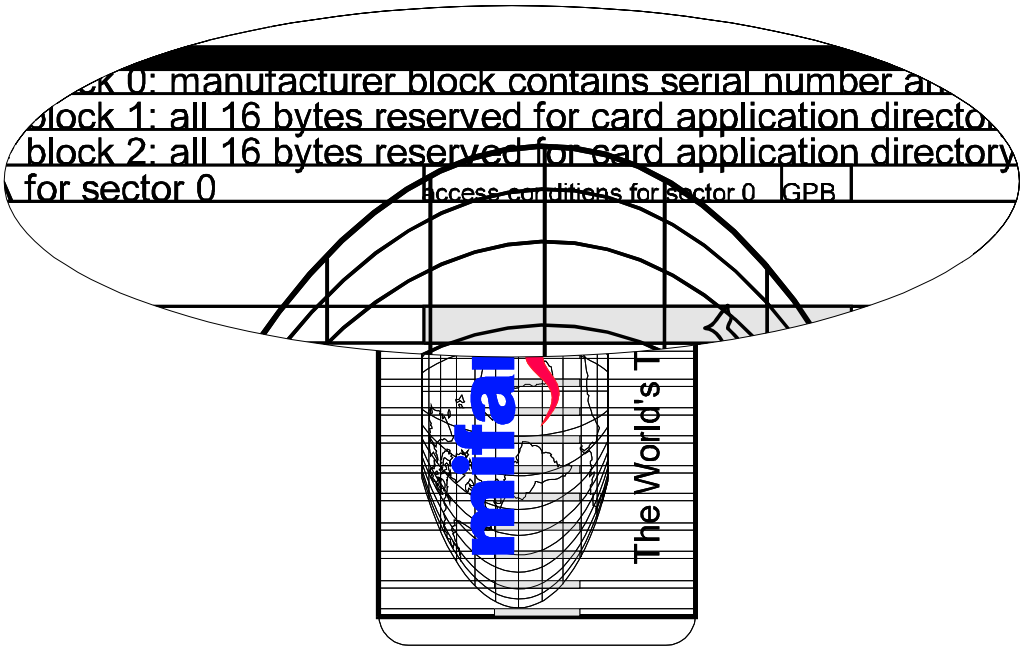

Fig 1. MIFARE Application Directory

The MIFARE Application Directory standard proposes the introduction of common data structures for card application directory entries. Registered application identifiers (AIDs) in sector 0x00 (and sector 0x10, if applicable) of any MIFARE card enable identification of all registered card applications. Terminal software should take advantage of this feature using those sector pointers instead of physical sector addresses.

In the future it might easily happen that there are more than one MIFARE card in a person's wallet. The comfort of not having to take out the card of one's wallet should be possible also with more MIFARE cards in one wallet. A typical case can be that one person has cards for different applications (e.g. airline miles collection and city fare collection). With the MAD the airline check-in terminal identifies two cards and is able to choose the correct one very fast, simply by checking the MAD.

The current document describes the MAD version 1, 2 and 3.

MAD1 is limited to 16 Sectors (as used in MIFARE Classic).

MAD2 specifies the usage of the MIFARE with a memory >1k (e.g. MIFAREPro and MIFARE ProX, MIFARE 4k, etc.).

MAD2 is fully compatible to the MAD1, i.e. an MAD1 system can use cards, that use MAD2 without any changes. In this case only the lower 1k EEPROM can be addressed.

All the relevant changes are described on Table "Revision history" on page 2.

MAD3 specifies the usage of Registered application identifiers in the context of MIFARE DESFire.

Observing the following proposed MIFARE Application Directory rules following proposed opens a lot of future benefits:

Table 1. Future benefits

basic requirements	⇒ additional information	⇒ additional flexibility
• reserve 2 blocks in sector 0(and also reserve 3 blocks in sector 16 for MAD2) • keep the given format • request for AID[1] • use public read-key for sector 0 • use secret write-key for sector 0 • use indirect addressing mode in terminal program	⇒identify any application on any MIFARE card together with the sectors in use ⇒identify card issuer ⇒identify free or blocked sector	⇒already existing MIFARE cards may serve for new additional applications ⇒already existing MIFARE applications on multiple cards may be combined on one single card ⇒easy adaptation of memory structure in case of additional features or blocked sectors

[1] AID application identifier request form can be found in annex A

2. Data elements for application directories and selection

2.1 Application identifier

Is a unique 16 bit code divided into two fields:

Table 2. Application identifier

bit 15							bit 8	bit 7							bit 0
(8 bit)function cluster code								(8 bit)application code							

To enable easy classification of the whole range of possible applications the function cluster code is used. Some codes are already prepared and outlined in annex C.

2.2 CRC-byte

Table 3. CRC-byte

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

8 bits include a cyclic redundancy code according to the 8 bit CRC coprocessor. The coprocessor should be reset and afterwards either the Info-byte and ID1 to ID$F (sector 0x00) or Info-byte and ID$11 to ID$27 (sector 0x10) (lower byte followed by higher byte) should be passed to the CRC coprocessor **exactly in this order**. This code allows an integrity check of the directory blocks.

2.3 Info-byte

Table 4. info-byte

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
free		pointer to CPS					

The information of the card publisher sector is particularly useful if somebody needs to find out the organization responsible for distribution of free card sectors for new applications. These free card sectors may easily be used for additional applications.

Bit 0 … 5pointer to card publisher sector (see Section 3.8)

0x10 shall not be used.

0x28 … 0x3F shall not be used.

Bit 6, 7RFU (reserved for future use)

2.4 General purpose byte (GPB)

The general purpose byte of the access condition field of sector trailer 0 describes further details of the MAD standard. It is the 10th byte of block 3. The code 0x69 should not be used for standardized cards and refers to non-personalized cards.

Table 5. General purpose byte

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
DA	MA	RFU				ADV	

ADV(MAD version code): 01 for MAD version 1 (Sectors 1 … 00xF)

10 for MAD version 2 (Sectors 1 … 0x27)

MA (multiapplication card) 1yes

0 monoapplication card

DA (MAD available) 1yes

0 sector 0 does not contain MAD (all further MAD conventions are not considered)

The GPB for MAD version 2 in sector 16 will be set to RFU (0x00).

2.5 Read-key A

Table 6. Read-key A

Key A of sector 0 should be public and set to the following code:	byte 5	byte 4	byte 3	byte 2	byte 1	byte 0
	a5	a4	a3	a2	a1	a0

2.6 Write-key B

Key B of sector 0 is programmed by the card issuer and should be kept secret. If additional applications join the same MIFARE card key B may be forwarded to the organization which provides the new services in order to enable directory (MAD) adaptation during re-initialization of the MIFARE cards.

3. Coding of the application directories

3.1 MAD version numbers

This standard proposes MAD version 1, 2 and 3.

For MAD1 and MAD2 the version number is encoded in the GPB (see chapter General purpose byte (GPB)). For MAD3 the version number is coded in a special file (see chapter MAD and MIFARE DESFire). For future MIFARE cards this MAD standard may change together with the version numbering.

3.2 MAD types

This standard allows 3 types of MAD:

- monoapplication card without directory entries
- monoapplication card with directory entries
- multiapplication card with directory entries

The MAD type is encoded in the GPB (see Section 2.4).

3.3 Function clusters

Function cluster codes enable easy classification of applications. Currently used codes may be found in annex C. Any organization requesting for a new AID may suggest a code out of this list. If this information is missing the registration authority will determine the code.

3.4 Administration codes

Function cluster code 00 hex assigns specific administration codes to the corresponding sector:

AID - administration codes:

00 00 hexsector is free

00 01 hexsector is defect, e.g. access keys are destroyed or unknown

00 02 hexsector is reserved

00 03 hexsector contains additional directory info (useful only for future cards)

00 04 hexsector contains card holder information in ASCII format.

00 05 hexsector not applicable (above memory size)

3.5 Card holder information

The administration code 0x00 0x04 indicates to public card holder information in the corresponding sector. There is no binding rule but just the following recommendation given for storing card holder information using RLC (Run-Length-Coding):

Table 7. Card holder information

				bit7	bit0
byte n	**byte n-1**	**.....**	**byte 1**	**byte 0**	
00	last character		character 1	type	length<n>

byte 0:length= lower 6 bit (number of used bytes including 0x00, max. 63)

type = highest 2 bit (00=surname; 01=given name; 10=sex; 11=any other data)

byte 1 to <n>:ASCII text as specified in type (first character at byte 1; ends with 0x00)

Unused bytes should be set to 0x00. For storing the sex the following convention is suggested - use „m“ (code 0x6D) for masculine and „f“ (code 0x66) for feminine. In case of insufficient storage space in one sector the card holder information may be continued in the next sector referenced by the administration code 0x00 0x04.

e.g:surname:Sampleman

given name:Philip

masculin

Tel+1/1234/5678

all data is readable with key A but key B is necessary for writing.

The hexadecimal contents of the corresponding sector should look like this:

Table 8. Hexadecimal contents

byte 15	byte 14	byte 13	byte 12	byte 10	byte 10	byte 9	byte 8	byte 7	byte 6	byte 5	byte 4	byte 3	byte 2	byte 1	byte 0
6C	69	68	50	47	00	6E	61	6D	65	6C	70	6D	61	53	0a
33	32	31	2F	31	2B	6C	65	54	D0	00	6D	82	00	70	69
00	00	00	00	00	00	00	00	00	00	38	37	36	35	2F	34
s	e	c	r	e	t	69	88	77	78	a5	a4	a3	a2	a1	a0

The card issuer is responsible for appropriate key protection of card administration sectors. It is advisable to protect all sectors of the card against unauthorized writing with secret keys B. This is recommended even for free and unused sectors.

In special cases, for example when storing public card holder information this data may be released for public reading using the default key A: a0a1a2a3a4a5 hex.

3.6 MIFARE Application Directory (MAD structure)

The location of each AID points to a specific sector on the card.

The location of an AID within sector 0 specifies the sector in use for the corresponding application.

Schematic of sector 0:

Table 9. Schematic of sector 0

byte 15	byte 14	byte 13	byte 12	byte 11	byte 10	byte 9	byte 8	byte 7	byte 6	byte 5	byte 4	byte 3	byte 2	byte1	byte 0
m	a	n	u	f	a	c	t	u	r	e	r	c	o	d	e
AID for sector 0x07		AID for sector 0x06		AID for sector 0x05		AID for sector 0x04		AID for sector 0x03		AID for sector 0x02		AID for sector 0x01		info	CRC
AID for sector 0x$F		AID for sector 0x$E		AID for sector 0x$D		AID for sector 0x$C		AID for sector 0x$B		AID for sector 0x$A		AID for sector 0x09		AID for sector 0x08	
s	e	c	t	o	r	t	r	a	i	l	e	r	0x	0	0

Table 10. Schematic of sector 0x10 of MIFARE 4k card (MAD version 2)

byte 15	byte 14	byte 13	byte 12	byte 10	byte 10	byte 9	byte 8	byte 7	byte 6	byte 5	byte 4	byte 3	byte 2	byte 1	byte 0
AID for sector 0x17		AID for sector 0x16		AID for sector 0x15		AID for sector 0x14		AID for sector 0x13		AID for sector 0x12		AID for sector 0x11		info	CRC
AID for sector 0x1F		AID for sector 0x1E		AID for sector 0x1D		AID for sector 0x1C		AID for sector 0x1B		AID for sector 0x1A		AID for sector 0x19		AID for sector 0x18	
AID for sector 0x27		AID for sector 0x26		AID for sector 0x25		AID for sector 0x24		AID for sector 0x23		AID for sector 0x22		AID for sector 0x21		AID for sector 0x20	
s	e	c	t	o	r	t	r	a	i	l	e	r	0x	1	0

The info byte structure is same as in info byte structure of MAD1. If one more sector is required for information, then lowest 6 bits can be used to code the new sector number, otherwise info byte of sector 0x00 = info byte of sector 0x10.

3.7 CRC calculation

Byte 0 of block 1 of Sector 0 (MAD1, MAD2) and Sector 0x10 (MAD2) will contain 8 bit cyclic redundancy code (CRC). It is generated at the generation of the MAD.

This code should be checked whenever the MAD is read in order to ensure data integrity. Both for the CRC generation and the CRC check the internal CRC coprocessor of the MIFARE“ reader ASIC may be used. Actually the mif_calc_crc() function from the MIFARE“ LowLevelLibrary allows an easy calculation of the CRC code.

For the CRC-calculation of Sector 0 the Info byte should be processed first, then ID1, ID2 … ID0xE, ID0xF in this order.

For the CRC-calculation of Sector 0x10 the Info byte should be processed first, then ID0x11, ID0x12 … ID0x26, ID0x27 in this order.

Always process the lower byte first within the AID´s followed by the higher byte. That means the following process order:

Sector 0x0:block 1, byte 1 to byte 0xF; block 2, byte 0 to byte 0xF

Sector 0x10:block 0, byte 1 to byte 0xF; block 1, byte 0 to byte 0xF, block 2, byte 0 to byte 0xF

Of course the calculation can also be achieved via appropriate software.

8 bit CRC uses the polynomial:x8 + x4 + x3 + x2 + 1and is preset with 0xE3

example for CRC calculation with a sample MAD (hex values):

Table 11. CRC calculation

byte 15	byte 14	byte 13	byte 12	byte 10	byte 10	byte 9	byte 8	byte 7	byte 6	byte 5	byte 4	byte 3	byte 2	byte 1	byte 0
AID for sector 7		AID for sector 6		AID for sector 5		AID for sector 4		AID for sector 3		AID for sector 2		AID for sector 1		info	CRC
00	04	00	00	00	00	00	00	08	01	08	01	08	01	01	89
AID for sector $F		AID for sector $E		AID for sector $D		AID for sector $C		AID for sector $B		AID for sector $A		AID for sector 9		AID sector 8	
30	11	00	00	00	00	00	00	10	02	10	02	10	03	10	03

3.8 Pointer to card publisher sector

This information is particularly useful if somebody needs to find out the organization responsible for distribution of free card sectors for new applications. These free card sectors may easily be used for additional applications.

The lower 6 bits (4bits for MAD1) of the Info-byte contain a binary pointer to one of the 38 sectors in use (15 sectors for MAD1). The owner of the corresponding sector is considered to be the card publisher, responsible for card issue, card maintenance and also for maintenance of the MAD. 0x00 should be used if the card publishing organization does not use any sector on the MIFARE“ card.

0x10 shall not be used.

0x28 … 0x3F shall not be used.

3.9 Key protection of MAD

Block 3 of sector 0 (MAD1, MAD2) and block 3 of sector 0x10 (MAD2) contain key information as well as access condition information. The MAD should be well write-protected with a secret key B defined by the card issuer. Anybody should be allowed to read the MAD. This is achieved by using a public read key A (for sector 0 and sector 0x10, if applicable):

key A: a0a1a2a3a4a5 hex

Access conditions should allow reading with key A|B and writing with key B. According to the MIFARE card product specification this means the following code:

C1X0 C2X0 C3X0: x x x(don´t care for manuf.code)

C1X1 C2X1 C3X1: 1 0 0

C1X2 C2X2 C3X2: 1 0 0

C1X3 C2X3 C3X3: 0 1 1

example for sector trailer 0 with hex codes

Type of example card:multiapplication with directory

Table 12. example for sector trailer 0 with hex codes

byte 15	byte 14	byte 13	byte 12	byte 10	byte 10	byte 9	byte 8	byte 7	byte 6	byte 5	byte 4	byte 3	byte 2	byte 1	byte 0
s e c r e t						C1	88	77	78	a5	a4	a3	a2	a1	a0
key B						access condition				key A					

All currently unused sectors should be well write protected with secret write keys defined by the card issuer in order to prevent unintended redefinition of access conditions and keys. It is recommended to use different keys for all free sectors. This enables future release of some sectors to new service providers without the need of releasing all free sectors.

3.10 MAD and MIFARE DESFire

For detailed information on the functionality of the MIFARE DESFire IC please refer to the "MIFARE DESFire MF3 IC D40 Short From Specification" available at NXP Document Control.

The MIFARE DESFire card IC features a flexible file system which organizes user data in applications which hold files. Applications are identified with a 3 byte application identifier (AID). AIDs have to be unique per card and are defined at application creation time.

A dedicated list of currently installed application does NOT have to be maintained by the card issuer, as the MIFARE DESFire IC maintains this list automatically. To collect a list of applications on a card, the MIFARE DESFire command GetApplicationIDs is used. This command returns a list holding all MIFARE DESFire AIDs present on the card.

In order to transfer the advantages of the MIFARE classic AID structure to the MIFARE DESFire IC following definitions are made:

- The 3 bytes MIFARE DESFire AID can be used to store the 2 byte MIFARE classic AID
- The first nibble of the MIFARE DESFire AID is fixed to 0xF to indicate: MIFARE classic AID is used.
- The next 4 nibbles hold the MIFARE classic AID.
- The last nibble can be freely chosen to support multiple MIFARE DESFire AIDs within the context of one MIFARE classic AID. This allows to have 16 different MIFARE DESFire AIDs using one single MIFARE classic AID.
- The MIFARE DESFire Card Master Key settings have to allow the MIFARE DESFire command GetApplicationIDs without authentication.
- The MIFARE DESFire AID 0xFF FF FF is reserved.

Table 13. MIFARE DESFire AID

0x	MSB		2nd byte		LSB	
0x	Nibble 0	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5
0x	F	Two-byte MIFARE Classic ID				0..F

The reserved MIFARE DESFire AID 0xFF FF FF is used to store general issuer information:

- File 0x0 has to be a value file with free access for GetValue, holding the value 0x00 00 03, indicating the MAD version 3.
- File 0x1 shall be configured as StandardDataFile with Free Read Access. This file holds the contact details of the Card Holder (user of the card) in CSV plain text, see Section 3.5.
- File 0x2 shall be configured as StandardDataFile with Free Read Access. This file holds the contact details of the Card Publisher (owner of PICC Master Key) in CSV plain text, see Section 3.8.
- Files 0x3 to 0xF are RFU and shall not be used within MIFARE DESFire AID 0xFF FF FF.
- Application Software in Terminals (PCDs) shall ignore files 0x3 to 0xF.

3.10.1 Example

2-byte MIFARE Classic Application ID is mapped to 3-byte DESFire application ID.Let's take MIFARE Classic Application ID according to MAD = 0x4857, where MSB (0x48) is the cluster code for access control and application code is LSB (0x57).

The mapping to DESFire AID is shown in the following table.

Table 14. Example - Mapping of 2-byte MIFARE Classic AID to 3-byte DESFire AID

0x	MSB		2nd byte		LSB	
0x	Nibble 0	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5
0x	F	Two-byte MIFARE Classic ID				0..F
0x	F	0x4857				X
0x	F	4	8	5	7	X
0x	F4857X (X can be any value from 0 to F)					

So, the corresponding 3-byte DESFire Application ID = 0xF4857X; where 0xF4 is the MSB and 0x7X is the LSB.

According to ISO/IEC 7816-4, the bit number 8 to 5 of first byte “F” means “Proprietary category, no registration of application providers”.

Please note according to ISO/IEC 14443 and DESFire, the lowest significant byte is exchanged first e.g. in this case “7X85F4”.

3.11 MAD and MIFARE DESFire EV1

The same approach as explained in Section 3.10 can be implemented for MIFARE DESFire EV1. For detailed information on the functionality of the MIFARE DESFire EV1 IC, please refer to the “MF3ICD81 MIFARE DESFire EV1 Functional Specification”, available via NXP document control.

3.12 MAD and MIFARE Plus

For MIFARE Plus the MAD shall be implemented as described in Section 3.6.

The MIFARE Plus AES keys A for reading the sector 0x00 and the sector 0x10 shall be: 0x a0a1a2a3a4a5a6a7a0a1a2a3a4a5a6a7

For the use of MAD in SL3 the communication must allow plain communication, i.e. in SL3 the byte 5 of the sector trailer must be configured accordingly.

Remark: The default setting for the byte 5 default value, as specified in the MIFARE Plus configuration block, automatically allows plain communication, if not changed during personalization.

Remark: For the 2K MIFARE Plus the AIDs of the sectors 0x21 (33dec) ... 0x28 (40dec) must be set to 00 05 (hexsector not applicable, above memory size).

Refer to the datasheet of MIFARE Plus for more details (BU-ID Doc. no. 1637**).

4. Use of the application directories

4.1 Directory scan procedure for MAD1 and MAD2

The purpose of the MAD is to gain additional information and flexibility. These benefits ask for specific proceedings of application software:

Any transaction should start with a directory scan; that means authentication of sector 0 with key A and reading at least blocks 1 and 2. In most cases block 3 is necessary to get general information about the directory structure found in the GPB of block 3.

The next step is to look for the relevant AIDs in the directory blocks which point to the actual sector addresses in use. Several identical AIDs may point to different sectors belonging to the same application. The data structure within the application sectors must be organized with application software. If sectors are changed during life time of the card application, the software needs specific algorithms for locating single data records in several sectors.

If the GPB (ADV) in block 3, sector 0 identifies the MAD2 (i.e. the use of the sectors 16…39 in the extended memory), the sector 10 hex has to be authenticated with key A. The block 0, 1 and 2 contain the AIDs of the extended directory for the sectors 0x11 … 0x27.

As extension of the MAD2 is organized in the same way as the basic directory in sector 0, the same structure of application software can be used.

4.2 Indirect addressing mode

Data identification and manipulation algorithms should only use the indirect addressing mode by using the sector pointers which are extracted out of the MAD.

4.3 Directory scan procedure for MAD3

To check whether an application is present on a MIFARE DESFire IC, the command “GetApplicationIDs” is used.

Please refer to chapter MAD and MIFARE DESFire respectively “MIFARE DESFire Functional Specification” for more details.

4.4 Registration of MIFARE classic application identifiers

Each MIFARE classic application should be encoded in an unique AID. To achieve this goal a central registration authority is set up. Any organization may request for AIDs for new MIFARE classic application free of charge using the attached registration form (see ANNEX A). The contents of sector B of this form will be inserted in a common database.

4.5 MIFARE standardization group and registration authority

The MIFARE standardization group (MSG) is made up of several major organizations using the MIFARE contactless smart-card in multiple applications.

The MSG has nominated NXP Semiconductors, Austria, to deal with the issues of the registration authority. In addition it serves as contact address for any further requests:

Table 15. Registration

NXP Semiconductors GmbH	Tel.: +43 / 3124 / 299 - 277
Mikron-Weg 1	Fax: +43 / 3124 / 299 - 124
A-8101 Gratkorn, Austria *MIFARE MAD Registration Office*	mailto: nxp.docu-control@nxp.com

A frequently updated list of registered application identifiers can be downloaded from the web page

http://www.nxp.com

5. MAD Sector 0x00 (MAD1 and MAD2)

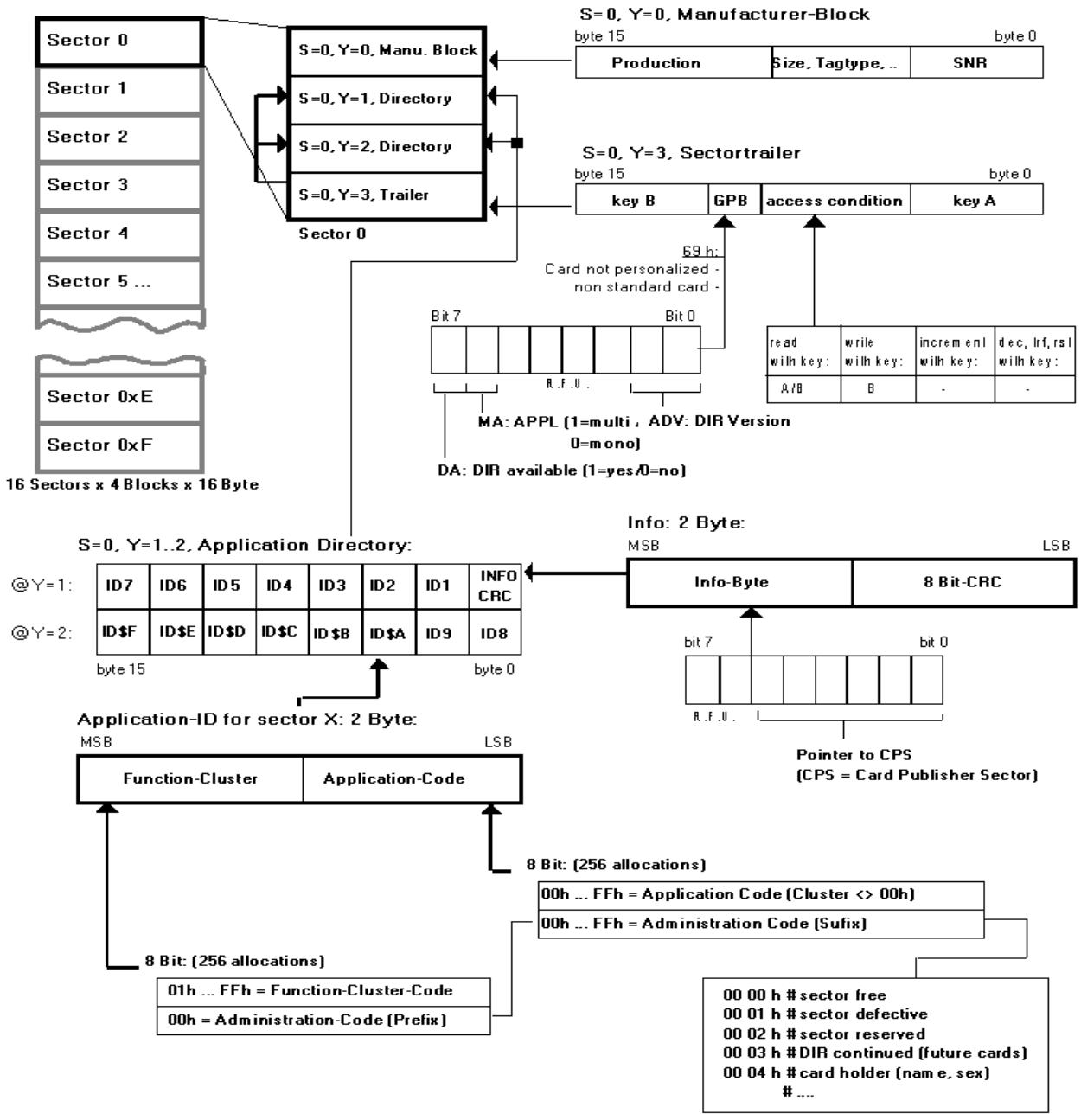

Fig 2. MAD Sector 0x00 (MAD1 and MAD2)

6. MAD Sector 0x10 (MAD2)

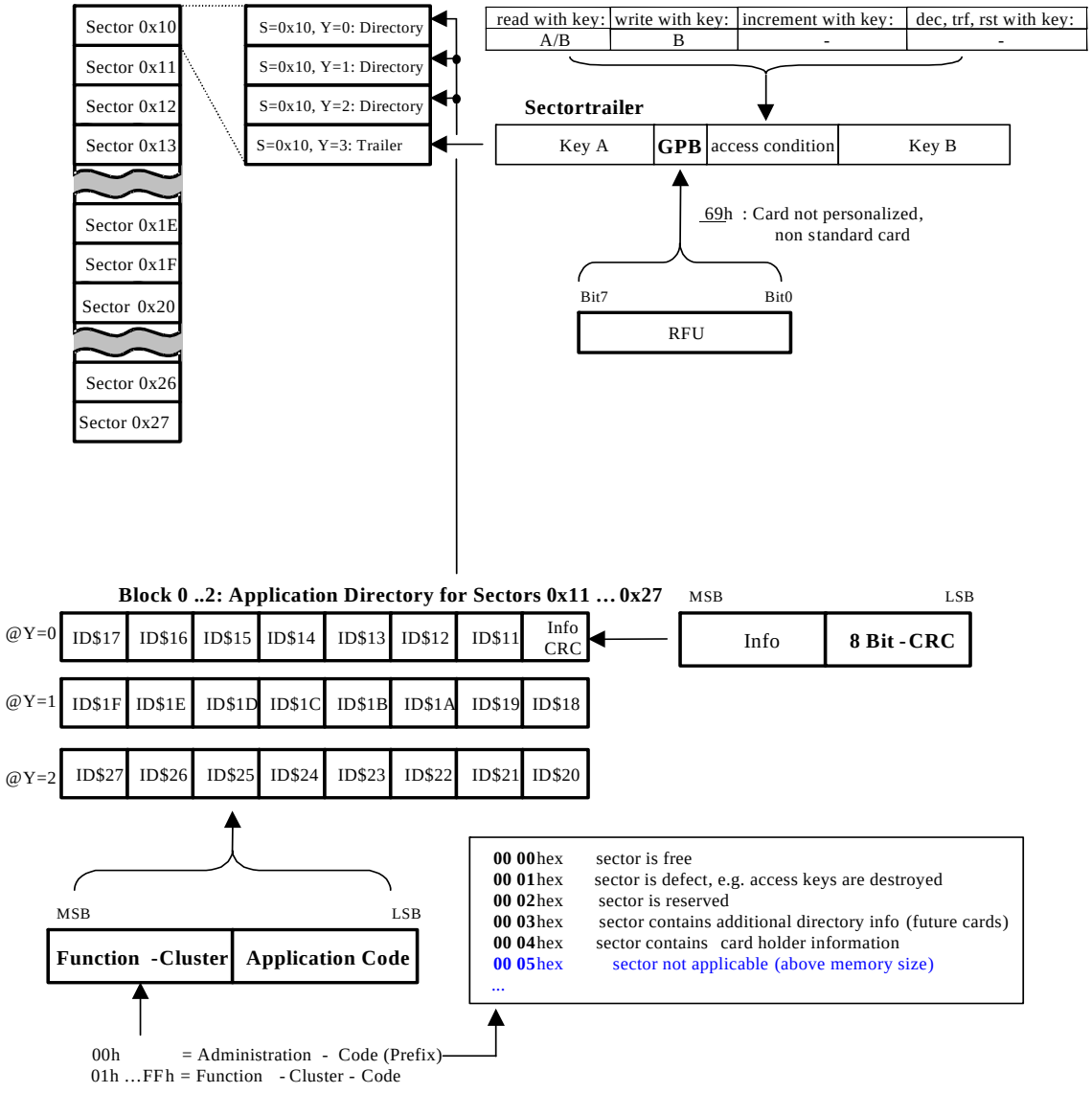

Fig 3. MAD Sector 0x10 (MAD2)

7. ANNEX A, Registration form[1]

REQUEST FOR REGISTERED APPLICATION IDENTIFIER (AID)

information in sector A is not published.

A. To be completed by the requesting organization

100	Name of organization				
101	Address for correspondence				
102	Principal contact in organization				
103	Telephone number	104	Fax number	105	Email address
106	Date	107	Signature		

information in sectors B and C will be published. The requesting organization may omit completition for parts of sector B if this should remain secret.

B. Data to be registered and published

201	Names of service provider organizations				
202	Names of technical system integration organizations				
203	Name of clearing house				
204	Description of application				
205	Suggested functional cluster				
206	Locations of application				
207	Number of sectors in use	208	Launching date	209	Number of desired AID´s
210	Please reserve the following AID´s	211	Please release the following reserved AID´s		

C. To be completed by the registration authority

310	AID granted	311	Functional cluster		
320	AID granted	321	Functional cluster		
330	AID granted	331	Functional cluster		
340	AID granted	341	Functional cluster		
390	Request received by	391	Date	392	Signature

Fig 4. Request for registered application identifier (AID)

1. find help information on next page

8. ANNEX B, Help information for registration form

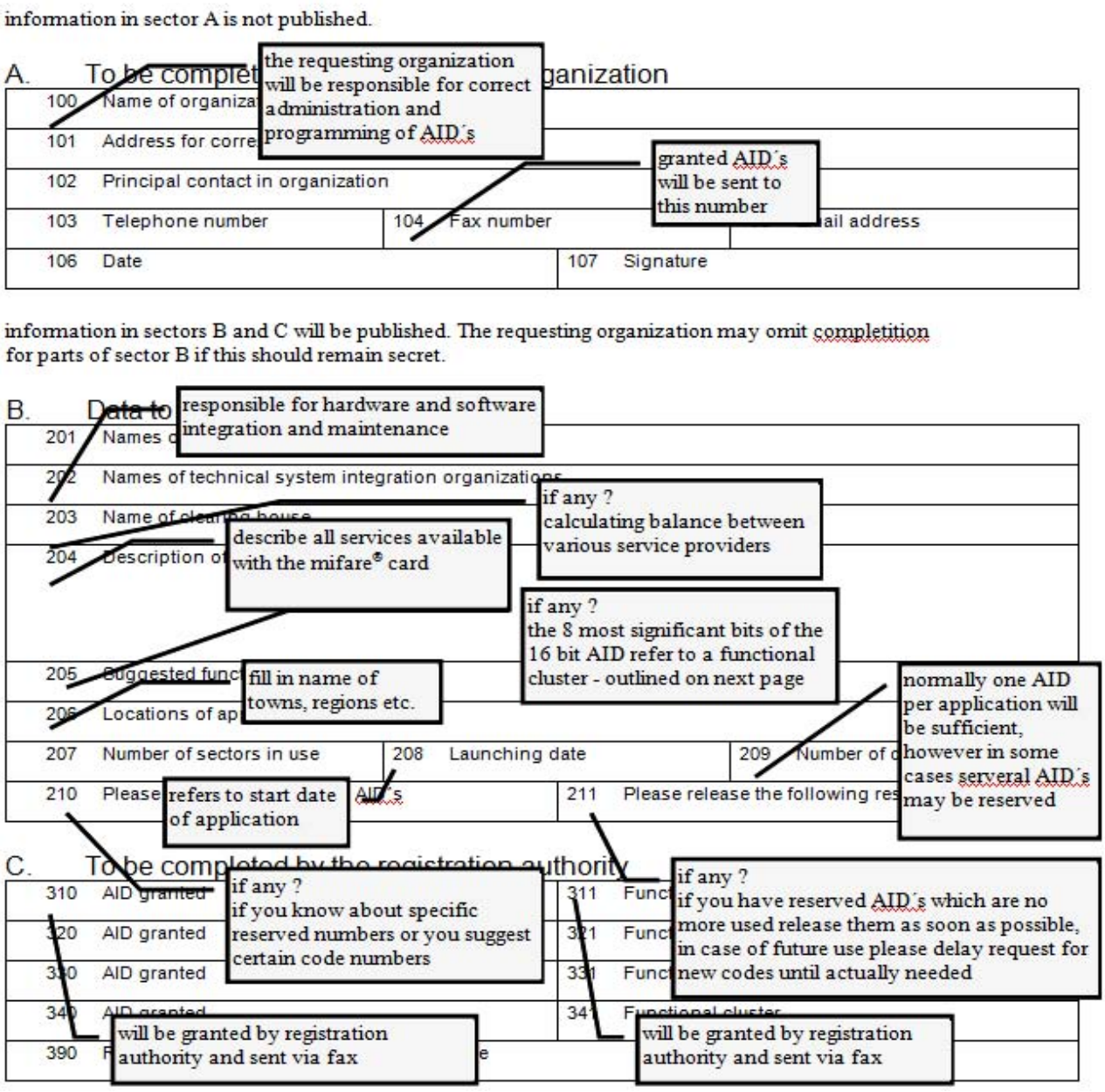

Fig 5. Help information for registration form

9. ANNEX C, Functional cluster codes

All Cluster Code values not listed in Table 16 are rerserved for future use.

Table 16. Function cluster codes

cluster code (hex)	function
00	card administration
01-07	miscellaneous applications
08	airlines
09	ferry trafic
10	railway services
12	transport
18	city traffic
19	Czech Railways
20	bus services
21	multi modal transit
28	taxi
30	road toll
38	company services
40	city card services
47-48	access control & security
49	VIGIK
4A	Ministry of Defence, Netherlands
4B	Bosch Telecom, Germany
4A	Ministry of Defence, Netherlands
4C	European Union Institutions
50	ski ticketing
51-54	access control & security
58	academic services
60	food
68	non food trade
70	hotel
75	airport services
78	car rental
79	Dutch government
80	administration services
88	electronic purse
90	television
91	cruise ship
95	IOPTA
97	Metering
98	telephone
A0	health services

Table 16. Function cluster codes *…continued*

A8	warehouse
B0	electronic trade
B8	banking
C0	entertainment & sports
C8	car parking
C9	Fleet Management
D0	fuel, gasoline
D8	info services
E0	press
E1	NFC Forum
E8	computer
F0	mail
F8-FF	miscellaneous applications

Table 17. (16 bit) AID code

bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
(8 bit)function cluster code								(8 bit)application code							

10. Legal information

10.1 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.

10.2 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.

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 accepts 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 national 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.

10.3 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.

10.4 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.

11. Tables

12. Figures

13. Contents

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

Date of release: 7 July 2010
Document identifier: AN10787