2. MIFARE Classic Key Diversification

MIFARE Classic crypto1 uses 6-byte key for crypto calculation (stream cipher, authentication). This crypto1 keys can be diversified using the UID of the chip. The MIFARE Classic chip may have different types of UID as follows:

- Single size UID, 4-byte long UID
- Double size UID, 7-byte long UID

2.1 MIFARE Classic key diversification for 4-byte UID

2.1.1 MF RC171 like key diversification

MIFARE Classic with 4-byte UID is in the field since 90s. A 2KTDES based key diversification mechanism exists since then for MIFARE Classic keys. This mechanism is supported by MF RC171 as well as all MIFARE SAMs. This method of key diversification is only recommended, if a system needs to be backward compatible with the existing system (MF RC171 like) implementation. For all new systems the method as described in 2.1.2 is recommended.

2.1.1.1 Requirements

1. 6 -Byte MIFARE key.
2. 1-byte MIFARE block address (eventually the unique value for respective sector).
3. 4-Byte MIFARE UID (chip serial number).
4. 16-Byte TDES key (2KTDES).

2.1.1.2 Algorithm:

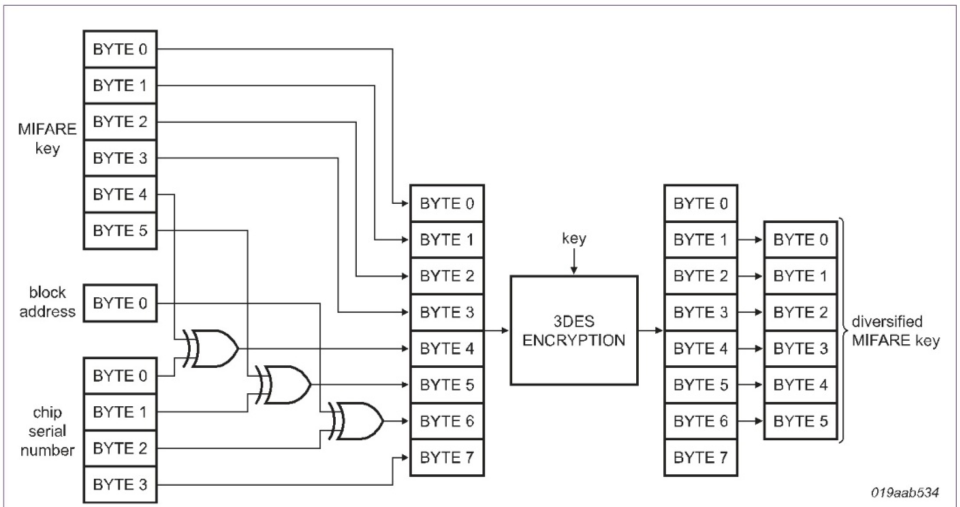

Fig 2. MIFARE Classic key diversification using 4-byte UID

Notes:

1. LSB of the UID is UID Byte 0 and MSB of the UID is UID Byte 3. (i.e the same order as it is received from MIFARE).
2. The block address denotes a parameter must be unique per sector. In example, the MIFARE block number holding the sector trailer block number can be used as shown in the following figure.

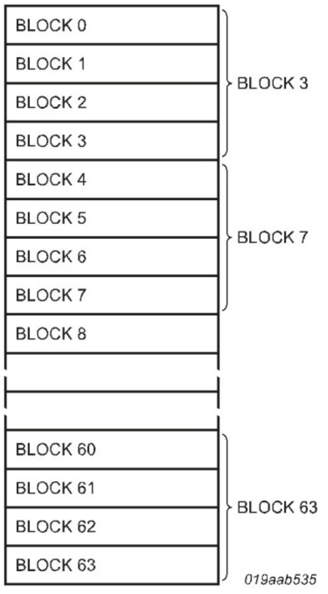

Fig 3. MIFARE Classic 1K sector structure

This block number can be any value ranging from 0x00 to 0xFF which is uniquely assigned to one sector. Anyway, the same value has to be used for key writing and authentication. Using the sector trailer block number is one of many possible options, another one being the sector number or any other defined value.

2.1.1.3 Example

1. MIFARE Key : “A0A1A2A3A4A5”
2. MIFARE block number 0x05.
3. MIFARE UID: “F4EA548E”
4. 3DES Key : “00112233445566778899AABBCCDDEEFF”

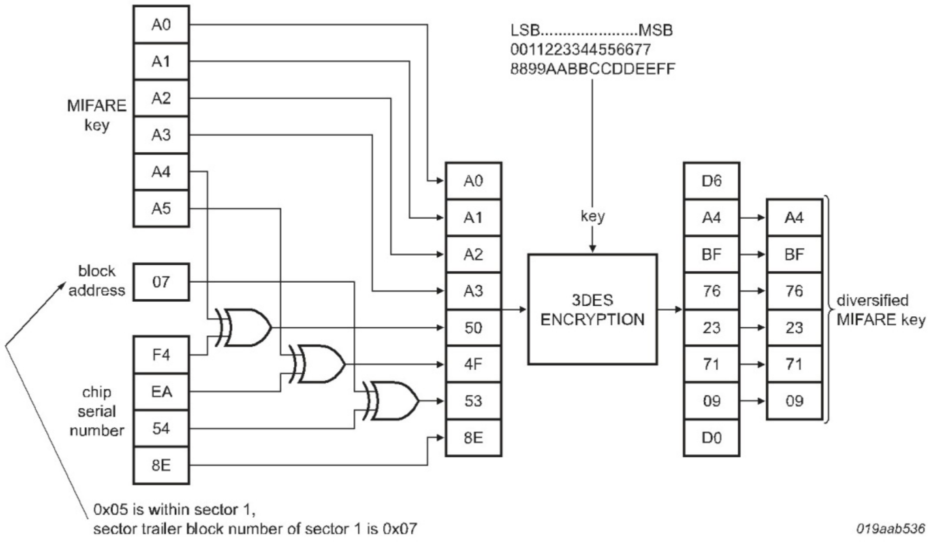

Fig 4. MIFARE classic key diversification example using 4-byte UID

The diversified MIFARE key is “A4BF76237109”

2.1.2 CMAC based Key Diversification

The CMAC based key diversification is the “State of the art” symmetric key diversification mechanism. Adaptive CMAC is calculated based on the Master Key. Using AES master key is recommended.

2.1.2.1 Requirements

1. 16 -Byte AES Master Key (K).
2. 4-Byte MIFARE UID (chip serial number).
3. 1-byte sector address
4. Diversification input made from 4-byte UID and sector address. Some user defined bytes can be added as well (Diversification input up to 31 bytes). Let’s call it M.

2.1.2.2 Algorithm:

1) Calculate CMAC input D:

   D ← 0x01 || M || Padding

   Padding is chosen such that D always has a length of 32 bytes. Padding bytes are according to the CMAC padding, i.e. 80h followed by 00h bytes. So the length of Padding is 0 to 30 bytes.

2) Calculate the Boolean flag ‘Padded’, which is true if M is less than 31 bytes long, false otherwise. The Boolean argument “Padded” is needed because it must be known in AES128CMAC which K1 or K2 is to be used in the last computation round.

3) Calculate output:

   Diversified Key ← AES128CMAC (K, D, Padded)

Processing load:

One AES 128 key load, 3 AES 128 computations

Fig 5 shows the algorithm as a block diagram.

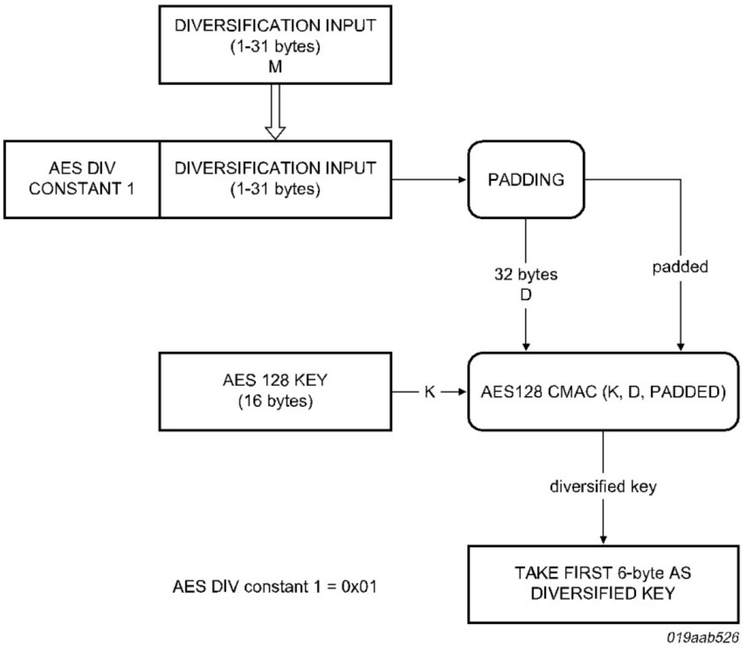

Fig 5. CMAC based MIFARE Classic key diversification

2.1.2.3 Example

Master key (K) = 00112233445566778899AABBCCDDEEFF, which is used to generate 6-byte MIFARE Classic diversified key.

Table 1. Example – CMAC based MIFARE Classic Key diversification using 4-byte UID

step	Indication		Data / Message	Comment
CMAC sub key generation				
1	Master key (K)	=	001122334455667788 99AABBCCDDEEFF	The key, which is going to be diversified
2	K0	=	FDE4FBAE4A09E020 EFF722969F83832B	CIPHK(0b), AES (K, 16-byte 0s).
3	K1	=	FBC9F75C9413C041 DFEE452D3F0706D1	The first sub key, see in [CMAC].
4	K2	=	F793EEB928278083B FDC8A5A7E0E0D25	The second sub key, see in [CMAC].
Diversified key generation				
5	UID	=	F4EA548E	4-byte UID of PICC
6	Sector number	=	05	
7	Diversification input (M)	=	F4EA548E05	Data from step 5 to step 7. It doesn't matter how you make your diversification input, diversification input must be unique for unique PICC e.g. here the UID is unique and the same diversification input must be used in personalization and validation of the PICC. Maximum length of M is 31 bytes.
8	Add the Div Constant 1 at the beginning of M	=	01F4EA548E05	Div constant is fixed, must be 0x01 for AES 128 keys.
9	Do I need Padding	=	Yes	The algorithm always needs 32-byte block for AES; so far we have 18 bytes (step 9).
10	Padding	=	800000000000000000 000000000000000000 0000000000000000	26-byte padding to make 32-byte block.
11	CMAC input D	=	01F4EA548E0580000 000000000000000000 000000000000000000 00000000000	32 bytes.
12	Last 16-byte is XORed with K2	=	01F4EA548E0580000 000000000000000F7 93EEB928278083BF DC8A5A7E0E0D25	As the padding is added the last block is XORed with K2, if padding is not added, then XORed with K1.

step	Indication		Data / Message	Comment
13	Encryption using K	=	2E194999013A58A07 CC1F033643EFDC80 60801E2E71634BCE A2518F9E2C43AC9	Standard AES encryption with IV = 00s in CBC mode
14	CMAC	=	060801E2E71634BCE A2518F9E2C43AC9	Last 16-byte block.
15	6-byte MIFARE Classic Diversified key	=	060801E2E716	First 6-byte of the CMAC

2.2 Key Diversification for 7-byte UID

2.2.1 MF RC171 like key diversification

In this case last 4-byte (cascade level 2) of the MIFARE Classic UID has to be used as the chip serial number. And all the steps are same like the scheme explained in 2.1.1. This method of key diversification is only recommended, if a system needs to be backward compatible with the existing system (MF RC171 like) implementation. For all new systems the method as described in 2.1.2 is recommended.

2.2.1.1 Risk of using only last 4-byte UID in key diversification

The complete 7-byte value of the serial number is unique (UID); means there shall be at least one bit different from other chip serial number. This difference can be in any bit position. Therefore using only last 4-byte may not bring required diversity. In this case using the complete 7-byte in diversification is meaningful and strongly recommended.

2.2.2 CMAC based key diversification

The CMAC based key diversification is the “State of the art” symmetric key diversification mechanism. Adaptive CMAC is calculated based on the Master Key. Using AES master key is recommended. The scheme is explained in 2.1.2.

2.2.2.1 Example

Master key (K) = 00112233445566778899AABBCCDDEEFF, which is used to generate 6-byte MIFARE Classic diversified key.

Table 2. Example – CMAC based MIFARE Classic Key diversification using 7-byte UID

step	Indication		Data / Message	Comment
CMAC sub key generation				
1	Master key (K)	=	001122334455667788 99AABBCCDDEEFF	The key, which is going to be diversified

step	Indication		Data / Message	Comment
2	K0	=	FDE4FBAE4A09E020 EFF722969F83832B	CIPHK(0b), AES (K, 16-byte 0s).
3	K1	=	FBC9F75C9413C041 DFEE452D3F0706D1	The first sub key, see in [CMAC].
4	K2	=	F793EEB928278083B FDC8A5A7E0E0D25	The second sub key, see in [CMAC].
Diversified key generation				
5	UID	=	04793D21801D80	7-byte UID of PICC
6	Sector number	=	05	
7	Diversification input (M)	=	04793D21801D8005	Data from step 5 to step 7. It doesn't matter how you make your diversification input, diversification input must be unique for unique PICC e.g. here the UID is unique and the same diversification input must be used in personalization and validation of the PICC. Maximum length of M is 31 bytes.
8	Add the Div Constant 1 at the beginning of M	=	0104793D21801D800 5	Div constant is fixed, must be 0x01 for AES 128 keys.
9	Do I need Padding	=	Yes	The algorithm always needs 32-byte block for AES; so far we have 18 bytes (step 9).
10	Padding	=	800000000000000000 000000000000000000 0000000000	23-byte padding to make 32-byte block.
11	CMAC input D	=	0104793D21801D800 580000000000000000 000000000000000000 00000000000	32-byte.
12	Last 16-byte is XORed with K2	=	0104793D21801D800 580000000000000F7 93EEB928278083BF DC8A5A7E0E0D25	As the padding is added the last block is XORed with K2, if padding is not added, then XORed with K1.
13	Encryption using K	=	3C81E4C53BA2AE63 5F6725D084E0F1885 508229585D0376654 BC266B5F5997DB	Standard AES encryption with IV = 00s in CBC mode
14	CMAC	=	5508229585D037665 4BC266B5F5997DB	Last 16-byte block.
15	6-byte MIFARE Classic Diversified key	=	5508229585D0	First 6-byte of the CMAC