AN13452

MIFARE Ultralight AES features and hints

Rev. 1.1 — 22 February 2022

Application note
COMPANY PUBLIC

Document information

Information	Content
Keywords	Multiple ticketing, secured data storage, implementation hints, AES authentication, memory layout, configuration
Abstract	This document presents features and hints for a secured and optimized application development using MIFARE Ultralight AES cards.

Revision history

Revision history

Rev	Date	Description
1.1.	20220222	Security status changed into "Company public"
1.0	20220208	Initial release

1 Introduction

1.1 Purpose and scope

This application note is intended to describe the features and functionality of the MIFARE Ultralight AES from an application point of view. The document gives examples of how to use MIFARE Ultralight AES and some best practices and recommendations. For more information on how to handle the MIFARE Ultralight AES security features, please refer to [UM11764]

1.2 Disclaimer

MF0AES(H)30 comes with an external CC EAL3+ certification targeting basic attack potential (AVA_VAN.2). Hence, the contactless IC does not claim to be completely resistant. In case of broader protection is required, products with a higher security certification should be considered.

Note therefore that whenever terms like locking, read-only, fraud protection, security feature and the like are used, this does not imply that there would never be such an attack possible to circumvent such a feature.

1.3 How to use this document

This document contains a collection of hints and features that could be of interest for users, who plan to use the MIFARE Ultralight AES.

None of this information is intended to replace any of the relevant data sheets or user guidelines.

All the numerical examples are just examples, describing the usage of commands and providing reference values to verify any implementation.

2 MIFARE Ultralight AES application hints

2.1 Memory features

In addition to the user memory area the MIFARE Ultralight AES offers the features of an OTP[1] area and lock bytes to lock the OTP and user area. The usage of LOCK bits is described in [DS5379].

The configuration pages are located after the user memory area. In configuration pages, several chip configurations can be set. After configuration pages are located 2 x 4 pages for two AES keys.

2.1.1 Memory organization

The EEPROM memory is organized in pages with 4 bytes per page. The memory organization can be seen in Table 1 below, the functionality of the different memory sections is described in the following sections.

Page 03h is the OTP page and the default value of the OTP bytes is 00 00 00 00h. These bytes can be bit-wise modified using the WRITE command. It is not possible to clear a bit that was set in this area.

Page 02h contains the lock bytes 0 and 1 which represent the field programmable read-only locking mechanism.

Lock bytes 2,3 and 4, configuration and key page addresses depend on the memory size, as they are located after the user memory. For MIFARE Ultralight AES with 144 bytes of user memory, the lock bytes 2, 3 and 4 are located at page 28h.

Table 1. Memory organization for 144-byte user memory variant

Page address		Description	Byte number			
Decimal	Hex		0	1	2	3
0	00h	Manufacturer Data and lock bytes 0 and 1	serial number			
1	01h		serial number			
2	02h		serial number	internal	lock byte 0	lock byte 1
3	03h	32-bit user programmable OTP area	OTP	OTP	OTP	OTP
4	04h		user memory			
5	05h					
...	...					
38	26h					
39	27h					
40	28h	Lock bytes 2, 3 and 4	lock bytes	lock bytes	lock bytes	RFU
41	29h	Configuration pages	CFG_0			
42	2Ah		CFG_1			
...	...		RFU			

1 **O**ne **T**ime **P**rogramming

Table 1. Memory organization for 144-byte user memory variant*...continued*

Page address		Description	Byte number			
Decimal	Hex		0	1	2	3
45	2Dh		LOCK_KEYS			
...	...		RFU			
48 to 51	30h to 33h	Data Protection key	AES authentication key [DataProtKey]			
52 to 55	34h to 37h	UID retrieval key	AES authentication key [UIDRetrKey]			

Note: RFU bytes shall be kept in their default state and shall not be used to store user data.

2.1.2 Lock bytes

Each page from 03h (OTP) to 0Fh can be individually locked by setting the corresponding locking bit Lx to logic 1 to prevent further write access. After locking, the corresponding page becomes read-only memory. Additionally, the block-lock bits in page 2 byte 2 (lock byte 0) lock the actual configuration of the lock bits.

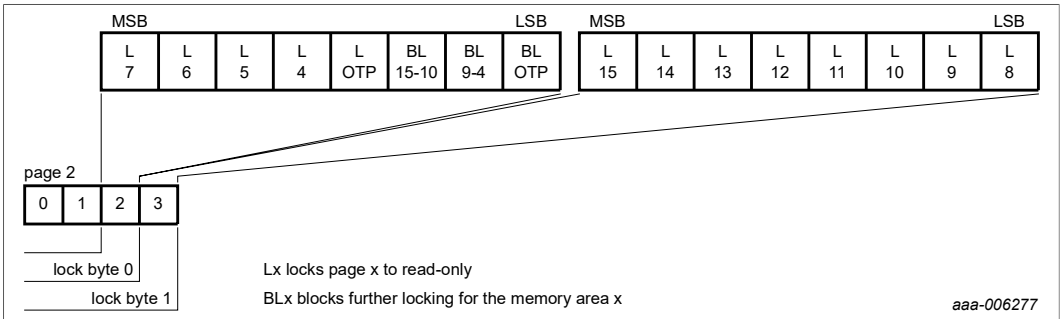

Figure 1. Lock bytes page 2

The lock bytes 2,3 and 4 for the rest of the memory, are located in the first page after the user memory. The granularity there is depending on the total user memory size. The block-locking bits for the lock bytes 2 and 3 are located in in lock byte 4.

Note: At personalization, once configuration of the memory area is frozen, it is recommended to set all block-locking bits.

In case the use case does not allow for all block-locking bits to be set, ensure that for Lock byte pages where the block-locking bits are not fully set, authentication is required and preferably CMAC-based secure messaging is enabled. Note that if block-locking bits on page 2 cannot be set, this means that the whole user memory needs to be protected!

2.2 Key handling

MIFARE Ultralight AES offers two independent AES-128 keys, located in the memory pages 30h-37h.

1. AES Key0 [DataProtKey]: This key is used for accessing the protected user memory and counter 2. Also, it can be used to retrieve the real UID and ECDSA signature of the MIFARE Ultralight AES in case of Random ID is configured and this key is not UID diversified.
2. AES Key1 [UIDRetrKey]: This key can only be used to retrieve the UID and ECDSA (Elliptic Curve Digital Signature Algorithm) signature in case of Random ID is

enabled. This key should be non-diversified, as the UID used for eventual key diversification cannot be retrieved without it.

NXP recommends using a diversified AES-128 key as AES Key0 (see [AN10922]) and use a non-diversified AES-128 key as AES Key1 for UID and signature retrieval.

The keys are programmed into the memory by using a normal WRITE command, starting at the first page of a given key (e.g. page 30h) with byte 0 as the key LSB up to the last page of the given key (e.g. 33h) byte 3 as the key MSB.

e.g: Key = 00(MSB) 112233445566778899AABBCCDDEE FF(LSB):

Table 2. AES key in memory

Page address	Byte Number			
Hex	Byte 0	Byte 1	Byte 2	Byte 3
30h	FF(LSB)	EE	DD	CC
31h	BB	AA	99	88
32h	77	66	55	44
33h	33	22	11	00(MSB)

Both keys can be used for authentication. If Key0 is used, the MIFARE Ultralight AES moves into AUTHENTICATED state, if Key1 is used, the MIFARE Ultralight AES moves into TRACEABLE state. (For details, see [DS5379])

2.3 Retrieval of the UID and originality signature in case of Random ID configuration

MIFARE Ultralight AES offers the possibility to use a **Random ID (RID)** during card activation. If this option is enabled, the MIFARE Ultralight AES uses a random 4 byte UID during card activation, to ensure privacy of the card owner in ACTIVE(*) state.

Note: Consequently, it needs to be made sure that no user individual data shall be placed in the freely available user memory when using Random ID.

If the MIFARE Ultralight AES is configured for Random ID, the real UID can be retrieved after authentication with either the AES Key0 [DataProtKey] or the AES Key1 [UIDRetrKey]. After authenticating, the UID can be read using a READ command from page 0,1 and 2, as well as the NXP Originality Signature can be read using the READ_SIG command. If no authentication was performed, this pages and the signature are masked with 00h once reading them.

2.4 Configuration options

As soon as the user configuration is finalized during personalization, it is also advised to lock the configuration pages by setting the bit **LOCK_USR_CFG**. This ensures that the configuration cannot be changed again. **Note:** This bit can, as all other lock bits, only be set once!

In order to be able to use the CMAC-based Secure Messaging, this needs to be activated by setting the **SEC_MSG_ACT** bit. After setting this bit, the MIFARE Ultralight AES will only accept commands that are sent by using the Secure Messaging (i.e. by appending a CMAC). Details on how this Secure Messaging exactly works can be found in Section 3.

MIFARE Ultralight AES supports the virtual card architecture concept, by replying to the *VirtualCardSelectLast* command (VCSL) with a configurable response. This response (1 byte) can be set using the VCTID byte. By default, this is set to 05h.

MIFARE Ultralight AES provides several configuration bytes after the lock bytes to configure specific behavior of the IC. All configuration options are described in detail in the data sheet [DS5379].

The configuration options **LOCK_AES_KEY0** and **LOCK_AES_KEY1** can be used to lock the AES Keys 0 and 1. It is advised that after personalizing the keys, these bits are set to make the keys unchangeable if the use case allows. Also, it is advised that once the key config is frozen, the **BLOCK_LOCK_KEY** bit is set, to lock the key locking bits (this can also mean, that one or both LOCK_AES_KEYx bits might stay at 0).

3 MIFARE Ultralight AES memory access protection

MIFARE Ultralight AES provides the possibility to use an AES-128 key (Key0 [DataProtKey]) for memory and/or counter 2 protection, as well as secure messaging for data integrity protection during communication. By default, the memory protection feature is disabled, and the corresponding AES keys can be freely written. To personalize the AES keys, a secure environment is advised, as the MIFARE Ultralight AES does not support any way to inject keys in an encrypted form.

3.1 AUTH0 and PROT configuration option

To allow access to a part or all of the user memory of MIFARE Ultralight AES via a successful authentication using the data protection key, the AUTH0 and PROT configuration options are used.

AUTH0: defines the page address from which onwards the specific memory access will be accessible only after authentication. The value can be between 00h and the last page available in memory. Values above the last available memory page address have no effect on the memory access.

The PROT option defines, if only the write access is restricted by AES Key0 [DataProtKey], but read is possible without authentication (PROT=0), or both are restricted (PROT=1).

Note: Since AUTH0 has 7 bits it can be effectively set until 7Fh, which is above available user memory, meaning the whole user memory is freely accessible.

Example:

- AUTH0 = 12h
- PROT = 1

In this example, all pages below 12h are accessible freely, remaining pages starting at 12h (included) are only read/writeable after successful authentication with AES Key0 [DataProtKey].

Table 3. Memory protection example: orange marked area is only accessible after authentication with Key0[DataProtKey]

Page Address		Byte number			
dec	hex	1	2	3	4
1	1	serial number			
2	2	serial number			
3	3	serial number	internal	lock byte	lock byte
4	4	OTP	OTP	OTP	OTP
5	5	User memory			
...	...				
17	11				
18	12	user memory			
...	...				
39	27				
40	28	lock byte	lock byte	lock byte	RFU
41	29	CFG_0			
42	2A	CFG_1			

Table 3. Memory protection example: orange marked area is only accessible after authentication with Key0[DataProtKey]*...continued*

Page Address		Byte number			
dec	hex	1	2	3	4
43	2B	RFU			
44	2C				
45	2D	CMAC_CFG	RFU		
46	2E	RFU			
47	2F				
48	30	AES Key 0 [DataProtKey]			
49	31				
50	32				
51	33				
52	34	AES Key 1 [UIDRetrKey]			
53	35				
54	36				
55	37				
56	38	RFU			
57	39				
58	3A				
59	3B				

3.2 AES counter 2 protection

MIFARE Ultralight AES offers in total 3x 24-bit one-way counters, of which one (counter no. 2) can be protected by AES authentication.

By default, the protection is switched off.

To enable the counter protection, the respective bit in the configuration pages need to be set:

- CNT_INC_EN: If set to **0**, the counter can be incremented only after successful AES authentication with the memory protection key.
- CNT_RD_EN: If set to **0**, the counter can be read only after successful AES authentication with the memory protection key.

3.3 AUTH_LIM configuration option

As card-only side-channel attacks are a common risk for MIFARE Ultralight AES, a failed authentications limit is present. It can be activated by setting corresponding configuration element to a value between 001h and 3FFh. The failed authentication counter is by default disabled (set to value 000h).

Note: It is recommended setting this limit to a value of 100 (064h) or below.

The negative authentication counter counts each unsuccessful attempt of authentication. Each successful authentication reduces the counter by 10h. Once the negative authentication counter reaches its limit, the access to protected memory cannot be further authenticated even if the used key is correct. Any further authentication attempt results in a failure.

3.4 Authentication example

The following example shows an authentication with AES Key0 [DataProtKey].

The key used in this example is an all zeros AES-128 key (default).

Table 4. Authentication example (CRC not shown)

Step	Command	Direction	Message	Comment
1	Write Page 29	>	A22902000012	set SEC_MSG_EN, set AUTH0 to 0x12. Both configuration options are located in page 29h for the 144 byte memory variant.
2	Response	<	0A	
3	Write Page 2A	>	A22A80050000	set PROT bit (b7) in byte 1 of page 2Ah (for the 144 byte memory variant), other configurations left at default
4	Response	<	0A	
5				Reset and reactivate card
6	Authenticate part1	>	1A00	Authenticate with card key 0
7	Response	<	AF374142DAFB0AB97183D846EB7ED379E0	Status code (AFh) + 16 byte encrypted RndB
8	Decrypt RndB	=	0D2BBA17011098E9864C8AA5192AF796	IV[1] = 00s
9	Generate RndA	=	42BDF7E08E110F14B6D3323D14F1C2B9	
10	Generate RndB'	=	2BBA17011098E9864C8AA5192AF7960D	rotate RndB by 1 byte
11	Encrypt (RndA || RndB')	=	794693B2A31E7F10964A2BD834590AC485F84E9F8B13197AB32433346F60B821	
12	Authenticate part2	>	AF794693B2A31E7F10964A2BD834590AC485F84E9F8B13197AB32433346F60B821	AFh + encrypted RndA || RndB'
13	Response	<	00A17673DCC20D27EE80584ACCFC39E0A3	success code (00h) + encrypted RndA'
14	Decrypt RndA'	=	BDF7E08E110F14B6D3323D14F1C2B942	IV = 00s
15	Verify			RndA' = rotate(RndA)
16	Generate session vector	=	5AA50001008042BDFACB34060E0498E9864C8AA5192AF796B6D3323D14F1C2B9	SV = 5Ah||A5h||00h||01h||00h||80h||RndA[15..14]||RndA[13..8] XOR RndB[15..10])||RndB[9..0]||RndA[7..0]
17	Calculate session key	=	D6B4F8AC7A66CFA041DC179A154543BF	CMAC of SV using the secret key

[1] Initialization Vector

Note: Steps 16 and 17 are only needed if secure messaging is used, otherwise it can be skipped.

After the authentication with AESKey 0 [DataProtKey], the MIFARE Ultralight AES is in state AUTHENTICATED.

At this point, the memory starting from page 12h can be read and written only after successful authentication with this key (as shown in Table 3).

4 CMAC-based secure messaging for data integrity

Optionally, the MIFARE Ultralight AES offers a CMAC (Cipher-based Message Authentication Code) based secure messaging to add data integrity protection during communication. The CMAC calculation is done according to NIST Special Publication 800-38B [NIST SP800-38B].

To enable the secure messaging, the SEC_MSG_EN configuration bit must be set (see Table 4 step 1).

The key used for the CMAC calculation is a session key based on the random numbers generated during the last authentication either with the memory protection key or the UID retriever key. An example can be found in Table 4 step 14 and 15.

The CMAC appended to the command is calculated over

- 2 byte command counter
- 1 byte command code
- command arguments

and for the responses, CMAC is calculated over

- 2 byte command counter
- command response

Note: For both command and response CMAC calculation, CRC bytes are excluded, but still added as last bytes when sending the command.

The command counter is a 2-byte counter reset to 0000h upon reception of valid AuthenticatePart1 command, and is after AuthenticatePart2 subsequently incremented after reception of a command and again after sending the response.

Note: In all cryptographic operations, the command counter is represented as LSB first.

For commands where the expected response is only an ACK, this ACK is replaced by just the CMAC calculated over the command counter. Note that also in this case, a CRC is appended after the CMAC.

In case the CMAC-based secure messaging cannot be applied, a system level CMAC (calculated outside the MIFARE Ultralight AES and stored in the user memory) can still be considered as described in [AN11340] (section 2.2.1).

Important Note: MIFARE Ultralight AES offers no confidentiality protection on the transmitted or stored data. In case this is desired, consider using a higher security product like MIFARE DESFire (Light) or MIFARE Plus.

4.1 Secure Messaging example

This example builds on the authentication that was performed in Table 4. The following example will perform a GetVersion, Read and Write command into the user memory of MIFARE Ultralight AES page 04h, using the secure messaging.

The session key is **D6B4F8AC7A66CFA041DC179A154543BF** and the command counter is 0000h

Step	Command	Direction	Message	Comment
GetVersion				
1	Generate CMAC input data	=	000060	CMAC Input Data = CmdCtr || Cmd
2	Calculate CMAC	=	551D3B5D8FE8A902A9D6A3752E164C70	CMAC(K_{MAC}, MAC input data)
3	Truncate CMAC ($CMAC_T$)	=	1D5DE802D6751670	using "MFP truncation"[1], every even byte is used.
4	GetVersion	>	601D5DE802D6751670	Cmd (60h)|| CMAC from step 3
5	Response	<	0004030104000F03235C940315BE9A12	Version Response || CMAC
6	CMAC Input	=	01000004030104000F03	CmdCtr (0001h) LSB first || response
7	verify $CMAC_T$	=	235C940315BE9A12	calculated $CMAC_T$ equals received $CMAC_T$
Read Page 04h				
8	Generate $CMAC_T$ for read command	=	FD9FC13ECFD0FDF2	CMAC Input Data = CmdCtr || Cmd || CmdArgs: 0200 30 04
9	Read page 04h	>	3004FD9FC13ECFD0FDF2	Cmd || CmdArgs (04h) || $CMAC_T$
10	Response	<	AABBCCDD00000000000000000000000076172B2C9F4B123C	16 byte data || $CMAC_T$
11	calculate $CMAC_T$ and verify	=	76172B2C9F4B123C	CMAC Input: CmdCtr (0003h) LSB first || response = 0300AABBCCDD00000000000000000000000
Write page 04h				
12	Generate $CMAC_T$ for write command	=	876BA2ACF99B33A8	CMAC Input Data = CmdCtr || Cmd || CmdArgs: 0400 A2 04 AABBCCDD
13	Write page 04h	>	A204AABBCCDD876BA2ACF99B33A8	Cmd || CmdArgs || $CMAC_T$
14	Response	<	EA81F87A65A80B91	$CMAC_T$ only, no status code present
15	calculate $CMAC_T$ and verify	=	EA81F87A65A80B91	CMAC Input: CmdCtr (0005h) = 0500

[1] Special truncation method introduced first at MIFARE Plus. Instead of using the 8 MSBytes, every even byte is used.

5 MIFARE Ultralight AES counters

The MIFARE Ultralight AES features **three** independent 24-bit one-way counters, of which **one** can be protected by AES-128 authentication (counter no. 2). **It is strongly recommended to use the counter no. 2 with AES protection** as much as possible. This is the most secure way to implement a one-way counter on MIFARE Ultralight AES.

The counters are initialized to 000000h at delivery. Each counter can be read using the READ_CNT command and increased using the INCR_CNT command. The INCR_CNT command works with values from 000000h to FFFFFFh. Note that the value of zero can be used as well, although it will not increment the counter in practice.

An example is indicated in Figure 2.

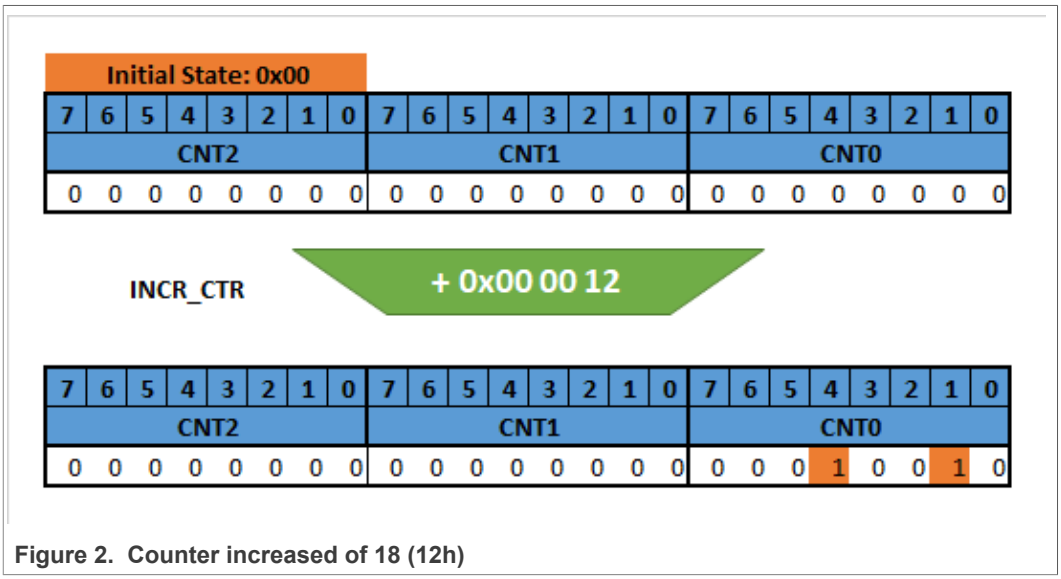

Figure 2. Counter increased of 18 (12h)

MIFARE Ultralight AES counters come with anti-tearing support to avoid unintended values caused by a tear-off from the reader during a transaction. Following steps are recommended on the counter for e.g. ticketing purposes:

1. Read the counter using READ_CNT in order to check the current counter value (**preferably using CMAC-based secure messaging**)
2. Increase the counter using INCR_CNT:
   a. If the INCR_CNT response is equal to NAK5/7, this might be a tearing event, repeat steps 1) and 2)
   b. If the INCR_CNT response is equal to NAK6, the counter is corrupted or unusable, invalidate the ticket. This is not expected to occur during normal usage.
   c. In case of timeout or reset, repeat steps 1) and 2)
   d. Otherwise, execute step 3)
3. Read again the counter using READ_CNT to check that the new expected counter value has been correctly stored. **This step is not needed when using CMAC-based secure messaging and a valid CMAC is received on the INCR command!**
   a. If the value is not correctly stored, repeat steps 1) to 3) up to N times
   b. If after N times the value is still not correct, invalidate the ticket

How a ticket is invalidated depends on the underlying system, e.g. it could be done by writing to the OTP area, setting counters to the maximum value or setting some flag within the user memory.

5.1 Using OTP memory for multiple ticketing

The MIFARE Ultralight AES offers the possibility to also use the OTP bytes in page 03h as counter. All bits of the OTP bytes are pre-set to “0” at the delivery. These bits can be set one time to “1”. This gives the possibility to interpret them as a counter e.g. in a public transport use case to count the number of trips. Therefore, the number of “1” in OTP area of page 03h can be considered as counter value, beside the 3x independent 24-bit counters on MIFARE Ultralight AES (see Section 5). Meaning, the OTP bytes of page 03h offer a number of 32 states that could be used to allow a certain number of passings through a turnstile.

Same as counters, also OTP bits are tear-protected. After a tearing event, OTP bit shall contain either the intended value or the previous value before the last intended change.

Note: The OTP area can be protected by AES authentication and secure messaging as well, to prevent unwanted altering of the data stored in there.

5.1.1 Recommended implementation of One-Time Programmable bits as counter

There are different ways the One-Time Programmable bits can be used as counter, but there is one recommended way to implement the counter to reduce the occurrence of unintended OTP values. The defined start value of the counter in the OTP area, page 03h, is recommended to be the highest possible value with the remaining trips allowed in the system. E.g. set the page 03h of the OTP area to 0x0F FF FF FF. Now, the OTP area is pre-set for 4 trips by writing the bits in the MSB sequentially from 0x0F FF FF FF to 0xFF FF FF FF (max).

5.2 24-bit one-way counter implementation considerations

To decrease the possibility for data manipulation in applications and to allow for the detection of manipulated counter values on the three independent anti-tearing supported 24-bit one-way counters following implementation hints and considerations shall be made:

Note: Counter 02h can be protected by AES-128 authentication and CMAC-based secure messaging. There, no additional measures are needed.

- Implementation of a data integrity protection on the counter values that cannot be protected with the AES counter protection and CMAC-based secure messaging: A MAC which has been calculated outside the ticket is stored in the user memory on the ticket to give the possibility to detect if there is a malicious change on the counter value.
- It is recommended to implement countermeasures outside the ticket to support a backend fraud detection mechanism in the infrastructure. It shall give the possibility to detect, based on the UID of MIFARE Ultralight AES, if the counter value linked to the ticket (UID) is as expected. If not, the card with this UID shall be rejected.
- To limit the counter usage and reduce the occurrence of unexpected counter values, define the start value of the three independent 24-bit one-way counters at the highest possible counter value during personalization, by considering the required counts in application. That can be applied, by setting the counter value to the maximum 0xFF FF FF minus the number of counts allowed in the system, e.g. set the counter value to

0xFF FF F5. The counter value of 0xFF FF F5 allows 10 trips by increase of the value to 0xFF FF FF (max). (see Figure 3)

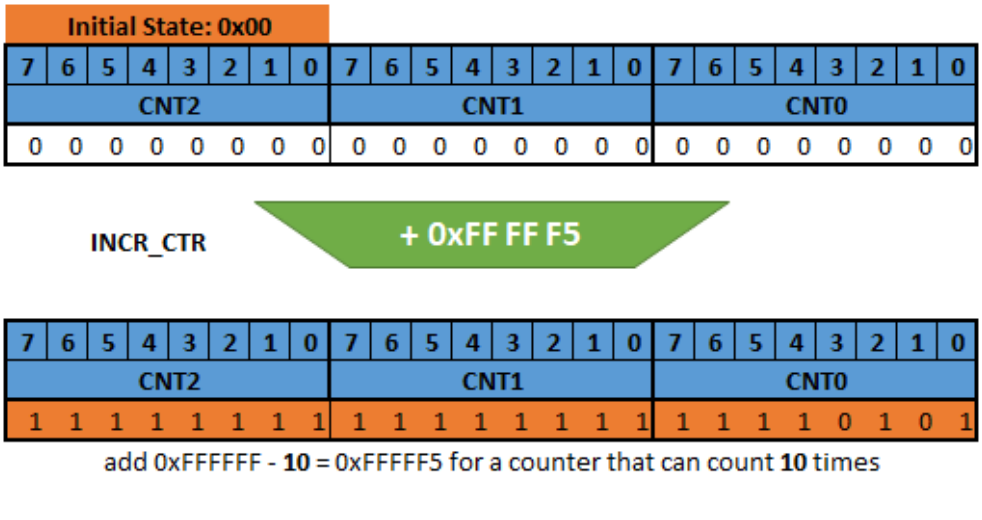

Figure 3. Example of a max. 10 times counter

See also [AN12653]

6 Originality Check

6.1 MIFARE Ultralight AES anti-cloning based on ECC signature

The MIFARE Ultralight AES supports the NXP originality check based on a 48 byte ECC signature (see [DS5379]).

The purpose of originality check during (pre-)personalization is to protect customer investments by identifying mass penetration of non-NXP originated MIFARE Ultralight AES ICs into an infrastructure. As individual signatures can still be copied, this originality check does not completely prevent hardware copy or emulation of individual MIFARE Ultralight AES ICs. Therefore, this signature validation should be complemented with a check to detect if multiple ICs with the same UID are being introduced in the system.

6.1.1 Changing of NXP originality signature

Upon delivery, the MIFARE Ultralight AES holds an ECDSA (Elliptic Curve Digital Signature Algorithm) signature calculated over the UID with the NXP private ECC originality key, and shall serve the purpose of identifying possible copies. MIFARE Ultralight AES offers the possibility to change this signature to some other specific value, using the WRITE_SIG command. This command is very similar to a normal WRITE command, but not targeting the user memory, but the signature memory. It consists of 12 4-byte pages 00h to 0Bh. Byte 0 in page 00h holds the LSB, byte 3 in page 0Bh holds the MSB.

This feature can be used to replace the NXP signature during personalization with a system specific-signature calculated with a system-specific key pair, to identify MIFARE Ultralight IC's that do not belong to this system in way.

The content of this memory area will be returned when the READ_SIG command is sent.

The signature memory can be in 3 states: unlocked, locked or permanently locked. These states can be reached by using the LOCK_SIG command.

- In unlocked state, it is possible to write the signature using the WRITE_SIG command. The signature can be locked (LOCK_SIG with parameter 01h) and permanently locked (LOCK_SIG with parameter 02h).
- In locked state, the signature cannot be written, but unlocked using the LOCK_SIG command with parameter 00h
- In permanently locked state, the signature cannot be written, neither can it be unlocked anymore.

It is recommended to permanently lock the signature once no change needs to be made anymore.

In case CMAC secure messaging is activated, CMAC on WRITE_SIG and LOCK_SIG is required.

Table 5. Read / change signature example

Step	Command	Direction	Message	Comment
Read Signature				
1	READ_SIG	>	3C00	
2	ECDSA signature	<	1824472A4CC927C7CA423F2B75E8E15CD26F682D3D633B3E032879B11D2E7C0E5BDC720D7D4F3AB04DEC7229EC213C89	48-byte ECDSA signature over card UID calculated with the NXP key
Unlock Signature				
3	LOCK_SIG	>	AC00	Unlocks the signature and makes it writeable
4	ACK	<	0A	
Write new Signature				
5	WRITE_SIG	>	A900AAAAAAAA	Write the new signature page by page using the WRITE_SIG command (A9h) followed by the page number (00h-0Bh) and the 4 byte data to write. If SEC_MSG_EN = 1, each command would have a CMAC appended as well
6	ACK	<	0A	
7	WRITE_SIG	>	A901BBBBBBBB	
8	ACK	<	0A	
9	WRITE_SIG	>	A902CCCCCCCC	
10	ACK	<	0A	
11	WRITE_SIG	>	A903DDDDDDDD	
12	ACK	<	0A	
13	WRITE_SIG	>	A904EEEEEEEE	
14	ACK	<	0A	
15	WRITE_SIG	>	A905FFFFFFFF	
16	ACK	<	0A	
17	WRITE_SIG	>	A90600000000	
18	ACK	<	0A	
19	WRITE_SIG	>	A90711111111	
20	ACK	<	0A	
21	WRITE_SIG	>	A90822222222	
22	ACK	<	0A	
23	WRITE_SIG	>	A90933333333	
24	ACK	<	0A	
25	WRITE_SIG	>	A90A44444444	
26	ACK	<	0A	
27	WRITE_SIG	>	A90B55555555	
28	ACK	<	0A	
After checking, permanently lock the signature				
29	LOCK_SIG	>	AC02	Permanently lock signature after changing (Parameter 02h)
30	ACK	<	0A	
Read new Signature				

Table 5. Read / change signature example*...continued*

Step	Command	Direction	Message	Comment
31	READ_SIG	>	3C00	
32	Custom Signature	<	AAAAAAAABBBBBBBBCCCCCCCC CDDDDDDDDEEEEEEEEFFFFFFFF F000000001111111122222223333 33334444444455555555	New Signature is transmitted (LSB first)

6.1.2 Signature verification

The ECC signature can be retrieved from the MIFARE Ultralight AES IC using the READ_SIG command.

Table 6. Read Signature from MIFARE Ultralight AES

Step	Command	Direction	Message	Comment
1	READ_SIG	>	3C00	READ_SIG command, Addr is fixed to 00h
2	ECC originality Signature	<	1824472A4CC927C7CA423F2B75E 8E15CD26F682D3D633B3E032879 B11D2E7C0E5BDC720D7D4F3AB0 4DEC7229EC213C89	48 byte ECC signature

The NXP ECC signature is calculated on the UID of the MIFARE Ultralight AES, using the **ECDSA** algorithm on the **secp192r1** curve.

Note: For specific treatment of READ_SIG answer in case of Random UID, see Section 2.3

The curve parameters for secp192r1 are:

Table 7. Curve parameters

Parameter	Value
Seed	3045AE6FC8422F64ED579528D38120EAE12 196D5h
p	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFF FFFFFFFFFFFFFFh
a	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFF FFFFFFFFFFFFFCh
b	64210519E59C80E70FA7E9AB72243049FEB8 DEECC146B9B1h
x	188DA80EB03090F67CBF20EB43A18800F4F F0AFD82FF1012h
y	07192B95FFC8DA78631011ED6B24CDD573F 977A11E794811h
Order	FFFFFFFFFFFFFFFFFFFFFFFF99DEF836146 BC9B1B4D22831h

The Public Key for MIFARE Ultralight AES is (LSByte first):
0453BF8C49B7BD9FE3207A91513B9C1D238ECAB07186B772104AB535F7D3AE63 CF7C7F3DD0D169DA3E99E43C6399621A86

Example: (all numbers are LSByte first)

- Card UID: 042F6892457080

- Signature:
  1824472A4CC927C7CA423F2B75E8E15C
  D26F682D3D633B3E032879B11D2E7C0E
  5BDC720D7D4F3AB04DEC7229EC213C89

6.1.2.1 Python code example

Following code example shows the signature verification in python3.

```
from ecdsa import VerifyingKey, NIST192p

uid = '042F6892457080'
sig = '1824472A4CC927C7CA423F2B75E8E15CD26F682D3D633B3E032879B11D2E7
C0E5BDC720D7D4F3AB04DEC7229EC213C89'

public_key = '0453BF8C49B7BD9FE3207A91513B9C1D238ECAB07186B772104AB5
35F7D3AE63CF7C7F3DD0D169DA3E99E43C6399621A86'

vk = VerifyingKey.from_string(bytes.fromhex(public_key), curve=NIST192p)

try:
    if(vk.verify_digest(bytes.fromhex(sig), bytes.fromhex(uid))):
        print("Signature verification passed")
except:
    print("Signature verification failed")
```

6.2 AES originality check

AES originality check is provided by a 3rd AES key, which is additionally present in the MIFARE Ultralight AES. This key's value is only known to NXP, and can only be used with NXP-tools to verify the originality of the MIFARE Ultralight AES. The key cannot be changed. It can only be used for the AES originality check, no other privileges are linked to this key.

The easiest way to verify the originality of a MIFARE Ultralight AES is by using the NXP TagInfo smartphone app, available on Playstore and iOS App Store.

Note: A particular MIFARE Ultralight AES IC can only be verified a limited amount of time using the online check!

7 MIFARE Ultralight AES user memory anti-tearing proposal

The MIFARE Ultralight AES implements anti-tearing support for OTP, lock bits and counters (see [DS5379]). This means that in case of a tear-off event either the old value or the new (just written) value can be retrieved. This section describes what measures a MIFARE Ultralight AES application can implement in order to ensure tear-off protection for the user data pages, which are not covered by build in tearing protection mechanism.

For the tearing application implementation, 2 memory areas having the same size are needed see Figure 4.

Figure 4. Tearing application implementation. t: now, t-1: last valid operation, t+1: next operation

The application data is stored in 2 memory locations. The application data also contains a timestamp and a CMAC. Every time a new update is needed i.e. new data has to be written, only the set of data with the older timestamp is updated. The CMAC is added to guarantee the integrity of the written application data, and is calculated over the App Data and the timestamp.

In particular, the Figure 4 shows a typical update of the Application Data done on the older Application Data set (timestamp = t-1). As soon as the new application data is written, the timestamp is updated (timestamp = t+1) and the CMAC is also written.

If the update operation fails due to a tearing event and the application data becomes corrupted, this can be recognized based on the failure of the CMAC validation. In any case, the MIFARE Ultralight AES either contains the latest updated application data (timestamp = t+1) or the previous one (timestamp = t).

In case AES authentication and CMAC-based secure messaging is used, a regular CRC calculated over the App Data (instead of the CMAC) is sufficient.

8 Reference documents

[DS5379]	Datasheet of MIFARE Ultralight AES - DS5379xx
[UM11764]	MIFARE Ultralight AES - Information on Guidance and Operation - DocStore number 7086xx[1]
[AN12653]	End to end system security risk considerations for implementing contactless cards and tags - AN12653
[AN10922]	Symmetric Key diversification - AN10922
[AN11340]	MIFARE Ultralight EV1 Features and hints - AN11340
[ISO/IEC 14443-3]	ISO/IEC 14443-3 Identification cards — Contactless integrated circuit cards — Proximity cards - Part 3: Initialization and anticollision
[NIST SP800-38A]	National Institute of Standards and Technology (NIST). NIST Special Publication 800-38A: Recommendation for Block Cipher Modes of Operation. December 2001
[NIST SP800-38B]	National Institute of Standards and Technology (NIST). NIST Special Publication 800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. May 2005
[ISO/IEC 9797-1]	Information technology Security techniques Message Authentication Codes

[1] xx … document version number

9 Legal information

9.1 Definitions

Draft — A draft status on a document indicates that 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 in a draft version of a document and shall have no liability for the consequences of use of such information.

9.2 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.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

9.3 Licenses

Purchase of NXP ICs with NFC technology — Purchase of an NXP Semiconductors IC that complies with one of the Near Field Communication (NFC) standards ISO/IEC 18092 and ISO/IEC 21481 does not convey an implied license under any patent right infringed by implementation of any of those standards. Purchase of NXP Semiconductors IC does not include a license to any NXP patent (or other IP right) covering combinations of those products with other products, whether hardware or software.

9.4 Trademarks

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

NXP — wordmark and logo are trademarks of NXP B.V.

DESFire — is a trademark of NXP B.V.

MIFARE — is a trademark of NXP B.V.

MIFARE Classic — 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.

Tables

Figures

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: 22 February 2022
Document identifier: AN13452