

AN1303

MIFARE Ultralight as Type 2 Tag

Rev. 1.4 — 21 April 2011
130314

Application note
COMPANY PUBLIC

Document information

Info	Content
Keywords	NFC Forum, NFC Forum data mapping, NFC Forum Type 2 Tag Operation, Type 2 Tag, MIFARE Ultralight, MIFARE Ultralight , MIFARE Ultralight C, NDEF Tag Application
Abstract	<p>The NFC Forum is a standardization consortium that was formed to advance the use of Near Field Communication technology by developing specifications, ensuring interoperability among devices and services, and educating the market about NFC technology.</p> <p>The NFC Forum has defined a data format called NDEF to store different kind of application data. NDEF structured data may be stored inside contactless tag.</p> <p>The NFC Forum the “Type 2 Tag Operation” technical specification has been developed to describe how the reader/writer device (called NFC Forum device) can store/retrieve NDEF data on a Type 2 Tag platform.</p> <p>The NXP products MIFARE Ultralight and MIFARE Ultralight C are compatible with the NFC Forum “Type 2 Tag Operation” technical specification. This document extends the information and the functionalities about how an NFC Forum device can manage the MIFARE Ultralight product family as an NFC Forum Type 2 Tag platform.</p>



Revision history

Rev	Date	Description
1.4	20110421	Changed Mifare to MIFARE, changed Mifare Ultralight X to MIFARE Ultralight Family Memory Layout, added the chapter "Admin Security Feature", removed UID from Card Identification Procedure
1.3	20110124	Security status changed into public, no content change
1.2	20070515	Corrected and updated section 2.2 "MIFARE Ultralight X Memory Layout". Updated section 2.4 "Card Identification Procedure". Added "8. ANNEX: MIFARE Ultralight C Memory Layout". Replaced the term "64 bytes MIFARE Ultralight" with "MIFARE Ultralight".
1.1	20070821	Corrected and rephrased some text element, added figures, added section 6.4.5 "Transitions from READ/WRITE to Ultralight BLOCKED READ-ONLY", added "Additional Features" chapter 7, rewording of chapter 2 and 3
1.0	20061111	Final revision
0.1	20060721	First draft version

Contact information

For additional information, please visit: <http://www.nxp.com>

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

1. Introduction

The NFC technology allows to access standard ISO 14443A card products as the MIFARE family. A specification to store data for any kind of service or application is currently specified in the NFC Forum and it is called NFC Data Exchange Format (NDEF, see [NDEF]). To store NDEF formatted data (also called NDEF data or NFC Forum data) inside current contactless card products a mapping model is required. The specification [NFCT2T] describes this mapping model and how the NFC Forum device manages an NFC Forum Type 2 Tag platform to store NFC Forum defined data.

MIFARE Ultralight Family products are ISO/IEC 14443 Type A compliant contactless cards. MIFARE Ultralight Family products are compliant to the NFC Forum specification [NFCT2T] and it can be used as NFC Forum Type 2 Tag platform.

This document specifies from the NFC Forum device perspective in Reader/Writer mode:

- how to identify a specific MIFARE Ultralight Family card IC (e.g. MIFARE Ultralight or MIFARE Ultralight C),
- how to format a MIFARE Ultralight Family card IC as NFC Forum Type 2 Tag,
- how to manage a MIFARE Ultralight Family card IC as NFC Forum Type 2 Tag, and
- how to make use of the additional features of the MIFARE Ultralight Family card IC when operating as NFC Forum Type 2 Tag.

1.1 Implementation Guidelines

Implementers MAY decide to NOT implement all the possible features (procedures, states...) that this document specifies but only the recommended ones that are needed to support [NFCT2T] using MIFARE Ultralight Family card, and the ones required by implementers themselves or customer requirements.

It is RECOMMENDED to implement at least the features listed below to support [NFCT2T] using MIFARE Ultralight Family card:

- the memory layout and the relative card identification procedure, see [chapter 2](#),
- the basic states: INITIALISED, READ/WRITE and READ-ONLY, see [chapter 6](#), and
- the formatting procedures, see [section 6.5](#).

Note that the [NFCT2T] mandates only the support of the mandatory NDEF Message TLV i.e. the 1st NDEF Message TLV even though this application note describes additional features.

1.2 Applicable Documents

[ISOIEC 14443-3]	ISO/IEC14443-3 Type A Identification Cards- Contactless Integrated circuit(s) cards- Proximity Cards- Part 3: Initialisation and Anticollision
[NDEF]	"NFC Data Exchange Format (NDEF)", NFC Forum™, Technical Specification, May 2006.
[RFC2119]	"Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, S. Bradner, Harvard University, March 1997.

[MFUL]	“MF0 IC U1, Functional Specification Contactless Single-trip Ticket IC”, NXP Semiconductors, Product Data Sheet, Revision 3.2, 3 April 2007, Document Identifier 028632.
[MFULC]	“MF0ICU2, MIFARE Ultralight C”, NXP Semiconductors, Product Data Sheet, Revision 3.1, 2 April 2009, Document Identifier 137631.
[NFCT2T]	“Type 2 Tag Operation”, NFC Forum™, Technical Specification, July 09, 2007.

1.3 Convention and notations

1.3.1 Representation of numbers

The following conventions and notations apply in this document unless otherwise stated.

Binary numbers are represented by strings of digits 0 and 1 shown with the most significant bit (msb) left and the least significant bit (lsb) right , “b” is added at the end.

Example: 11110101b

Hexadecimal numbers are represented is using the numbers 0 - 9 and the characters A – F, an “h” is added at the end. The Most Significant Byte (MSB) is shown on the left, the Least Significant Byte (LSB) on the right.

Example: F5h

Decimal numbers are represented as is (without any tailing character).

Example: 245

1.3.2 Terms and Definition

According to the [NDEF] technical specification the data is represented in Network Byte Order (i.e. big endian). This means Most Significant Byte first and Most Significant Bit first (i.e. MSB first, and msb first)

The NFC Forum terminology in [NFCT2T] is different from the NXP terminology in [MFUL]. [Table 1](#) shows these differences. The terminology used in this document is the NXP terminology unless explicitly specified.

Table 1. Terminology Differences between [NFCT2T] and [MFUL, MFULC]

NFC Forum Terminology	NXP Internal Terminology	Description
Memory Block	Memory Page	It indicates a group of 4 contiguous bytes
CC Bytes	OTP bytes	The Capability Container (CC) bytes are stored in the One Time Programmable (OTP) bytes of the MIFARE Ultralight Family cards (see [MFUL, MFULC]).

1.4 Special Word Usage

The key words "SHALL", "SHALL NOT", "REQUIRED", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are used to signify the requirements in this document.

SHALL and REQUIRED have the same meaning. SHOULD and RECOMMENDED have the same meaning. MAY and OPTIONAL mean also the same. The key words are interpreted as described in [RFC2119].

1.5 Acronyms or Definitions or Glossary

Table 2. Terms and definitions

Term	Definition
Blank card	A MIFARE Ultralight card after production phase with its default setting is called blank card.
block-locking bit	Lock bit that when set to 1b makes read-only (lock) the lock bits.
card	A MIFARE Ultralight contactless card
CC	Capability Container, the CC stores control data for managing the NFC Forum defined data inside the tag. The CC bytes and the OTP bytes are the same.
lock bit	Bit that when it is set to 1b locks from writing one or more pages.
Lock Control TLV	TLV block that specifies memory area inside the tag containing lock bits.
lsb	least significant bit
LSB	least significant byte
Message Control TLV	TLV block that specifies a reserved memory area inside the tag.
MIFARE Ultralight Family	In this document the term MIFARE Ultralight Family indicates the family of IC products covering both MIFARE Ultralight and MIFARE Ultralight C as well as possible future versions.
MIFARE Ultralight	Contactless IC product as described in [MFUL]
MIFARE Ultralight C	Contactless IC product as described in [MFULC].
msb	most significant bit
MSB	most significant byte
NDEF	NFC Data Exchange Protocol, see [NDEF]
NDEF Message	Data packet structured as specified by the [NDEF] specification.
NDEF Message TLV	TLV block that contains an NDEF Message
NFC	Near Field Communication
NFC Forum	Standardization body, see http://www.nfc-forum.org/home
NFC Forum device	Reader device that is able to read and write an NFC Forum Type 2 Tag compliant to [NFCT2T]. The NFC Forum device implements the features and functionalities described in this application notes to operate a MIFARE Ultralight Family tag as NFC Forum Type 2 Tag
NULL TLV	Single byte TLV block mainly used for padding.
OTP	One Time Programmable, bytes where the bits are only possible to set from 0b to 1b
Proprietary TLV	TLV block that contains proprietary data

Term		Definition
RF		Radio Frequency
RFU		Reserved for Future Use
Tag		A MIFARE Ultralight Family contactless card
Terminator TLV		Last TLV block of the tag
TLV		Type Length Value block, data structure element composed of Tag, Length and Value field, see [NFCT2T]
Type 2 Tag / Tag platform	Type 2	Tag defined in the NFC Forum specification [NFCT2T]. MIFARE Ultralight Family contactless card is compatible with the Type 2 Tag.
UID		Unique Identifier, also called serial number in the [MFUL, MFULC] specification

2. Memory Layout

Type 2 Tag platform described in [NFCT2T] is fully compatible with MIFARE Ultralight. Two memory layouts (or memory structures) have been defined in [NFCT2T]:

1. the static memory structure that is defined for tag with memory size equal to 64 bytes. The MIFARE Ultralight (see [MFUL]) is fully compliant to the static memory structure, and
2. the dynamic memory structure that is defined for tag with memory size bigger than 64 bytes. Bigger versions of MIFARE Ultralight are compliant with the dynamic memory structure. The MIFARE Ultralight C (see [MFULC]) is also fully compliant to the dynamic memory structure.

In the sections below the features of the memory layouts of the MIFARE Ultralight Family products are described in details. These features are used by this document to provide additional functionalities to the NFC Forum device to operate the MIFARE Ultralight Family products that are not specified in the [NFCT2T] technical specification but they are still compatible with.

2.1 MIFARE Ultralight Memory Layout

In this section a brief and informative presentation of the memory layout of the MIFARE Ultralight is provided (see also [MFUL]).

The memory layout of the MIFARE Ultralight is shown in [Fig 1](#). It is composed of 64 bytes divided in pages of 4 bytes each. Different memory areas are present:

- the One Time Programmable (OTP) area on page 3. In the [NFCT2T] and in this document this area is also called Capability Container (CC),
- the data area from page 4 to page 16 for a total of 48 bytes, and
- the lock bytes Lock0 and Lock1. These bytes are called in [NFCT2T] static lock bytes.

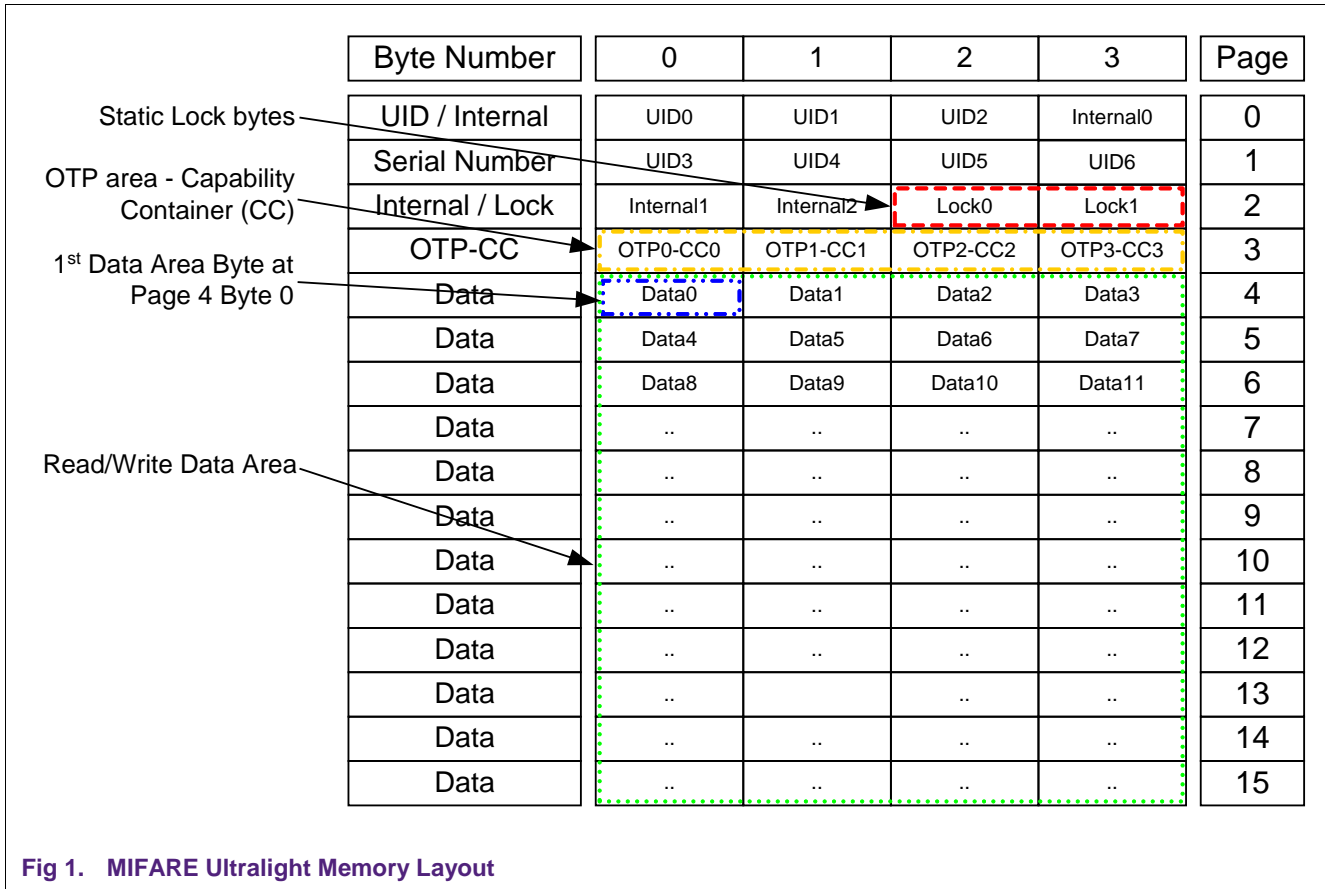


Fig 1. MIFARE Ultralight Memory Layout

The OTP area (or CC area in the NFC Forum terminology) contains special bits that can only be set to 1b and cannot reset back to 0b. For this reason the name One Time Programmable (OTP) bits. The write access of the OTP area is controlled by the lock bytes.

The data area is the 48 bytes memory that is possible to read and write.

The write access to the data area, OTP area and lock bytes, is controlled by the lock bytes.

2.1.1 Static Lock Bytes (Lock0, Lock1) Structure

The static lock bytes (Lock0 and Lock1) control the write access to the memory of the MIFARE Ultralight. The prefix “static” derives from the fix position of the lock bytes inside the memory layout.

Writing the static lock bytes is similar to the OTP bytes: lock bits can only be set to 1b and cannot be reset to 0b. Each lock bit when set to 1b makes read-only (i.e. locks) a specific page or a specific subset of lock bits. Fig 2 shows the static lock bits structure i.e. which pages or lock bits are controlled by the lock bits.

In particular, when setting to 1b:

- bit 7 of Lock1 of page 15 is locked,
- bit 7 of Lock0 of page 7 is locked,
- bit 3 of Lock0 of the OTP bytes are locked,

- bit 2 of Lock0 of the bits of Lock1 that lock the pages 10 to 15, are locked,
- bit 1 of Lock0 of the bits of Lock0 and Lock1 that lock the pages 4 to 9, are locked, and
- bit 0 of Lock0 of the bit 0 of Lock1 that locks the OTP bytes, are locked.

The bit 0 of Lock0, bit 1 of Lock0 and bit 2 of Lock0 lock the lock bits of Lock0 and Lock1. It is important to notice that setting to 1b the bits 0 to 2 of Lock0 freezes the locking configuration of the MIFARE Ultralight i.e. it is not possible to change from read/write to read-only the configuration of any page. Due to this capability bits 0 to bit 2 of Lock0 are also called block-locking bits.

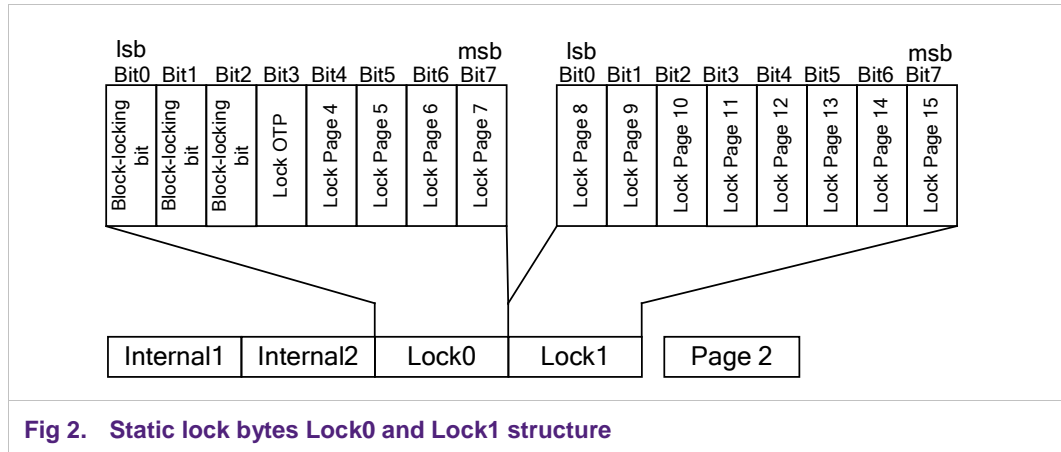


Fig 2. Static lock bytes Lock0 and Lock1 structure

2.1.2 MIFARE Ultralight Blank Card Settings

A MIFARE Ultralight card after production phase with the default setting is called blank card. To be a compliant NFC Forum Type 2 Tag a blank MIFARE Ultralight needs to be formatted (see [section 6.5](#)).

The default settings of a (blank) MIFARE Ultralight are (see [Fig 3](#)):

- The static lock bytes Lock0 and Lock1 are both set to 00h,
- The OTP bytes are set to 00h, and
- The bytes on page 4 are set to FFh.

Byte 0 and byte 1 of page 4 are also called Version Information, and they are used to identify the memory layout of the MIFARE Ultralight described in this section.

The byte settings of page 5 to page 15 are not specified.

Byte Number	0	1	2	3	Page	
Static Lock bytes	UID / Internal	UID0	UID1	UID2	Internal0	0
	Serial Number	UID3	UID4	UID5	UID6	1
OTP area - Capability Container (CC)	Internal / Lock	Internal1	Internal2	Lock0=00h	Lock1=00h	2
	OTP	OTP0=00h	OTP1=00h	OTP2=00h	OTP3=00h	3
Page 4 Bytes	Data	Data0=FFh	Data1=FFh	Data2=FFh	Data3=FFh	4
	Data	Data4	Data5	Data6	Data7	5
	Data	Data8	Data9	Data10	Data11	6
	Data	7
	Data	8
	Data	9
	Data	10
	Data	11
	Data	12
	Data	13
	Data	14
	Data	15

Fig 3. MIFARE Ultralight Blank Card Settings

2.1.2.1 Version Information

The Version Information (see Table 3) provides the version number, and it is contained in Byte 0 and Byte 1 on page 4 of a blank card. The Version Information in case of the MIFARE Ultralight is set to: Byte 0 equal to FFh and Byte 1 equal to FFh. Generally the Version Information contains the memory layout information of the MIFARE Ultralight, and it is also present in the MIFARE Ultralight Family products (see section 2.2.2.1).

Table 3. Version Information

Page 4	
Byte 0	Byte 1
version number	
Major Version Number	Minor Version Number

2.2 MIFARE Ultralight Family Memory Layout

The MIFARE Ultralight Family memory layout (see Fig 4) has the following characteristics:

- The memory area from Page 0 to 15 is backward compatible to the memory area of the MIFARE Ultralight i.e. same number of pages, same static lock bytes, same OTP bytes and same memory layout.
- The position of the 1st byte of the data area is fixed at page 4 Byte 0.
- The data area (user area as defined in [NFCT2T]) is a not fragmented memory in the sense that it does not contain any reserved or lock byte.
- The new lock bytes used to lock the pages after page 15 are located starting from the byte at the beginning of the page just after the end of the data area. These new lock bytes are called dynamic lock bytes (see [NFCT2T]).

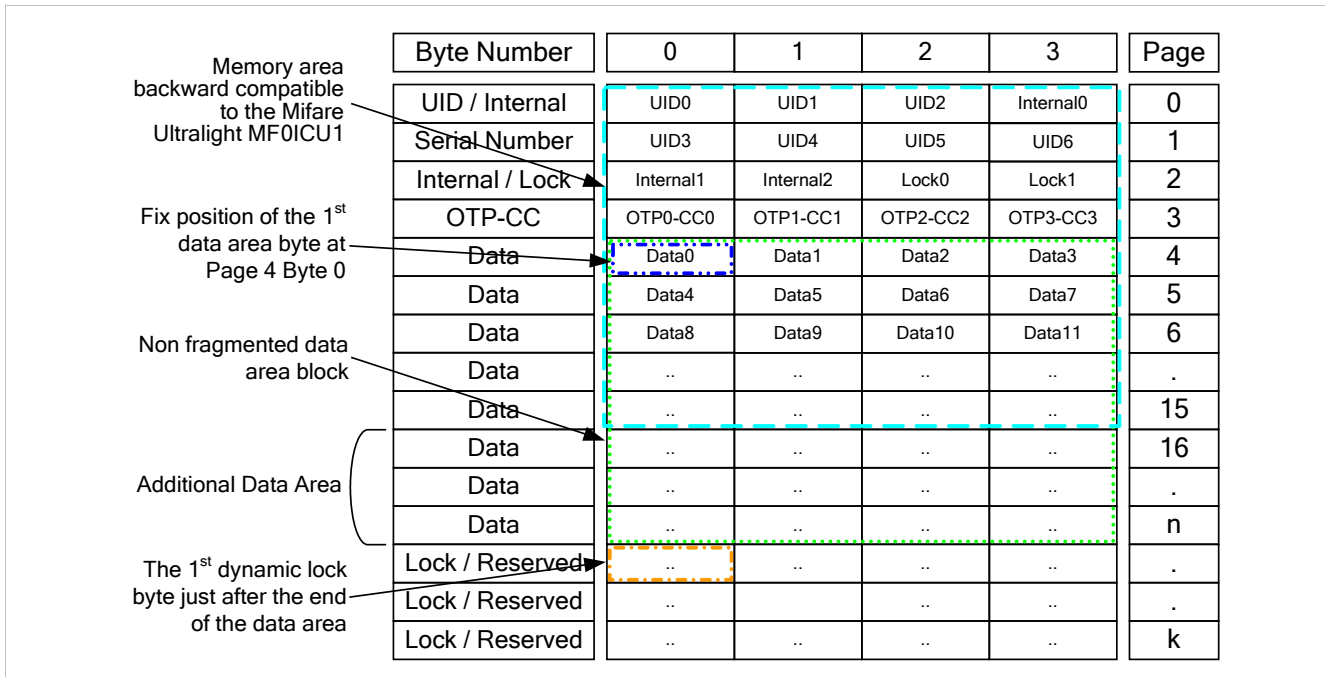


Fig 4. MIFARE Ultralight Family Memory Layout

Compare to the MIFARE Ultralight , the MIFARE Ultralight Family Memory Layout has:

- a memory size bigger than 64 bytes. The future version of MIFARE Ultralight Family products may be produced with different memory sizes. The part of the data area from page 15 to page n is called additional data area (see Fig 4), and
- a variable number of dynamic lock bits. The structure of the dynamic lock bits is more flexible and complex than the one of the static lock bits (a detailed description is provided in section 2.2.1).

The NFC Forum device SHALL be able to detect the memory layout previously described to control and manage the MIFARE Ultralight Family product. In section 2.2.2 it is explained how to determine the memory layout.

Note: One specific implementation of the MIFARE Ultralight Family memory layout is the MIFARE Ultralight C. For a dedicated description of the MIFARE Ultralight C see chapter 8.

2.2.1 Dynamic Lock Bytes Structure

The dynamic lock bits lock the memory area starting from page 16. Note that the pages 0-15 are locked by the static lock bits for backward compatibility with the MIFARE Ultralight . Note that the prefix “dynamic” of the lock bits derives from the implementation dependent memory position of these lock bits inside the memory layout.

The writing of the dynamic lock bits is similar to the OTP bytes: a dynamic lock bits can only be set to 1b and it cannot reset to 0b. Each dynamic lock bit when set to 1b makes read-only (i.e. locks) a specific locked area. A locked area may be composed of one or more pages. Therefore differently from the static lock bit, a dynamic lock bit is not restricted to lock only 1 page.

With the name “dynamic lock byte structure” it is meant: which locked area is controlled by which dynamic lock bit, and which are the block-locking bits of the dynamic lock bytes.

The dynamic lock byte structure, the locked area size and the block-locking bits are described by means of two examples in Fig 5, Fig 6 and Fig 7.

Byte Number	0	1	2	3	Page
UID / Internal	UID0	UID1	UID2	Internal0	0
Serial Number	UID3	UID4	UID5	UID6	1
Internal / Lock	Internal1	Internal2	Lock0	Lock1	2
OTP-CC	OTP0-CC0	OTP1-CC1	OTP2-CC2	OTP3-CC3	3
Data	Data1	Data1	Data2	Data3	4
Data
Data	15
Data	16
Data	17
Data	18
Data	19
Data
Data	n
Dynamic Lock	n+1
DynLock/Reserved	n+2
Reserved	n+3
Reserved
Reserved	k

2 Pages - Locked Area (blue-dashed-dotted square)

3 Pages - Locked Area (red-dashed square)

Dynamic Lock Bytes (green-dotted square)

Fig 5. 2 pages-locked area and 3 pages-locked area example.

Legend:

- the green-dotted square indicates the pages where the dynamic lock bytes are located. These pages may contain reserved bytes as well,
- the red-dashed square indicates an example of the locked area composed of 2 pages (called 2 pages – locked area), and
- the blue-dashed-dotted square indicates an example of the locked area composed of 3 pages (called 3 pages – locked area).

The 1st example of dynamic lock bytes structure regards a situation where the locked areas include 2 pages each (also called in this example 2 pages – locked area). The 2 pages – locked areas are indicated in Fig 5 with blue-dashed-dotted squares. A possible dynamic lock byte structure is shown in Fig 6. The bit 1 of DynLock0 controls the first locked area with page 16 and 17, the bit 2 controls the next locked area with page 18 and 19 and so on up to the last dynamic lock bit 3 of DynLock6. The bit 0 of DynLock0 to DynLock6 are a block-locking bit; if they are set to 1b, they freeze the locking configuration of the MIFARE Ultralight Family product i.e. it is not possible to change from read/write to read-only the configuration of any page of the data area. The reserved bits i.e. bit 4 to bit 7 of DynLock6 and the bits of the reserved byte, SHALL be always set by an NFC Forum device to 0b.

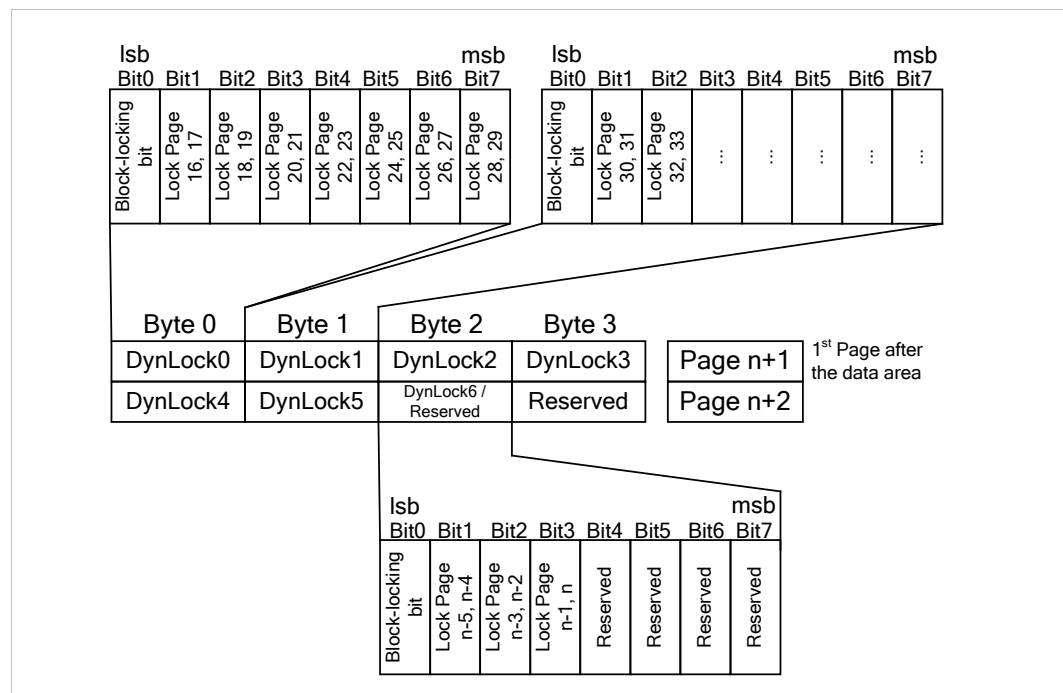


Fig 6. 1st Example of dynamic lock byte structure with 2 pages-locked area. For each dynamic lock bit the controlled pages are indicated.

The 2nd example of dynamic lock bytes structure regards a situation where the locked areas include 3 pages each (also called in this example 3 pages – locked area). The 3 pages – locked areas are indicated in Fig 5 with red-dashed squares. A possible dynamic lock byte structure is shown in Fig 7. The bit 1 of DynLock0 controls the first locked area with page 16, 17 and 18, the bit 2 controls the next locked area with page 19, 20 and 21 and so on up to the last dynamic lock bit 4 of DynLock5. The bit 0 of DynLock0, the bit 2 of DynLock1 ... up to the bit 0 of DynLock5 are block-locking bits; if they are set to 1b, they freeze the locking configuration of the MIFARE Ultralight Family product i.e. it is not possible to change from read/write to read-only the configuration of any page of the data area. The reserved bits i.e. bit 5 to bit 7 of DynLock5 and the bits of the reserved byte, SHALL be always set by an NFC Forum device to 0b.

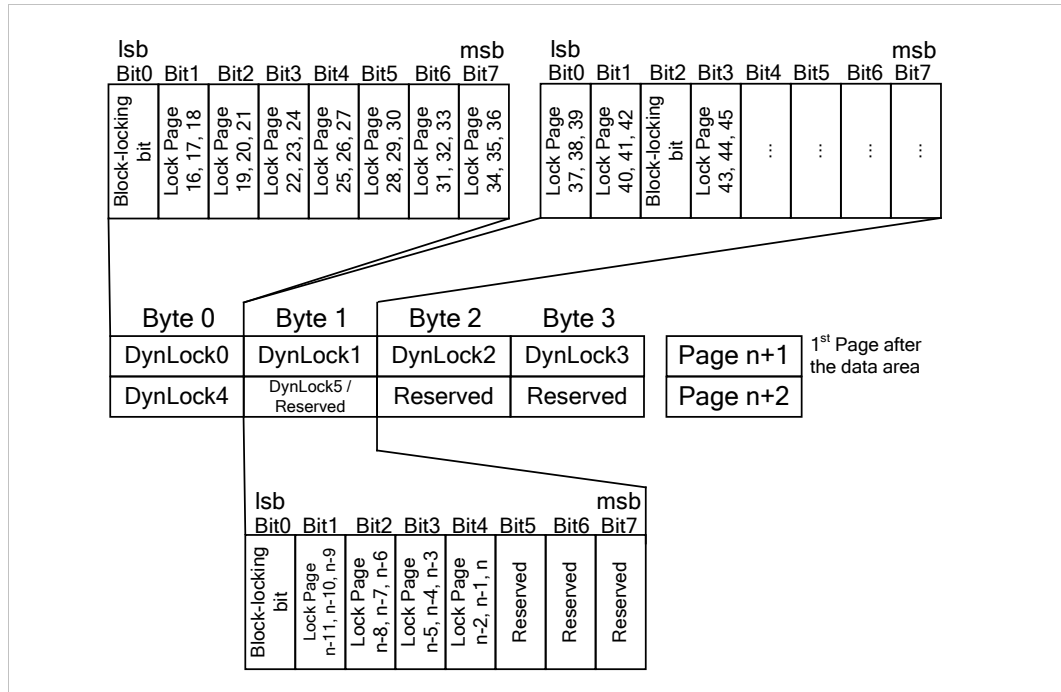


Fig 7. 2nd Example of dynamic lock byte structure with 3 pages-locked area. For each dynamic lock bit the controlled pages are indicated.

The location of the block-locking bits can be anyone inside the dynamic lock bytes and it is not limited to bit 0 of each dynamic lock byte. In case the last dynamic lock byte contains both dynamic lock bits and reserved bits, the dynamic lock bits are stored from the lsb to the msb (see examples).

The reserved bits and bytes after the dynamic lock bits in [Fig 6](#) and [Fig 7](#) are used to fill the last bits of the page SHALL be set to 0b.

2.2.2 MIFARE Ultralight Family product Blank Card Settings

A MIFARE Ultralight Family product after production phase with its default setting is called blank card. To be a compliant NFC Forum Type 2 Tag a blank MIFARE Ultralight Family card needs to be formatted (see [section 6.5](#)).

The default settings of a blank MIFARE Ultralight Family product card are (see [Fig 8](#)):

- The static lock bytes Lock0 and Lock1 are both set to 00h,
- The OTP bytes are set to 00h,
- The dynamic lock bytes are set to 00h, and
- From page 4 the MIFARE Ultralight Family Memory Layout Version Information is located (see below for more details).

The bytes after the MIFARE Ultralight Family Memory Layout Version Information memory area are not defined.

Byte Number	0	1	2	3	Page
Static Lock Bytes UID / Internal	UID0	UID1	UID2	Internal0	0
Serial Number	UID3	UID4	UID5	UID6	1
OTP area - Capability Container (CC) Internal / Lock	Internal1	Internal2	Lock0=00h	Lock1=00h	2
OTP	OTP0=00h	OTP1=00h	OTP2=00h	OTP3=00h	3
Mifare ultralight X Version Information Data	Major Version Number	Minor Version Number	LChunkSize MSB	LChunkSize LSB	4
Data	#ofLChunks MSB	#ofLChunks LSB	LChunksPer DynamicLockBit	#ofDynamic LockBits	5
Data	DynBlockLocking BitMask	DynBlockLocking BitMask	..	Data11	6
Data
Data	15
Data	16
Data
Dynamic Lock Bytes Data	n
DynLock/Reserved	DynLock0=00h	DynLock1=00h	n+1
DynLock/Reserved
Reserved	k

Fig 8. MIFARE Ultralight Family product Blank Card Settings

2.2.2.1 Version Information

The Version Information contained from page 4 onwards on a blank card provides details like the version number, the size and number of the locked areas related to the dynamic lock bytes and the dynamic lock byte structure. Table 4 shows the data structure of the Version Information starting from page 4.

Table 4. Version Information

Page 4			
Byte 0	Byte 1	Byte 2	Byte 3
version number		Size of locked chunk (LChunkSize)	
Major Version Number	Minor Version Number	LChunkSize MSB	LChunkSize LSB
Page 5			
Byte 0	Byte 1	Byte 2	Byte 3
Number of locked chunks (#ofLChunks)		Dynamic lock bits information	
#ofLChunks MSB	#ofLChunks LSB	LChunksPer DynamicLockBit	#ofDynamic LockBits
Page 6			
Byte 0	Byte 1	Byte 2	Byte 3
Dynamic Block Locking Bit Mask (starting from the lsb of Byte 0)			
DynBlockLockingBitMask			

Version Number (Page 4 Byte 0-1)

Page 4 Byte 0-1 indicates the MIFARE Ultralight Memory Layout version number, in particular:

- Page 4 Byte 0 contains the Major Version Number, and
- Page 4 Byte 1 contains the Minor Version Number.

For more information about the supported version numbers and how to manage them see [section 2.5.1](#).

Size of Locked Chunk (*LChunkSize*, Page 4 Byte 2-3)

The NFC Forum device SHALL use the values contained in Page 4 Byte 2-3 to retrieve the size of the Locked Chunk (called *LChunkSize*) in number of bytes, in particular:

- Page 4 Byte 2 contains the MSB of the *LChunkSize*, and
- Page 4 Byte 3 contains the LSB of the *LChunkSize*.

Number of Locked Chunks (*#ofLChunk*, Page 5 Byte 0-1)

The NFC Forum device SHALL use the values contained in Page 5 Byte 0-1 to retrieve the number of Locked Chunk (called *#ofLChunk*), in particular:

- Page 5 Byte 0 contains the MSB of the *#ofLChunk*, and
- Page 5 Byte 1 contains the LSB of the *#ofLChunk*.

To calculate the size in bytes of the additional data area (called *AdditionalDataAreaSize*) i.e. from Page 16 to Page n (see [Fig 4](#)) the NFC Forum device SHALL use the following formula:

$$\text{AdditionalDataAreaSize} = \text{LChunkSize} \cdot \text{\#ofLChunks} \text{ in [bytes]}$$

The *AdditionalDataAreaSize* is used later on in the formatting procedures to set the byte 2 of the Capability Container (see [section 6.5](#)).

Dynamic Lock bits Information (Page 5 Byte 2-3)

The NFC Forum device SHALL use the values contained in Page 5 Byte 2-3 to retrieve the information about the dynamic lock bits, in particular:

- Page 5 Byte 2 contains the number of Locked Chunk that are locked setting to 1b a dynamic lock bit (called *LChunksPerDynamicLockBit*), and
- Page 5 Byte 3 contains the number of dynamic lock bits (called *#ofDynamicLockBits*).

The size of a locked area (see [section 2.2.1](#)) in bytes is equal to:

$$\text{LockedAreaSize} = \text{LChunkSize} \cdot \text{LChunksPerDynamicLockBit}$$

Dynamic Block Locking Bit Mask (from Page 6)

The NFC Forum device SHALL use the values contained from Page 6 to identify the block-locking bits of the dynamic lock bits. The Dynamic Block Locking Bit Mask (*DynBlockLockingBitMask*) is coded as a bit mask. *DynBlockLockingBitMask* length in bits is equal to the *#ofDynamicLockBits*. Depending of the *#ofDynamicLockBits* value the *DynBlockLockingBitMask* can occupies one or more bytes starting from the lsb of byte 0 in Page 6.

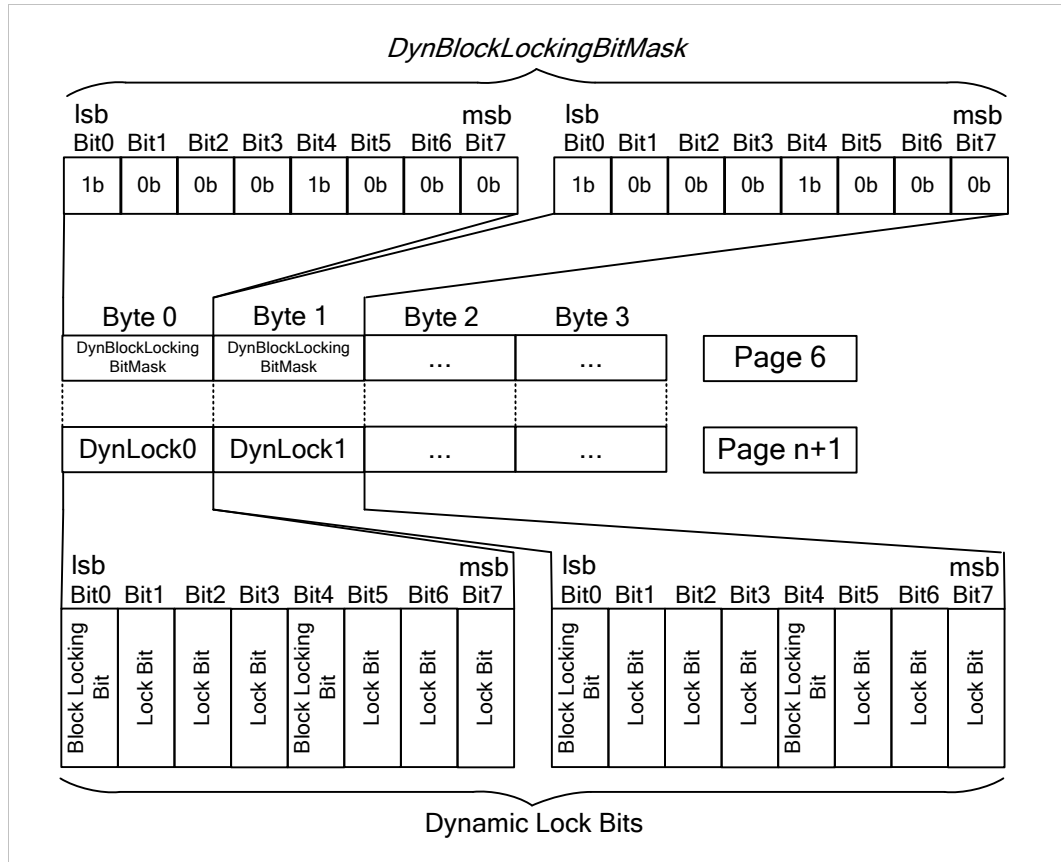


Fig 9. DynBlocklockingBitMask structure example

There is a one to one association between each bit of the *DynBlockLockingBitMask* and each bit of the dynamic lock bits (see Fig 9):

- The 1st bit (bit 0 page 6) of the *DynBlockLockingBitMask* field is associated to the 1st bit (bit 0 page n+1) of the dynamic lock bytes,
- The 2nd bit (bit 1 page 6) of the *DynBlockLockingBitMask* field is associated to the 2nd bit (bit 1 page n+1) of the dynamic lock bytes,
- ...and so on.

A bit set to 0b in the *DynBlockLockingBitMask* indicates a lock bit in the dynamic lock bytes. A bit set to 1b in the *DynBlockLockingBitMask* indicates a block-locking bit in the dynamic lock bytes.

Note that the *DynBlockLockingBitMask* does not indicate which lock bits a block-locking bit controls.

2.2.2.2 How to Retrieve the Dynamic Lock Byte Structure from the Version Information

This section describes how to retrieve the dynamic lock byte structure (see section 2.2.1) from the Version Information of the MIFARE Ultralight Family Memory Layout (see section 2.2.2.1).

As described in section 2.2.1 starting from page 16 of the MIFARE Ultralight Family product memory, the data area is divided into several locked areas. A locked area is

locked by a single dynamic lock bit i.e. the locked area is associated to a specific dynamic lock bit. The size in bytes of a locked area is indicated by the following formula:

$$LockedAreaSize = LChunkSize \cdot LChunksPerDynamicLockBit$$

The dynamic lock byte structure of MIFARE Ultralight Family product or in other words the association between dynamic lock bit and the locked areas is described below by means of an example (see also [Fig 5](#) and [Fig 6](#)):

- the locked area containing page 16-17 (8 bytes) is associated to bit 1 of DynLock0 (the lsb bit 0 is a block-locking bit),
- the locked area containing page 18-19 (8 bytes) is associated to bit 2 of DynLock0,
- the locked area containing page 20-21 (8 bytes) is associated to bit 3 of DynLock0,
- ...
- and so on until the last lock bit is reached.

Being the block-locking bits not associated to any locked area, they are not considered in the example. The block-locking bit position inside the dynamic lock bytes is indicated by the *DynBlockLockingBitMask*.

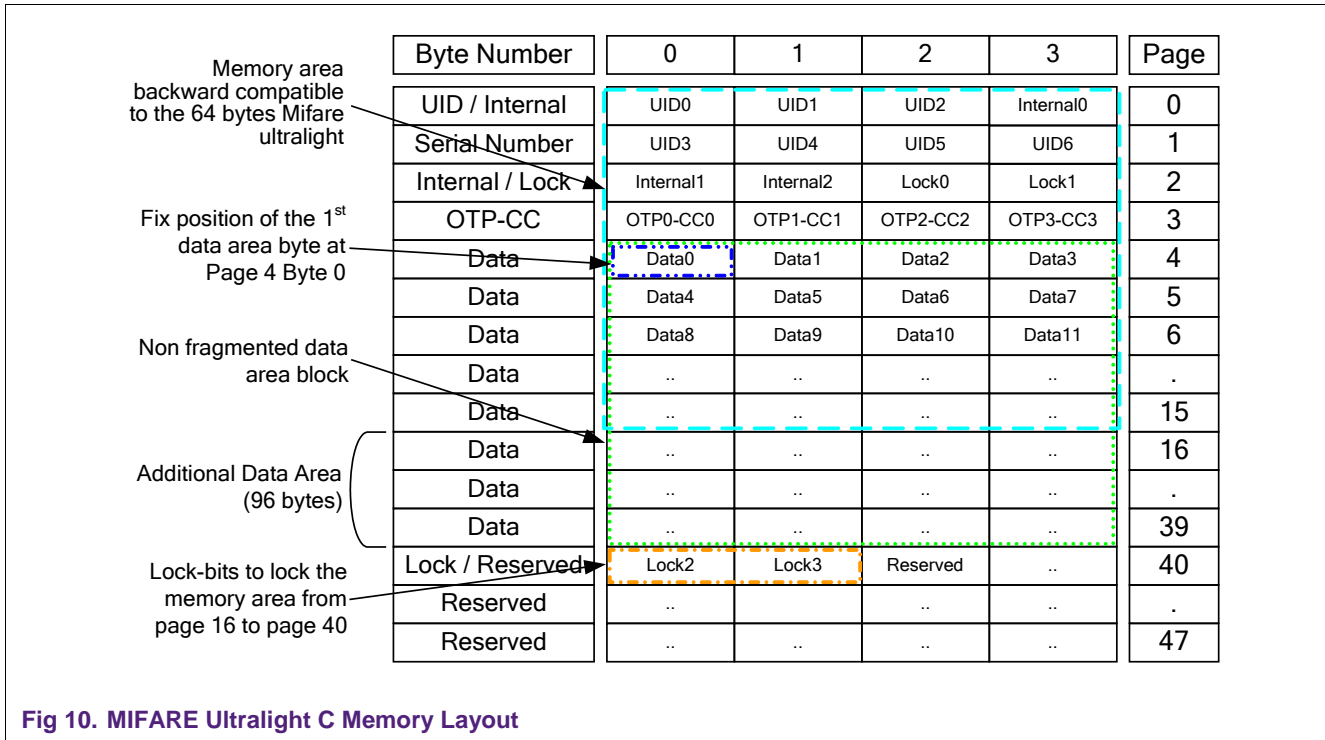
Another clarifying example is shown in [Fig 5](#) and [Fig 7](#) with a locked area size of 3 pages (12 bytes).

2.3 MIFARE Ultralight C Memory Layout

The MIFARE Ultralight C memory layout is a particular implementation of the MIFARE Ultralight Family Memory Layout (see [section 0](#)):

The memory layout of the MIFARE Ultralight C is shown in [Fig 10](#). MIFARE Ultralight C is composed of 192 bytes divided in pages of 4 bytes each. Different memory areas are present:

- The memory area from page 0 to 15 is backward compatible to the memory area of the MIFARE Ultralight i.e. same number of pages, same static lock bytes, same OTP bytes and same memory layout.
- The position of the 1st byte of the data area is fixed at Byte 0 of page 4.
- The data area (user area as defined in [NFCT2T]) is a not fragmented memory area in the sense that it does not contain any reserved or lock byte. Compare to the MIFARE Ultralight , MIFARE Ultralight C has an additional data area of 96 bytes.
- Two new lock bytes (called Lock2 and Lock3) lock the memory area from page 16 to page 39. Lock2 is located at Byte 0 of page 40 and Lock3 is located at Byte 1 of page 40. These two lock bytes are the dynamic lock bytes (see [NFCT2T]) of MIFARE Ultralight C.



Compare to MIFARE Ultralight, MIFARE Ultralight C has:

- a memory size of 144 bytes. The MIFARE Ultralight C has an additional data area from page 16 to page 39 called additional data area (see Fig 10), and
- two dynamic lock bytes on page 40 called Lock2 and Lock3.

The NFC Forum device SHALL be able to detect the memory characteristics previously described to be able to control and manage the MIFARE Ultralight C. In section 2.2.2.1 it is explained how to retrieve these memory characteristics from the Version Information.

2.3.1 Dynamic Lock Bytes (Lock2 and Lock3) Structure

In MIFARE Ultralight C the dynamic lock bits lock the memory area starting from page 16 up to page 39. Note that the pages 0-15 are locked by the static lock bits for backward compatibility with the MIFARE Ultralight . The prefix “dynamic” derives from [NFCT2T].

The writing of the dynamic lock bits is similar to the OTP bytes: a dynamic lock bits can only be set to 1b and it cannot be reset to 0b. Each dynamic lock bit when set to 1b makes read-only (i.e. locks) a specific locked area.

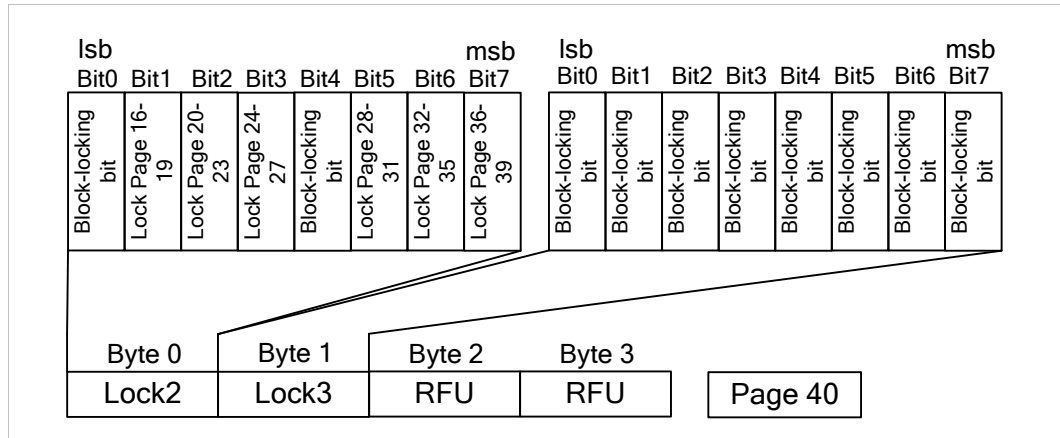


Fig 11. Dynamic lock byte Lock2 structure

Fig 11 describes in details the locking structure (i.e. which bit locks which pages and the block-locking bits) of the dynamic lock byte Lock2 and Lock3. In particular the following bits lock the pages of the additional data area:

- bit 1 (Isb) of Lock2 locks page 16 to page 19,
- bit 2 of Lock2 locks page 20 to page 23,
- bit 3 of Lock2 locks page 24 to page 27,
- bit 5 of Lock2 locks page 28 to page 31,
- bit 6 of Lock2 locks page 32 to page 35, and
- bit 7 (msb) of Lock2 locks page 36 to page 39.

Instead the following bits, called block-locking bits, are used to lock the locking configuration of Lock2 and Lock3:

- bit 0 (Isb) and bit 4 of Lock2,
- bit 0 (Isb) to bit 3 of Lock3, and
- bit 4 to bit 7 (msb) of Lock3.

It is important to notice that setting to 1b the bit 0 to bit 2 of Lock0, the bit 0 and the bit 4 of Lock2, and bit 0 to bit 7 of Lock3, the locking configuration of the MIFARE Ultralight C is frozen i.e. it is not possible to change from read/write to read-only the configuration of any page.

Note that for consistency reasons with the procedure described in [section 2.2.2.2](#), the bit 4 to bit 7 (msb) of Lock3 are considered by this Application Note block-locking bits.

2.3.2 Blank Card Settings

A MIFARE Ultralight C card after production phase with the default setting is called blank card. To be a compliant NFC Forum Type 2 Tag a blank MIFARE Ultralight C card needs to be formatted (see [section 6.5](#)).

Byte Number	0	1	2	3	Page	
Static Lock Bytes	UID / Internal	UID0	UID1	UID2	Internal0	0
	Serial Number	UID3	UID4	UID5	UID6	1
OTP area - Capability Container (CC)	Internal / Lock	Internal1	Internal2	Lock0=00h	Lock1=00h	2
	OTP	OTP0=00h	OTP1=00h	OTP2=00h	OTP3=00h	3
Mifare Ultralight 2 Version Information	Data	Major Version Number = 02h	Minor Version Number = 00h	LChunkSize MSB = 00h	LChunkSize LSB = 10h	4
	Data	#ofLChunks MSB = 00h	#ofLChunks LSB = 06h	LChunksPerDynamicLockBit = 01h	#ofDynamicLockBits = 10h	5
	Data	DynBlockLocking BitMask = 11h	DynBlockLocking BitMask = FFh	Data10	Data11	6
	Data
	Data	15
	Data	16
	Data
Dynamic Lock Byte Lock2	Data	39
	DynLock/Reserved	Lock2=00h	Lock3=00h	Reserved	..	40
	Reserved
	Reserved	47

Fig 12. MIFARE Ultralight C Blank Card Settings

The default settings of a (blank) MIFARE Ultralight C card are (see Fig 12):

- The static lock bytes Lock0 and Lock1 are both set to 00h,
- The OTP bytes are set to 00h,
- The Version Information on page 4 to page 6 is set to (see section 2.2.2.1):
 - Major version number equal to 02h,
 - Minor version number equal to 00h,
 - Locked chunk size (LChunkSize) equal to 0010h,
 - Number of locked chunks (#ofLockedChunks) equal to 0006h,
 - Locked chunk per dynamic lock bit (LChunkPerDynamicLockBit) equal to 01h,
 - Number of dynamic lock bits (#ofDynamicLockbits) equal to 10h, and
 - Dynamic block-locking bit-mask (DynBlockLockingBitMask) equal to 11FFh.
- The dynamic lock byte Lock2 and Lock3 are both set to 00h.

The byte 1 to 3 of Page 6 and Page 7 to Page 39 is not specified. The reserved bytes are also not specified.

2.3.3 MIFARE Ultralight C Admin Security Feature

MIFARE Ultralight C allows the 3DES authentication for protected data access. This capability can be used also when the MIFARE Ultralight C is configured as NFC Forum Type 2 Tag to grant the write access by means of a successful 3DES authentication before an actual write operation can be done. The owner of the 3DES authentication key is the administrator (i.e. Admin) of the MIFARE Ultralight C, in the sense that it is the only entity capable to modify the write protected data area. This feature is called Admin Security Feature.

The Admin Security Feature can be activated and used at any state of the MIFARE Ultralight Life Cycle (see [section 6.2](#)) and even during the formatting (see [section 6.5](#)). To activate and operate the MIFARE Ultralight C Admin Security Feature, see 3DES Authentication in [MFULC].

The 3DES authentication key MUST be diversified when using the Admin Security Feature.

2.4 Mapping of NFC Forum data

The NFC Forum device SHALL map NFC Forum data inside MIFARE Ultralight Family Products using the TLV blocks defined in [NFCT2T]. The TLV blocks are contained inside the data area according to [NFCT2T]. The MIFARE Ultralight Family Products contains the following TLV blocks:

1. one or more NDEF Message TLV,
2. zero, one or more Proprietary TLV,
3. zero, one or more Lock Control TLV,
4. zero, one or more Memory Control TLV,
5. zero, one or more NULL TLV, and
6. zero or one Terminator TLV.

To set a blank MIFARE Ultralight Family card into a valid Type 2 Tag platform the NFC Forum device SHALL use the formatting procedure (see [section 6.5](#)). Only then it is possible to store the above named TLV blocks.

2.5 Card Identification Procedure

The NFC Forum device SHALL only use the card identification procedures to identify a MIFARE Ultralight Family and to format such. Two card identification procedures have been specified:

1. card identification procedure for blank card (i.e. cards with settings defined after the production phase, see [section 2.5.1](#)), and
2. card identification procedure when the card is in a valid state according to [chapter 6](#) (see [section 2.5.2](#)).

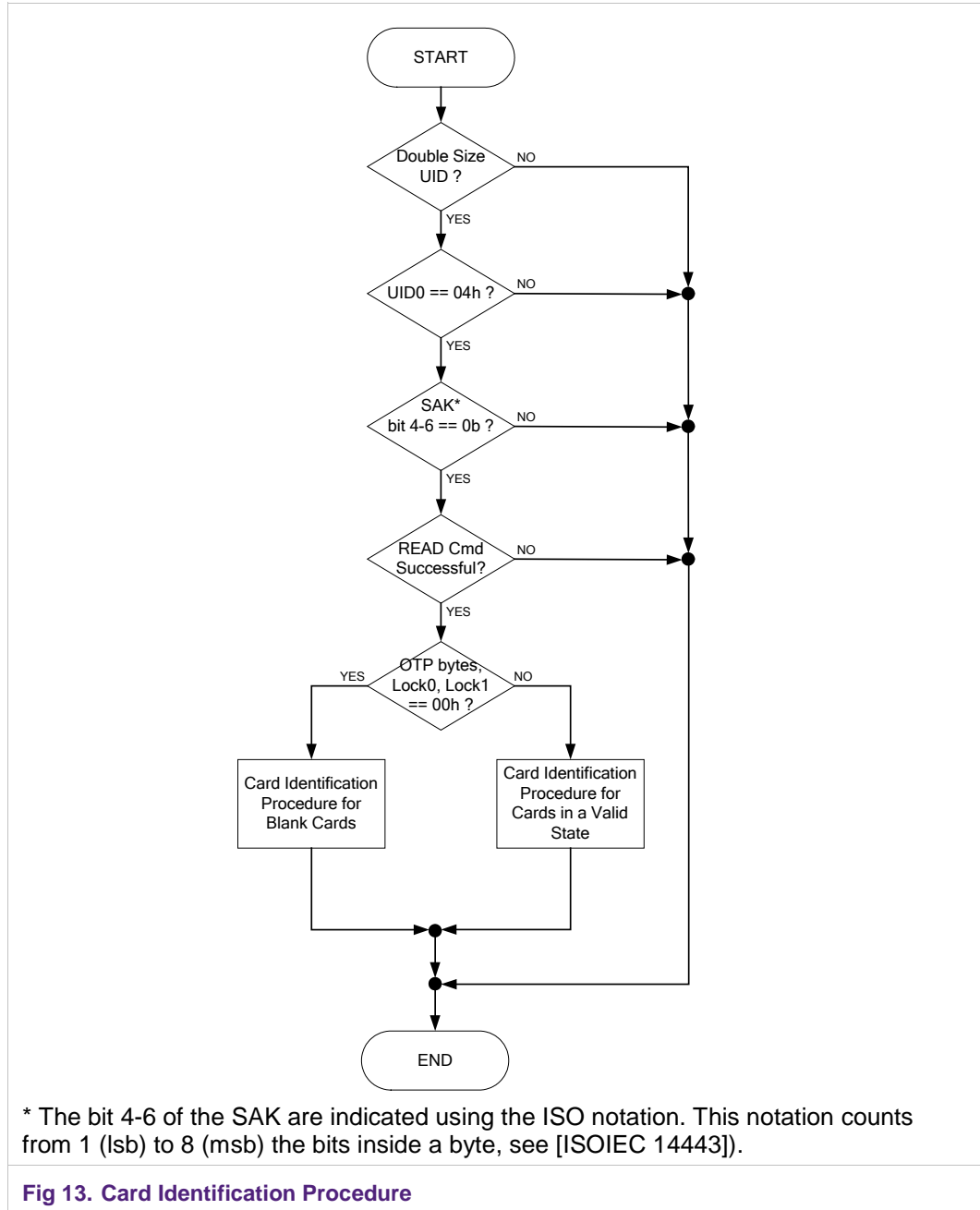
The card identification procedures retrieve the following information:

- the card is a MIFARE Ultralight Family Product,
- which type of MIFARE Ultralight Family Product the card is,
- the memory layout and locking structure, and
- additional information depending of the setting of the card e.g. dynamic lock bits, settings of the MIFARE Ultralight Family product...

To identify a MIFARE Ultralight the NFC Forum device SHALL perform the following common steps to the two card identification procedure (see [Fig 13](#)):

1. Check the Unique Identifier (UID) to be double size (7 bytes, see [ISOIEC 14443-3]),
2. Check inside the UID the byte UID0 (manufacturer ID) to be equal to 04h to indicate the NXP manufacturer.
3. Check that bit 4, bit 5 and bit 6 using ISO notation (this notation counts from 1 (lsb) to 8 (msb) the bits inside a byte, for more information see [ISOIEC 14443]) of the Selective Acknowledge (SAK, see [ISOIEC 14443-3]) are equal to 0b.

4. Use the READ command (see [chapter 5](#)) to read the memory area starting from Page 2. The READ command reads 16 bytes from Page 2 to 5.
5. If the READ command returns successfully, the card is a MIFARE Ultralight Family Product.
6. If the 2 bytes of the lock bits in Page 2 (i.e. the static lock bits of Lock0 and Lock1) and the 4 OTP bytes in Page 3 (see [MFUL]) are all equal to 00h. The MIFARE Ultralight Family card may be a blank card. The NFC Forum device SHALL use the card identification procedure for blank cards, see [section 2.5.1](#).
7. If any byte of the static lock bytes or the 4 OTP bytes in Page 3 (see [MFUL]) is different from 00h, the MIFARE Ultralight Family card may contain NFC Forum data. The NFC Forum device SHALL use the card identification procedure for cards in a valid state according to [chapter 6](#), see [section 2.5.2](#).



According to the NFC Forum specification [NFCDIGPROT] a generic tag is a Type 2 Tag Platform if bit b7-6 (using the [NFCDIGPROT] notation) of the SEL_RES (a.k.a. SAK) are equal to 00b. The Card Identification Procedure is consistent with the [NFCDIGPROT] and gives the additional information if the Type 2 Tag Platform is a MIFARE Ultralight Family product.

2.5.1 Card Identification Procedure for Blank Cards

After production phase the MIFARE Ultralight Family card is blank and it contains the Version Information (see [section 2.1.2.1](#) and [section 2.2.2.2](#)). In particular the NFC Forum device SHALL use the data contained in Byte 0-1 of Page 4 to retrieve the Major and Minor Version Number that identify the MIFARE Ultralight Family memory layout. The identification of the memory layout is needed by the Formatting Procedures (see [section 6.5](#)).

Note that the Version Information MAY be changed using normal WRITE operation. In case the MIFARE Ultralight Family card contains NFC Forum data the Version Information is not present. For this reason the card identification procedure described here is applied only blank cards.

The Major and Minor Version Number that this application note supports are described in [Table 5](#).

Table 5. Version Number supported by this application note

Major Version Number (Page 4, Byte 0)	Minor Version Number (Page 4, Byte 1)	Note
FFh	FFh	It indicates that the card is a MIFARE Ultralight memory layout. This MIFARE Ultralight is supported by this application note, see section 2.1.2 .
02h	00h	It indicates that the card is a MIFARE Ultralight Family memory layout. This MIFARE Ultralight Family memory layout is supported by this application note, see section 2.2.2 .

2.5.1.1 Version Treating

The version number of the Version Information is indicated with two numbers: major version number and minor version number. The handling of the different version numbers applied to the MIFARE Ultralight Family tag (called MFVNo) and the one implemented in the NFC Forum device (called NFCDevVNo) is explained in the 4 cases of [Table 6](#).

Table 6. Handling of the Version Number of the Version Information

No	Version Number Case	Handling
1	Major NFCDevVNo is equal to major MFVNo, and minor NFCDevVNo is bigger than or equal to minor MFVNo	The NFC Forum device SHALL access the MIFARE Ultralight Family tag and SHALL use all features of the application note applied to this MIFARE Ultralight Family tag.
2	If major NFCDevVNo is equal to major MFVNo, and minor NFCDevVNo is lower than minor MFVNo	Possibly not all features of the MIFARE Ultralight Family tag can be accessed. The NFC Forum device SHALL use all its features of the supported application note and SHALL access this Type 2 tag platform.

No	Version Number Case	Handling
3	If major NFCDevVNo is smaller than major MFVNo	Incompatible data format. The NFC Forum device cannot understand the MIFARE Ultralight Family tag data. The NFC Forum device SHALL reject this MIFARE Ultralight Family tag.
4	If major NFCDevVNo is bigger than major MFVNo	The NFC Forum device might implement the support for previous versions of this application note in addition to its main version. In case the NFC Forum device has the support from previous version, it SHALL access the MIFARE Ultralight Family tag. On the contrary, in case the NFC Forum device has not the support from previous version, it SHALL reject the MIFARE Ultralight Family tag.

2.5.2 Card Identification Procedure for Cards in a Valid State

In case any byte of the static lock bytes (i.e. Lock0 or Lock1) in page 2 or any byte of the 4 OTP bytes in Page 3 (see [MFUL, MFULC]) is different from 00h, the MIFARE Ultralight Family card can contain NFC Forum data. In this case the NFC Forum device SHALL apply the card identification procedure for cards in a valid state to check if the card is really in a valid state according to [chapter 6](#), and to retrieve the memory layout.

Note that the Version Information (see [section 2.2.2](#)) is not present in cards in a valid state.

To perform the card identification procedure when the card is in a valid state (see [chapter 6](#)), the NFC Forum device SHALL:

1. apply the NDEF detection procedure (see [NFCT2T]) to detect the presence of the 1st NDEF Message TLV (the mandatory one) and to check the MIFARE Ultralight Family card is in a valid state (see [section 6](#)).
2. If the Byte 2 of the OTP page is equal to 06h (i.e. data area is 64 bytes big, see [NFCT2T] [section 6.1](#)), the card is a MIFARE Ultralight .
3. If the Byte 2 of the OTP page is bigger than 06h, the card is a MIFARE Ultralight Family product.

After having identified the card, the position and size of the static and dynamic lock bits SHALL be retrieved. In case of MIFARE Ultralight only the static lock bytes are present (for position and size of the static lock bytes see [section 2.1](#)). Concerning MIFARE Ultralight Family product, the position and size of the static lock bits is the same as for the MIFARE Ultralight , and the position and size of the dynamic lock bits are provided by the Lock Control TLV (see [NFCT2T] [section 2.3](#)).

In case more than one Lock Control TLV is present in the MIFARE Ultralight Family product, only the 1st one (i.e. located in the lowest pages) SHALL be used to derive the locking structure. Below it is explained how an NFC Forum device SHALL derive the position and size of the dynamic lock bits from the Lock Control TLV:

- The size of the dynamic lock bits in number of bits is equal to the *Size* field of the Lock Control TLV indicates, and
- The starting position of the dynamic lock bits in the memory area is indicated by the *ByteAddr* value in number of bytes derived using the formula described in [NFCT2T] [section 2.3](#).

Note that it is not possible to retrieve the locking structure of the dynamic lock bits i.e. location of the block-locking bits and the lock bit – locked area association.

3. Read/Write Access

The read/write access right of the MIFARE Ultralight Family tag is based on the static and dynamic locking bit structure (see [chapter 2](#)).

It is important to highlight that the MIFARE Ultralight Family tag has always read access i.e. MIFARE Ultralight Family tag is always readable. The static and dynamic locking bit structure controls only the write access.

4. Framing / Transmission Handling

The framing and the transmission handling for MIFARE Ultralight Family tag is specified in [NFCT2T].

5. Command Set

The command set specified in the [NFCT2T] document is the same as specified in the [MFUL] except the command SECTOR SELECT. The SECTOR SELECT command SHALL be implemented in MIFARE Ultralight Family product bigger than 1Kbytes.

The following commands have to be supported by the NFC Forum device in order to read from or write to MIFARE Ultralight Family tag the NFC Forum defined data. The commands are the same that are specified in the NFC Forum technical specification [NFCT2T].

- READ
- WRITE
- SECTOR SELECT

6. Life Cycle

A MIFARE Ultralight Family product can be classified into several states. The state is reflected by the content of the MIFARE Ultralight Family product. A transaction is a set of operations to change the state on the tag starting from a specific state. Each state has its own valid transactions. In a specific moment in time the MIFARE Ultralight Family product is in only one state. An entry is an operation to prepare the MIFARE Ultralight Family product blank card (see [chapter 2](#)) into a specific state. The entries are also called formatting procedures (see [section 6.5](#)).

In this document two life cycles are presented:

- *The NFC Forum life cycle* (see [section 6.1](#)). This life cycle is the life cycle specified by the NFC Forum. The transitions and entries of this life cycle SHALL be implemented to have a reader device capable to act as NFC Forum device for Type 2 Tag platform, and to operate the MIFARE Ultralight Family product in order to switch between the different states. This application note describes in details the entries mentioned in [NFCT2T].
- *The MIFARE Ultralight life cycle* (see [section 6.2](#)). This life cycle shows the life cycle specified by the NFC Forum together with additional states and transitions that make use of all MIFARE Ultralight Family features. The additional transitions, and states MAY be partially implemented due to specific requirements and scenarios that are not covered by the NFC Forum.

6.1 NFC Forum Life Cycle

The specification [NFCT2T] describes the life cycle from the NFC Forum device perspective as a combination of states, transitions and entries. [Fig 14](#) describes the life cycle defined in the NFC Forum for the Type 2 Tag platform (called NFC Forum life cycle). The entries also called formatting procedures are just mentioned in the [NFCT2T] specification and they are described in details in [section 6.5](#) of this application note.

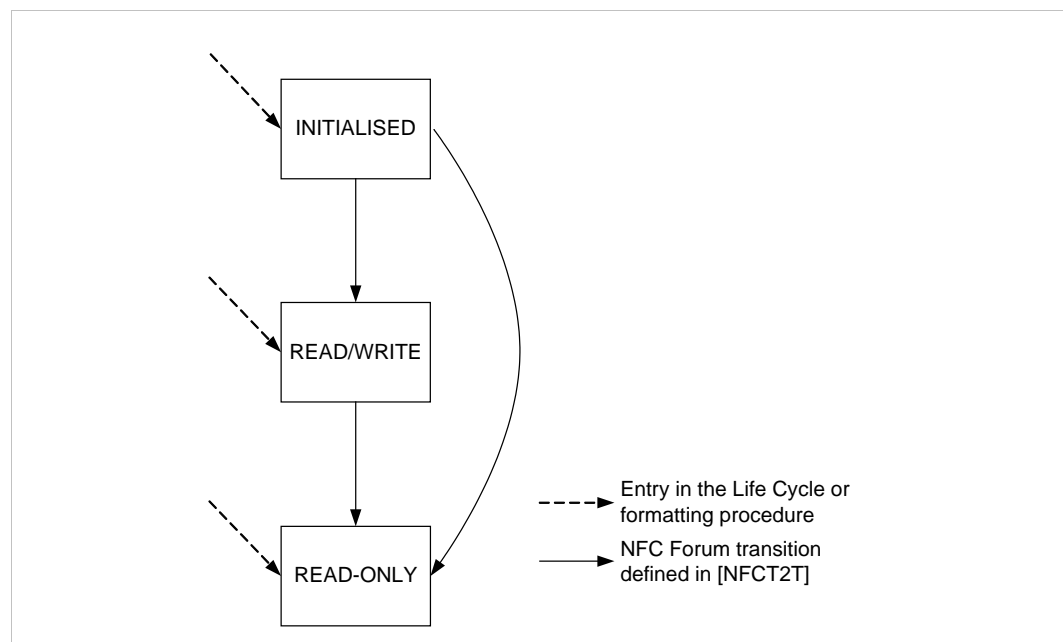


Fig 14. NFC Forum Life Cycle

6.2 MIFARE Ultralight Life Cycle

The MIFARE Ultralight Family tag MAY have additional states than the ones specified in the [NFCT2T]. These additional states together with additional transitions create the MIFARE Ultralight life cycle (see [Fig 15](#)).

The additional states are named in [Fig 15](#) using the prefix “Ultralight” being MIFARE Ultralight Family specific. A rounded square indicates parts of the MIFARE Ultralight life cycle that corresponds to the NFC Forum life cycle (see [Fig 14](#)).

To guarantee compatibility with the [NFCT2T] specification, the Ultralight READ/WRITE state, Ultralight BLOCKED READ/WRITE state, Ultralight READ-ONLY state and Ultralight BLOCKED READ-ONLY state are seen from a NFC Forum device that implements only the [NFCT2T] specification as READ/WRITE state and READ-ONLY state. [Table 7](#) shows the relations between the states defined in [NFCT2T] (see [Fig 14](#)) and the states defined in the MIFARE Ultralight life cycle (see [Fig 15](#)).

Table 7. Relation between the [NFCT2T] states and the states defined by this application note

States detected by a NFC Forum device implementing only the [NFCT2T] technical specification	States detected by a NFC Forum device implementing this application note
INITIALISED	INITIALISED
READ/WRITE	READ/WRITE
	Ultralight READ/WRITE
	Ultralight BLOCKED READ/WRITE
READ-ONLY	READ-ONLY
	Ultralight READ-ONLY
	Ultralight BLOCKED READ-ONLY

The states Ultralight READ/WRITE, Ultralight BLOCKED READ/WRITE, Ultralight READ-ONLY and Ultralight BLOCKED READ-ONLY have been introduced to be able:

- to block the overall configuration of the tag locking the lock bits,
- to manage the lock bits in a more flexible way e.g. being able to make specific pages read/write or read-only, and
- to be compatible with the [NFCT2T] specification as described in [Table 7](#).

In [Fig 15](#) the dotted arrows indicate the additional transitions that are not described in the [NFCT2T]. The additional transitions are MIFARE Ultralight Family specific.

The entries or formatting procedures are the same ones described in the NFC Forum life cycle (see [section 6.1](#)).

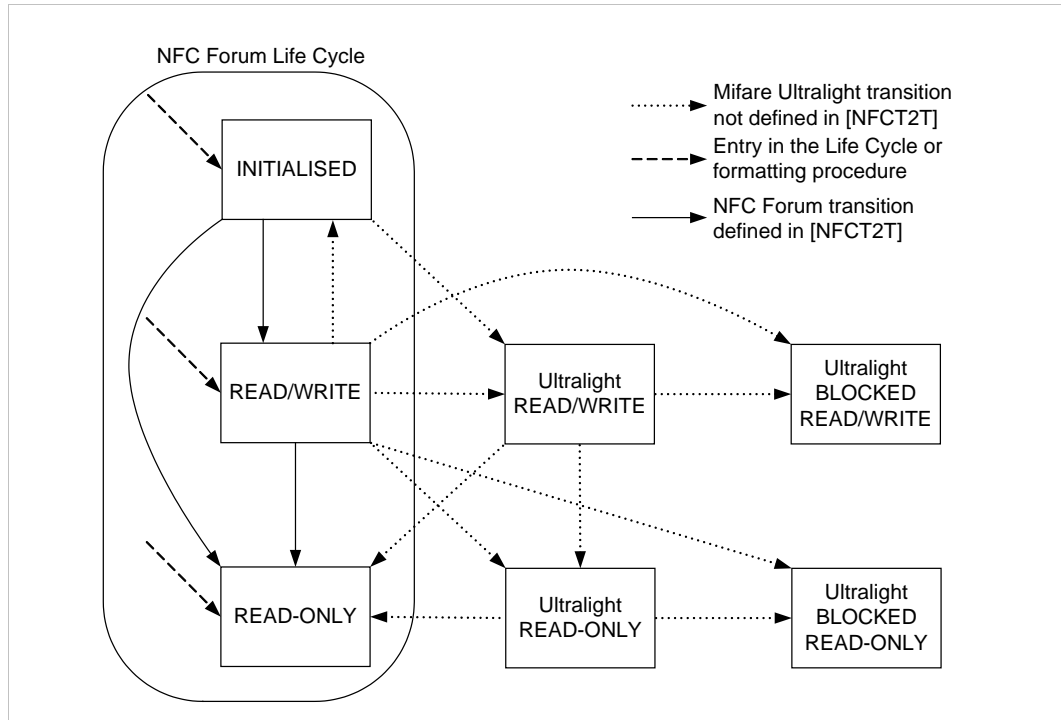


Fig 15. MIFARE Ultralight Life Cycle

Overall there are 7 states, 3 entries/formatting procedures and 14 transitions. It is not mandatory to support all states, entries and transitions in the NFC Forum device. An implementation that supports the NFC Forum Type 2 Tag platform MAY be tailored to support a subset of states, entries and transitions of the MIFARE Ultralight Life Cycle.

During any formatting procedure or transition the MIFARE Ultralight Family product needs to be held close to the NFC Forum device to be continuously powered. This can be communicated to the user e.g. showing a message on the user interface. It is up to the implementers to design the recovery mechanisms in case of interruptions or errors during a formatting procedure or a transition.

The identification of the states is based on: different settings of the lock bits (including the block-locking bits), different content in the data area, and different setting of the capability container (i.e. OTP bytes or CC bytes). [Table 8](#) compares the MIFARE Ultralight Life Cycle states highlighting the differences between them. The lock bits include both static and dynamic ones. A more comprehensive description of the states is given in [section 6.3](#).

Table 8. Comparative table between the different states

STATE	Block-locking bits	Lock bits of the CC page and of the 1st NDEF Message TLV pages without block-locking bits	Lock bits without block-locking bits and Lock bits of the CC page and of the 1st NDEF Message TLV pages	OTP-CC Byte 3 (R/W access)	Access to TLV blocks
INITIALISED	All set to 0b	All set to 0b	All set to 0b	00h	empty 1 st NDEF Message TLV
READ/WRITE	All set to 0b	All set to 0b	All set to 0b	00h	Non-empty 1 st NDEF Message TLV
Ultralight READ/WRITE	All set to 0b	All set to 0b	Someone set to 1b	00h	pages of the 1 st NDEF Message TLV SHALL NOT be locked; pages of other NDEF Message TLVs, Lock Control TLV, Memory Control TLV, Proprietary TLV MAY be locked
Ultralight BLOCKED READ/WRITE	All set to 1b	All set to 0b	Do not care	00h	pages of the 1 st NDEF Message TLV SHALL NOT be locked; pages of other NDEF Message TLVs, Lock Control TLV, Memory Control TLV, Proprietary TLV MAY be locked
READ-ONLY	All set to 1b	All set to 1b	All set to 1b	0Fh	-
Ultralight READ-ONLY	All set to 0b	All set to 1b	Someone set to 0b	0Fh	pages of the 1 st NDEF Message TLV SHALL be locked; pages of other NDEF Message TLVs, Lock Control TLV, Memory Control TLV, Proprietary TLV MAY NOT be locked
Ultralight BLOCKED READ-ONLY	All set to 1b	All set to 1b	Do not care	0Fh	pages of the 1 st NDEF Message TLV SHALL be locked; pages of other NDEF Message TLVs, Lock Control TLV, Memory Control TLV, Proprietary TLV MAY NOT be locked

In the [NFCT2T] the states INITIALISED, READ/WRITE and READ/ONLY are identified using the OTP-CC Byte 3 and the 1st NDEF Message TLV (the mandatory one, see [NFCT2T]). However to identify any of the additional 4 states (Ultralight READ/WRITE, Ultralight BLOCKED READ/WRITE, Ultralight READ-ONLY and Ultralight BLOCKED READ-ONLY) also the values of the lock bits and block-locking bits is needed.

The NFC Forum device that implements one or more of the additional states SHALL identify the states using the values of the lock bits and block-locking bits, and the 1st NDEF Message TLV. The OTP-CC Byte 3 is overridden by the values of the lock bits and block-locking bits. The NFC Forum device SHALL always set the OTP-CC Byte 3 following [Table 8](#).

6.3 States

This section describes how the NFC Forum device SHALL identify the different states shown in [section 6.2](#) and [Table 8](#). If needed the description of the states already described in the NFC Forum specification [NFCT2T] is explained more in detail.

It is supposed that the Capability Container (CC, or OTP bytes) is set correctly as described in [NFCT2T]. In particular the value of the byte 3 of the CC is not use to identify the state however it SHALL be set correctly as define in the previous [section 6.2](#) to be compatible with the [NFCT2T] specification.

The sections below complete the settings for each MIFARE Ultralight state that are described in the previous [section 6.2](#) and [Table 8](#).

In the sections below the term “lock bits” indicates the overall lock bits including the block-locking bits apart when where explicitly mentioned.

6.3.1 INITIALISED State

This state is also described in [NFCT2T]. The formatting procedure to prepare the MIFARE Ultralight Family tag in this state is described in [section 6.5.1](#).

The INITIALISED state SHALL be identified by:

- all lock bits set to 0b, and
- the empty 1st NDEF Message TLV.

6.3.2 READ/WRITE State

This state is also described in [NFCT2T]. The formatting procedure to prepare the MIFARE Ultralight Family tag in this state is described in [section 6.5.2](#).

The READ/WRITE state SHALL be identified by:

- all lock bits set to 0b, and
- the non-empty 1st NDEF Message TLV.

6.3.3 READ-ONLY State

This state is also described in [NFCT2T]. The formatting procedure to prepare the MIFARE Ultralight Family tag in this state is described in [section 6.5.3](#).

The READ/ONLY state SHALL be identified by:

- all lock bits set to 1b, and
- the non-empty 1st NDEF Message TLV.

6.3.4 Ultralight READ/WRITE State

The Ultralight READ/WRITE state is a special case of the READ/WRITE one. Particular settings of the lock bits identify this state.

The Ultralight READ/WRITE state SHALL be identified by:

- all lock bits that identifies the pages where the CC and the 1st NDEF Message TLV are memorized, set to 0b,
- at least one lock bit that identifies the pages where the CC and 1st NDEF Message TLV are not memorized, set to 1b,
- all block-locking bits set to 0b, and
- the non-empty 1st NDEF Message TLV.

A MIFARE Ultralight Family tag in Ultralight READ/WRITE state MAY have pages that are read-only. For example when one or more lock bits are set to 1b.

An NFC Forum device that implements only the [NFCT2T] specification detects in READ/WRITE state a MIFARE Ultralight Family tag in Ultralight READ/WRITE state (see [Table 7](#)).

The Ultralight READ/WRITE state MAY be used to protect the Lock Control TLVs, and the Memory Control TLVs making the relative pages read-only, but keeping the remaining data area read/write to store NDEF Message TLVs, Proprietary TLVs and Terminator TLV.

6.3.5 Ultralight BLOCKED READ/WRITE State

The Ultralight BLOCKED READ/WRITE state is a special case of the READ/WRITE state.

The Ultralight BLOCKED READ/WRITE state SHALL be identified by:

- all lock bits that identifies the pages where the CC and the 1st NDEF Message TLV are memorized, set to 0b,
- all block-locking bits set to 1b, and
- the non-empty 1st NDEF Message TLV.

An NFC Forum device that implements only the [NFCT2T] specification detects READ/WRITE state a MIFARE Ultralight Family tag in Ultralight BLOCKED READ/WRITE state (see [Table 7](#)).

The Ultralight BLOCKED READ/WRITE state MAY be used to exploit the feature on the Ultralight READ/WRITE state but blocking the tag access configuration to be further modified.

6.3.6 Ultralight READ-ONLY State

The Ultralight READ-ONLY state is a special case of the READ-ONLY one. Particular settings of the lock bits (i.e. different from 1b) identify this state. In this state the NFC Forum device MAY change the state of the MIFARE Ultralight Family tag to READ-ONLY state (i.e. setting all lock bits to 1b).

The Ultralight READ-ONLY state SHALL be identified by:

- all lock bits that identifies the memory pages where the CC and the 1st NDEF Message TLV are memorized, set to 1b,
- at least one lock bit that identifies the pages where the CC and 1st NDEF Message TLV are not memorized, set to 0b,
- all block-locking bits set to 0b, and
- the non-empty 1st NDEF Message TLV.

A MIFARE Ultralight Family tag in Ultralight READ-ONLY state MAY have memory pages that are read/write.

An NFC Forum device that implements only the [NFCT2T] specification detects in READ-ONLY state a MIFARE Ultralight Family tag in Ultralight READ-ONLY state (see [Table 7](#)).

The Ultralight READ-ONLY state MAY be used to protect the Lock Control TLVs, the Memory Control TLVs and the 1st NDEF Message TLV making the relative pages read-only, but keeping the remaining data area read/write to store additional NDEF Message TLVs, Proprietary TLVs, NULL TLVs, and Terminator TLV.

6.3.7 Ultralight BLOCKED READ-ONLY State

The Ultralight BLOCKED READ-ONLY state is a special case of the READ-ONLY state. In this state the access configuration of the complete memory of the MIFARE Ultralight Family tag is blocked.

The Ultralight READ-ONLY state SHALL be identified by:

- all lock bits that identifies the memory pages where the CC and the 1st NDEF Message TLV are memorized, set to 1b,
- all block-locking bits set to 1b, and
- the non-empty 1st NDEF Message TLV.

An NFC Forum device that implements only the [NFCT2T] technical specification, detects in READ-ONLY state a MIFARE Ultralight Family tag in Ultralight BLOCKED READ-ONLY state (see [Table 7](#)).

The Ultralight BLOCKED READ-ONLY state MAY be used to exploit the feature on the Ultralight READ-ONLY state but blocking the tag access configuration to be further modified.

6.4 State changes/Transitions

This section describes the possible state changes of the MIFARE Ultralight Family tag. [Fig 15](#) shows the states and the state changes (also called transitions) between them.

A transition SHALL be allowed when the MIFARE Ultralight Family tag is in a valid state as defined in [section 6.3](#).

In the transitions below the CC or OTP Byte 3 SHALL be set by the NFC Forum device according to the values defined in [Table 8](#) of the final state.

6.4.1 Transition from READ/WRITE to INITIALISED

This transition SHOULD NOT be implemented. It is described in this document only for sake of completeness.

To perform this transition the NFC Forum device SHALL:

1. Detect the 1st (mandatory) NDEF Message TLV stored in the data area of the MIFARE Ultralight Family tag, using the NDEF detection procedure (see [NFCT2T]). If no NDEF Message TLV is detected the MIFARE Ultralight Family tag is not in READ/WRITE state and the transition operations SHALL be stopped.
2. Replace the 1st NDEF Message TLV with an empty NDEF Message TLV (i.e. L field equal to 00h, and no V field) using the WRITE and if needed the SECTOR SELECT command (see [chapter 5](#) and [NFCT2T] for details concerning the WRITE and SECTOR SELECT commands).
3. Write the Terminator TLV in the first byte after the first NDEF Message TLV using the WRITE command.

The transition invalidates but does not clear the data written after the Terminator TLV that was present before. To clear the data after the Terminator TLV a sequence of WRITE commands and if needed SECTOR SELECT command SHALL be used.

6.4.2 Transitions from READ/WRITE to Ultralight READ/WRITE

To perform the transition from READ/WRITE to Ultralight READ/WRITE the NFC Forum device SHALL:

- set to 1b at least one of the lock bits related to pages that do NOT contain the CC and the 1st NDEF Message TLV,
- keep the lock bits related to the pages that contain the CC and the 1st NDEF Message TLV equal to 0b, and
- keep the block-locking bits equal to 0b.

6.4.3 Transitions from Ultralight READ/WRITE to Ultralight BLOCKED READ/WRITE

To perform the transition from Ultralight READ/WRITE to Ultralight BLOCKED READ/WRITE the NFC Forum device SHALL:

- keep the lock bits related to the pages that contain the CC and the 1st NDEF Message TLV equal to 0b, and
- set all the block-locking bits equal to 1b.

6.4.4 Transitions from READ/WRITE to Ultralight BLOCKED READ/WRITE

To perform the transition from READ/WRITE to Ultralight BLOCKED READ/WRITE the NFC Forum device SHALL:

- keep the lock bits related to the pages that contain the CC and the 1st NDEF Message TLV equal to 0b, and
- set all the block-locking bits equal to 1b.

6.4.5 Transitions from READ/WRITE to Ultralight BLOCKED READ-ONLY

To perform the transition from READ/WRITE to Ultralight BLOCKED READ-ONLY the NFC Forum device SHALL:

- set all the lock bits related to the pages that contain the CC and the 1st NDEF Message TLV equal to 1b, and
- set all the block-locking bits equal to 1b.

Before setting the lock bits the NFC Forum device SHALL set to 0Fh the Byte 3 of the CC.

6.4.6 Transition from INITIALISED to Ultralight READ/WRITE

To perform the transition from INITIALISED to Ultralight READ/WRITE the NFC Forum device SHALL combine in the following order:

- the transition from INITIALISED to READ/WRITE defined by the NFC Forum (see [NFCT2T]), and
- the transition from READ/WRITE to Ultralight READ/WRITE (see [section 6.4.2](#)).

6.4.7 Transitions from Ultralight READ/WRITE to Ultralight READ-ONLY

To perform the transition from Ultralight READ/WRITE to Ultralight READ-ONLY the NFC Forum device SHALL:

- set to 1b all the lock bits related to the pages that contain the CC and the 1st NDEF Message TLV, and
- keep the block-locking bits equal to 0b.

Before setting the lock bits the NFC Forum device SHALL set to 0Fh the Byte 3 of the CC.

6.4.8 Transitions from Ultralight READ/WRITE to READ-ONLY

To perform the transition from INITIALISED to Ultralight READ/WRITE the NFC Forum device SHALL:

- set all lock bits (including the block-locking bits) to 1b.

Before setting the lock bits the NFC Forum device SHALL set to 0Fh the Byte 3 of the CC.

6.4.9 Transitions from READ/WRITE to Ultralight READ-ONLY

To perform the transition from READ/WRITE to Ultralight READ-ONLY the NFC Forum device SHALL combine in the following order:

- the transition from READ/WRITE to Ultralight READ/WRITE (see [section 6.4.2](#)), and
- the transition from Ultralight READ/WRITE to Ultralight READ-ONLY (see [section 6.4.7](#)).

6.4.10 Transitions from Ultralight READ-ONLY to READ-ONLY

To perform the transition from Ultralight READ-ONLY to READ-ONLY the NFC Forum device SHALL:

- set all lock bits (including the block-locking bits) to 1b.

6.4.11 Transitions from Ultralight READ-ONLY to Ultralight BLOCKED READ-ONLY

To perform the transition from Ultralight READ-ONLY to Ultralight BLOCKED READ-ONLY the NFC Forum device SHALL:

- keep the lock bits related to the pages that contain the CC and the 1st NDEF Message TLV equal to 1b, and
- setting all block-locking bits to 1b.

6.5 Formatting Procedures

The formatting procedures for MIFARE Ultralight Family tag are a sequence of READ and WRITE commands.

In the formatting procedures the OTP area of the MIFARE Ultralight Family tag (see [chapter 2](#)) is used to store the CC bytes. The mandatory (i.e. 1st) NDEF Message TLV and the Terminator TLV are written in the data area (see [NFCT2T]).

Byte 0, byte 1 and byte 2 of the OTP area (or CC area using the NFC Forum terminology, see [NFCT2T]) SHALL NOT be modified once they have been written by the formatting procedure. In particular being the version number (CC byte 1) fixed once written, it means that it is not possible to update a MIFARE Ultralight Family tag to a new future version of the NFC Forum Type 2 Tag.

6.5.1 INITIALISED Formatting Procedure

The NFC Forum device SHOULD use the INITIALISED formatting procedure to prepare the tag to store NFC Forum defined data (e.g. NDEF message) in INITIALISED state (see [section 6.3.1](#)). It is assumed that the MIFARE Ultralight Family tag is configured to allow the INITIALISED formatting procedure i.e. MIFARE Ultralight Family tag is a blank card (see [chapter 2](#)).

The INITIALISED formatting procedure writes inside the MIFARE Ultralight Family tag the empty NDEF Message TLV and the Terminator TLV.

In case of MIFARE Ultralight tag, the INITIALISED formatting procedure does not write any Lock Control TLV, or Memory Control TLV.

Concerning MIFARE Ultralight Family product that is not a MIFARE Ultralight, the INITIALISED formatting procedure writes one Lock Control TLV to indicate the dynamic lock bits that are located starting from the byte just after the data area (see [chapter 2](#) and [Fig 4](#)). The Lock Control TLV is written before any NDEF Message TLV. Between the Lock Control TLV and the 1st NDEF Message TLV zero, one or more NULL TLV MAY be written by the NFC Forum device. The NULL TLVs are used to be able to lock only the Lock Control TLV setting the relative lock bit(s) and to avoid the partially locking of the 1st NDEF Message TLV. It is RECOMMENDED to write the Lock Control TLV in separated read-only pages from the others TLV blocks filling up the pages with NULL TLVs.

In case of MIFARE Ultralight C, it is STRONGLY RECOMMENDED to diversify the the 3DES authentication key.

Before performing the INITIALISED formatting procedure, the NFC Forum device SHALL use the card identification procedure described in [section 2.5.1](#) to detect that the tag is a MIFARE Ultralight blank card, and if it is the case to retrieve the Version Information summarized in [Table 4](#) that is needed by the INITIALISED formatting procedure.

Below the INITIALISED formatting procedure is shown (see [chapter 5](#) for command details):

1. Send WRITE command to set the CC-OTP page 3 in the following way:
 - a. byte 0 equal to E1h i.e. the magic number,
 - b. byte 1 equal to the version of the Type 2 Tag Operation specification (e.g. in case of [NFCT2T] the byte is equal to 10h) that is applied to format the MIFARE Ultralight Family tag,
 - c. byte 2 equal to the data area size divided by 8, the data area size is calculated during the card identification procedure (see [section 2.5.1](#)):
 - in case of MIFARE Ultralight byte 2 is equal to 06h,
 - in case of MIFARE Ultralight C byte 2 is equal to 12h, and
 - in case of MIFARE Ultralight Family product (see [section 2.2.2.1](#)),

$$\text{byte2} = (\text{AdditionalDataAreaSize} + 48) / 8$$
 - d. byte 3 is set to 00h, i.e. read and write access.
2. In case of MIFARE Ultralight Family product that is not a MIFARE Ultralight, send one or more WRITE commands to write the Lock Control TLV. The fields of the Lock Control TLV SHALL be set as following:
 - a. The *Position* field:
 - *PageAddr*, see algorithm in [section 6.5.1.1](#), and
 - *ByteOffset*, see algorithm in [section 6.5.1.1](#).
 - b. The *Size* field is equal to *#ofDynamicLockBits*, see [section 2.2.2.1 and Table 4](#).
 - c. The *Page control* field:
 - *BytesPerPage* see algorithm in [section 6.5.1.1](#), and
 - *BytesLockedPerLockBit* is equal to ([section 2.2.2.1 and Table 4](#))

$$\log_2(L\text{ChunkPerDynamicLockBit} \cdot L\text{ChunkSize}).$$
3. Send one or more WRITE commands to write an empty NDEF Message TLV, the Terminator TLV, and in case NULL TLVs in the data area with:

- a. The empty NDEF Message TLV equal to 0300h, and
- b. The Terminator TLV equal to FEh,

6.5.1.1 Algorithm to Calculate ByteOffset, BytePerPage and PageAddr

The algorithm below is written in a pseudo-code and it calculates the parameters *ByteOffset*, *BytePerPage* and *PageAddr* of the Lock Control TLV from the address in bytes (called *Addr*) where the first byte of the dynamic lock bits is located i.e. the first byte after the data area (see [chapter 2](#)).

It is supposed that the first byte of the tag memory (i.e. Byte 0 of Page 0, see [Fig 4](#)) has the address equal to 0, and the address of the first byte of the dynamic lock bits is $Addr = 64 + AdditionalDataAreaSize$.

```

1  // this function converts an address Addr to the three parameters of the
2  // Lock Control TLV
3  // - PageAddr
4  // - BytePerPage
5  // - ByteOffset
6  Function Pippo(Float Addr)
7      Float PageAddr
8      Float BytePerPage
9      Float ByteOffset
10     Float a
11     Float b
12
13     If (Addr == 0) Then
14         PageAddr = 0
15         BytePerPage = 0
16         ByteOffset = 0
17     Else
18         For (i = 15; i>0; i--)
19             // Round to the first integer up
20             a = RoundUp(Addr / i)
21             b = Log2(a)
22             // Round to the first integer down
23             BytePerPage = RoundDown(b)
24
25             ByteOffset = Addr - i * 2 ^ BytePerPage
26             If ((ByteOffset > -1) && (ByteOffset < 16)) Then
27                 PageAddr = i
28                 i = 1
29             End_If
30         End_For
31     End_If
32
33 // Check that ByteOffset, BytePerPage and PageAddr are in the range of 0-15
34 If ((ByteOffset > -1) && (ByteOffset < 16)) &&
35 ((BytePerPage > -1) && (BytePerPage < 16)) &&
36 ((PageAddr > -1) && (PageAddr < 16)) Then
37     // ByteOffset, BytePerPage and PageAddr are in the range of 0-15
38     Print(PageAddr, BytePerPage, ByteOffset)

```

```
39         Exit()
40     End_If
41
42     // ByteOffset, BytePerPage or PageAddr are not in the range of 0-15 (4 bits)
43     PageAddr = 16
44     BytePerPage = 16
45     ByteOffset = 16
46     Print(PageAddr, BytePerPage, ByteOffset)
47
48 End Function
```

6.5.2 READ/WRITE Formatting Procedure

The READ/WRITE formatting procedure is a combination of two procedures listed below:

1. the INITIALISED formatting procedure (see [section 6.5.1](#)), and
2. the transition from INITIALISED to READ/WRITE (see [NFCT2T]).

The previous list also indicates in which order the NFC Forum device SHALL execute the procedures.

6.5.3 READ-ONLY Formatting Procedure

The READ/WRITE formatting procedure is a combination of three procedures:

1. the INITIALISED formatting procedure (see [section 6.5.1](#)),
2. the transition from INITIALISED to READ/WRITE (see [NFCT2T]), and
3. the transition from READ/WRITE to READ-ONLY (see [NFCT2T]).

The previous list also indicates in which order the NFC Forum device SHALL execute the procedures.

7. Additional Features

This chapter describes the additional features that the MIFARE Ultralight Family product MAY support. Even implementing these features the MIFARE Ultralight Family product formatted as Type 2 Tag platform SHALL remain compatible with the technical specification [NFCT2T].

7.1 Ultralight READ/WRITE, and Ultralight BLOCKED READ/WRITE

The Ultralight READ/WRITE state is used to block (i.e. make read-only) specific pages of the MIFARE Ultralight Family tag.

For instance the pages related to Lock Control TLV and Memory Control TLV can be locked to avoid future changes. To store Lock Control TLV and Memory Control TLV into pages separated from other TLV blocks like the NDEF Message TLV, the NULL TLV MAY be used to completely fill the Lock Control and Memory Control TLV pages.

The Ultralight BLOCKED READ/WRITE state is used to lock the access configuration of the MIFARE Ultralight tag keeping the write access for specific pages. An example where this state can be used is to avoid malicious or accidental change of the access configuration i.e. setting to 1b the block-locking bits.

7.2 Ultralight READ-ONLY, and Ultralight BLOCKED READ-ONLY

The Ultralight READ-ONLY state is used to keep non-blocked (i.e. readable and writeable) specific pages of the MIFARE Ultralight Family tag.

For instance the pages related to Lock Control TLV and Memory Control TLV can be locked to avoid future changes. To store Lock Control TLV and Memory Control TLV into pages separated from other TLV blocks like the NDEF Message TLV, the NULL TLV MAY be used to completely fill the Lock Control and Memory Control TLV pages.

The Ultralight BLOCKED READ-ONLY state is used to lock the access configuration of the MIFARE Ultralight Family tag keeping the write access for specific pages. An example where this state can be used is to avoid malicious or accidental change of the access configuration i.e. setting to 1b the block-locking bits.

7.3 Storing of Additional non-TLV Structured Data

Additional non-TLV structured data MAY be stored inside a MIFARE Ultralight Family tag in two different ways additional to the Proprietary TLV or the NDEF Message TLV (see also [NFCT2T]):

- Writing the data after the Terminator TLV (see [section 7.3.1](#)), or
- Writing the data into data areas specified using the Memory Control TLV (see [section 7.3.2](#)) or the Lock Control TLV.

The information of the existence of this data is application specific. A possible solution to indicate the presence of the data MAY be storing this information inside the tag using one record of the NDEF Message (see [NDEF]).

7.3.1 Writing Data after the Terminator TLV

[Fig 16](#) provides an example about how to write data after the Terminator TLV.

In this example two data areas called Any Data 1 and Any Data 2 are stored after the Terminator TLV. The information about the position and/or size of Any Data 1 and Any

Data 2 inside the MIFARE Ultralight Family tag MAY be stored in the first and second record of the NDEF Message encapsulated in the NDEF Message TLV.

The memory area containing the Unique Identifier (UID), lock bytes, reserved bytes, internal bytes, and any Lock Control TLV, or Message Control TLV is indicated with the block called Configuration Data.

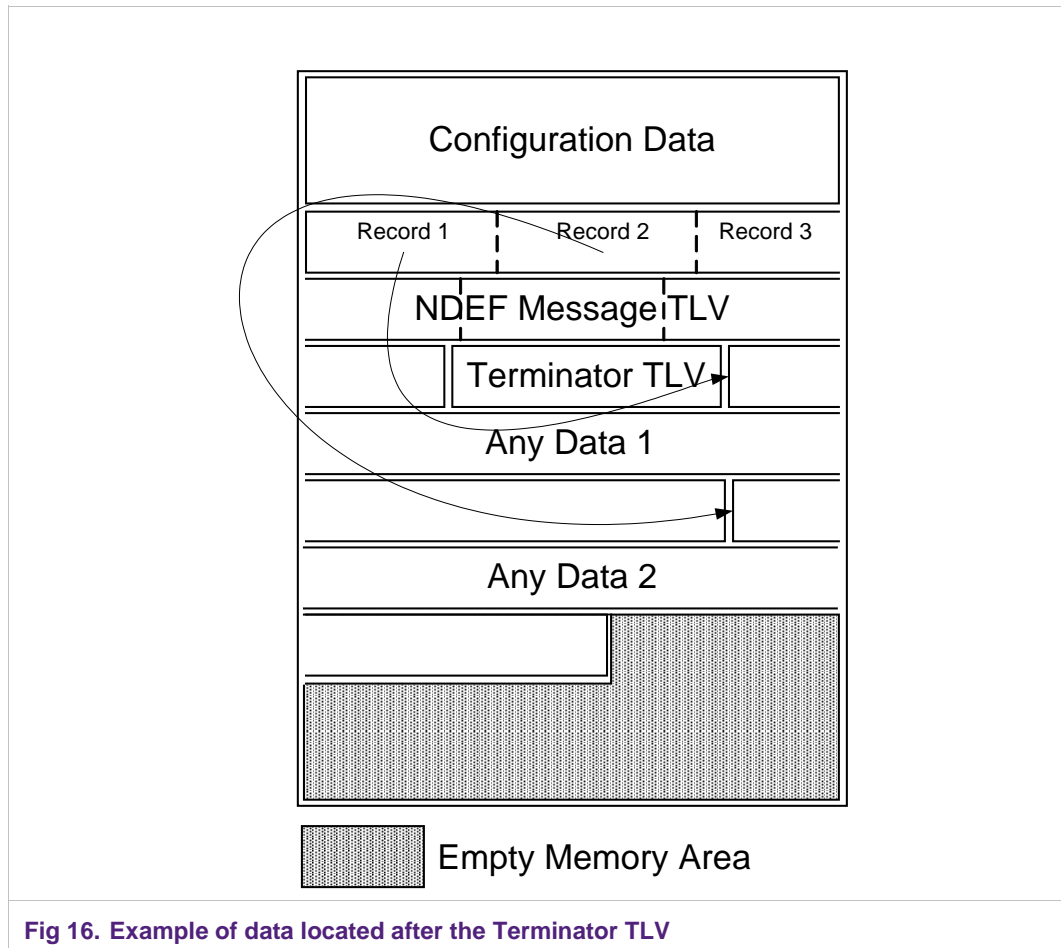


Fig 16. Example of data located after the Terminator TLV

7.3.2 Writing Data into Areas Specified Using the Memory Control TLV

The description of writing data (called Any Data) using the Memory Control TLV is given by means of an example in [Fig 17](#).

In this example the data is stored in the reserved area identified by the Memory Control TLV. Extra information about position and size of Any Data inside the reserved area are stored inside the NDEF Message encapsulated in the NDEF Message TLV. This solution is more flexible than the previous one in the sense that the data inside the reserved area MAY be located almost everywhere in the MIFARE Ultralight Family tag, and also between an NDEF Message TLV. However as specified in [NFCT2T] reserved areas SHALL NOT be present before page 15 i.e. the first 64 bytes of the tag.

Comparing this method of storing data with the method explained in [section 7.3.1](#), storing data in reserved memory areas has the advantage that the data itself is not corrupted if a bigger NDEF Message TLV is written.

The memory area containing the Unique Identifier (UID), lock bytes, reserved bytes, internal bytes, and any Lock Control TLV, or Message Control TLV in indicated with the block called Configuration Data.

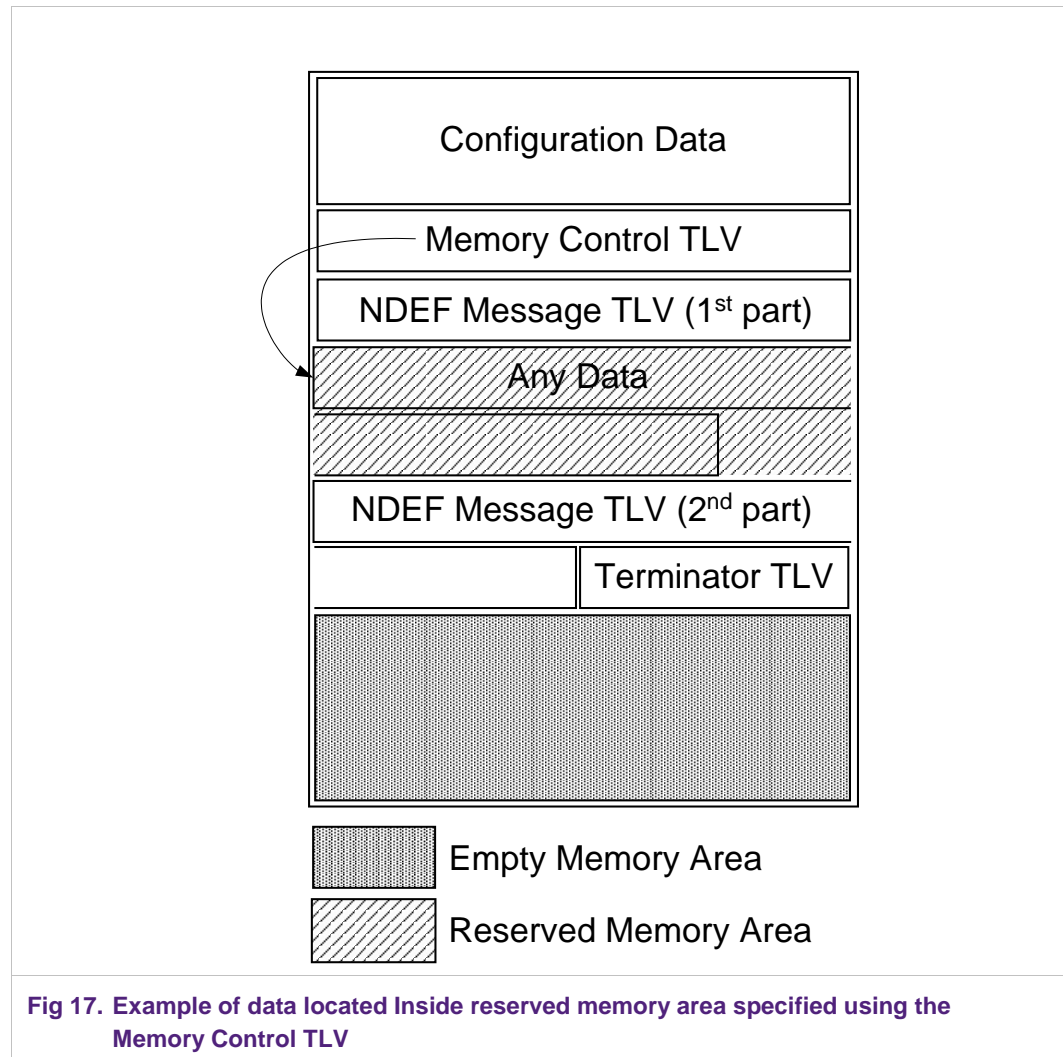


Fig 17. Example of data located inside reserved memory area specified using the Memory Control TLV

8. ANNEX: Example of a Type 2 Tag – MIFARE Ultralight C in INITIALISED state

The example below in [Fig 18](#) shows a MIFARE Ultralight C formatted as Type 2 Tag in INITIALISED state (see [NFCT2T]). In particular:

- the static lock bytes Lock0 and Lock1 are both set to 00h,
- the dynamic lock bytes Lock2 and Lock3 are both set to 00h,
- the OTP bytes are set as the Capability Container in [NFCT2T]:
 - Byte 0 is equal to E1h (magic number),
 - Byte 1 is the NFC Forum Type 2 Tag Operational version number 10h,
 - Byte 2 is set to 12h i.e. 144 bytes of data area, and
 - Byte 3 is set to 00h i.e. read and write access capability without any security.
- the Lock Control TLV on page 4 and 5:
 - *Tag* equal to 01h,
 - *Length* equal to 03h, and
 - *Value* equal to A01044h i.e. *PagesAddr* equal to Ah, *ByteOffset* equal to 0h, *Size* (number of dynamic lock bits) equal to 10h, *BytesPerPage* equal to 4h and *BytesLockedPerLockBit* equal to 4h.
- NDEF Message TLV on page 5: *Tag* equal to 03h and *Length* equal to 00h.
- Terminator TLV on page 5: *Tag* equal to FEh.

Byte Number	0	1	2	3	Page
UID / Internal	UID0	UID1	UID2	Internal0	0
Serial Number	UID3	UID4	UID5	UID6	1
Internal / Lock	Internal1	Internal2	Lock0=00h	Lock1=00h	2
OTP	OTP0=E1h	OTP1=10h	OTP2=12h	OTP3=00h	3
Data	LockControlTLV0 = 01h	LockControlTLV1 = 03h	LockControlTLV2 = A0h	LockControlTLV3 = 10h	4
Data	LockControlTLV4 = 44h	NDEFMessage TLV0 = 03h	NDEFMessage TLV1 = 00h	TerminatorTLV0 = FEh	5
Data	Data8	Data9	6
Data
Data	15
Data	16
Data
Data	39
DynLock/Reserved	Lock2=00h	Lock3=00h	Reserved	..	40
Reserved
Reserved	47

Fig 18. Type 2 Tag – MIFARE Ultralight C in INITIALISED state

9. ANNEX: Examples

In the examples below each command and response are written in hexadecimal format. The top-left byte of each command and response is sent first. CRC0 and CRC1 indicate the CRC bytes. BCC1 and Internal are reserved bytes.

For more information about command and response formatting see [MFUL].

9.1 Example of INITIALISED Formatting Procedure

This example shows how the INITIALISED Formatting Procedure (see [section 6.5.1](#)) may be implemented for a MIFARE Ultralight tag.

The example of INITIALISED Formatting Procedure is described below

1. READ command with page address = 02h to check if the lock bits are equal to 0h, and the CC-OTP is set to byte 0 = E1h, byte1 =10h, byte 2 bigger than or equal to 06h, and byte 3 equal to 00h.
 - a. Command: 30 02 CRC0 CRC1
 - b. Expected Response: BCC1 Internal 00 00 E1 10 06 00 00 00 00 00 00 00 00 00 00 CRC0 CRC1
2. WRITE command with page address = 03h to set the CC to byte 0 = E1h, byte 1 = 10h, byte 2 = 06h, byte 3 = 00h.
 - a. Command: A2 03 E1 10 06 00 CRC0 CRC1
 - b. Expected Response: Ah (acknowledge 4 bits)
3. WRITE command with page address = 04h to write an empty NDEF Message TLV and the Terminator TLV.
 - a. Command: A2 04 03 00 FE 00 CRC0 CRC1
 - b. Expected Response: Ah (acknowledge 4 bits)

9.2 Example of Writing an NDEF Message setting the tag in Ultralight BLOCKED READ/WRITE

This example shows how to set a MIFARE Ultralight tag from the INITIALISED state to the Ultralight BLOCKED READ/WRITE state. It is a combination of:

- Transition from INITIALISED to READ/WRITE (see [NFCT2T]),
- transition from READ/WRITE to Ultralight READ/WRITE (see [section 6.4.2](#)), and
- transition from Ultralight READ/WRITE to Ultralight BLOCKED READ/WRITE (see [section 6.4.3](#)).

The NDEF Message TLV occupies 2 pages (page 04h and 05h). The page 05h is filled with NULL TLVs to have only the NDEF Message TLV in page 04h and 05h. In this way the Terminator TLV is located on byte 0 of page 06h. The lock bytes are set to have read-only access to the NDEF Message TLV (i.e. Page 04h, and 05h), but not to the rest of the data area i.e. from page 06h where the Terminator TLV is stored. Moreover the lock bytes have the block-locking bits set to 1b to fix the access configuration of the tag.

As precondition the MIFARE Ultralight tag is in INITIALIZED state. The NDEF message inside the NDEF Message TLV is an empty NDEF message (see [APPENDIX A](#) of [NFCT2T]).

The example is described below:

1. WRITE command with page address 04h to set byte 0 = 03h, byte 1 = 03h, byte 2 = D0h, byte 3 = 00h to write the first part of the NDEF Message TLV.
 - a. Command: A2 04 03 03 D0 00 CRC0 CRC1
 - b. Expected Response: Ah (acknowledge 4 bits)
2. WRITE command with page address 05h to set byte 0 = 00h, byte 1 = 00h, byte 2 = 00h, byte 3 = 00h to write the second part of the NDEF Message TLV, and three NULL TLVs.
 - a. Command: A2 05 00 00 00 00 CRC0 CRC1
 - b. Expected Response: Ah (acknowledge 4 bits)
3. WRITE command with page address 06h to set byte 0 = EFh, byte 1 = 00h, byte 2 = 00h, byte 3 = 00h to write the Terminator TLV.
 - a. Command: A2 05 EF 00 00 00 CRC0 CRC1
 - b. Expected Response: Ah (acknowledge 4 bits)
4. WRITE command with page address 02h to set byte 0 = 00h, byte 1 = 00h, byte 2 = 3Fh, byte 3 = 00h to write the lock bytes equal to 3F00h to lock page 04h, page 05, and the access configuration of the tag.
 - a. Command: A2 05 00 00 3F 00 CRC0 CRC1
 - b. Expected Response: Ah (acknowledge 4 bits)

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 of 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.

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. A license for the patents portfolio of NXP B.V. for the NFC standards needs to be obtained at Via Licensing, the pool agent of the NFC Patent Pool, e-mail: info@vialicensing.com.

10.4 Trademarks

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

MIFARE — is a trademark of NXP B.V.

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

11. Contents

1. Introduction	3	6.4	State changes/Transitions	37
1.1 Implementation Guidelines	3	6.4.1	Transition from READ/WRITE to INITIALISED..	37
1.2 Applicable Documents	3	6.4.2	Transitions from READ/WRITE to Ultralight	
1.3 Convention and notations	4		READ/WRITE	37
1.3.1 Representation of numbers	4	6.4.3	Transitions from Ultralight READ/WRITE to	
1.3.2 Terms and Definition	4		Ultralight BLOCKED READ/WRITE	38
1.4 Special Word Usage	4	6.4.4	Transitions from READ/WRITE to Ultralight	
1.5 Acronyms or Definitions or Glossary	5		BLOCKED READ/WRITE	38
2. Memory Layout	7	6.4.5	Transitions from READ/WRITE to Ultralight	
2.1 MIFARE Ultralight Memory Layout	7		BLOCKED READ-ONLY	38
2.1.1 Static Lock Bytes (Lock0, Lock1) Structure	8	6.4.6	Transition from INITIALISED to Ultralight	
2.1.2 MIFARE Ultralight Blank Card Settings	9		READ/WRITE	38
2.1.2.1 Version Information	10	6.4.7	Transitions from Ultralight READ/WRITE to	
2.2 MIFARE Ultralight Family Memory Layout	10		Ultralight READ-ONLY	38
2.2.1 Dynamic Lock Bytes Structure	12	6.4.8	Transitions from Ultralight READ/WRITE to	
2.2.2 MIFARE Ultralight Family product Blank Card			READ-ONLY	39
Settings	14	6.4.9	Transitions from READ/WRITE to Ultralight	
2.2.2.1 Version Information	15		READ-ONLY	39
2.2.2.2 How to Retrieve the Dynamic Lock Byte		6.4.10	Transitions from Ultralight READ-ONLY to	
Structure from the Version Information	17		READ-ONLY	39
2.3 MIFARE Ultralight C Memory Layout	18	6.4.11	Transitions from Ultralight READ-ONLY to	
2.3.1 Dynamic Lock Bytes (Lock2 and Lock3)			Ultralight BLOCKED READ-ONLY	39
Structure	19	6.5	Formatting Procedures	39
2.3.2 Blank Card Settings	20	6.5.1	INITIALISED Formatting Procedure	39
2.3.3 MIFARE Ultralight C Admin Security Feature...	21	6.5.1.1	Algorithm to Calculate ByteOffset, BytePerPage	
2.4 Mapping of NFC Forum data	22		and PageAddr	41
2.5 Card Identification Procedure	22	6.5.2	READ/WRITE Formatting Procedure	42
2.5.1 Card Identification Procedure for Blank Cards	25	6.5.3	READ-ONLY Formatting Procedure	42
2.5.1.1 Version Treating	25	7. Additional Features	43	
2.5.2 Card Identification Procedure for Cards in a Valid		7.1	Ultralight READ/WRITE, and Ultralight	
State	26		BLOCKED READ/WRITE	43
3. Read/Write Access	28	7.2	Ultralight READ-ONLY, and Ultralight BLOCKED	
4. Framing / Transmission Handling	29		READ-ONLY	43
5. Command Set	30	7.3	Storing of Additional non-TLV Structured Data.	43
6. Life Cycle	31	7.3.1	Writing Data after the Terminator TLV	43
6.1 NFC Forum Life Cycle	31	7.3.2	Writing Data into Areas Specified Using the	
6.2 MIFARE Ultralight Life Cycle	32		Memory Control TLV	44
6.3 States	35	8. ANNEX: Example of a Type 2 Tag – MIFARE	Ultralight C in INITIALISED state	46
6.3.1 INITIALISED State	35	9. ANNEX: Examples	47	
6.3.2 READ/WRITE State	35	9.1	Example of INITIALISED Formatting Procedure	
6.3.3 READ-ONLY State	35		47
6.3.4 Ultralight READ/WRITE State	35	9.2	Example of Writing an NDEF Message setting	
6.3.5 Ultralight BLOCKED READ/WRITE State	36		the tag in Ultralight BLOCKED READ/WRITE ..	47
6.3.6 Ultralight READ-ONLY State	36	10. Legal information	49	
6.3.7 Ultralight BLOCKED READ-ONLY State	37	10.1	Definitions	49

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

10.2 Disclaimers.....49
10.3 Licenses.....49
10.4 Trademarks.....49
11. Contents.....50

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

© NXP B.V. 2011.

All rights reserved.

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

Date of release: 21 April 2011
130314
Document identifier: AN1303