AN11340

MIFARE Ultralight and MIFARE Ultralight EV1 Features and Hints

Rev. 3.1 — 9 July 2018 Application note

Document information

Info	Content
Keywords	Multiple ticketing, secured data storage, implementation hints
Abstract	This document presents features and hints for a secured and optimized application development using MIFARE Ultralight and MIFARE Ultralight EV1 cards.

Revision history

Rev	Date	Description
3.1	20180709	Editorial update
3.0	20130227	Updated examples from DES to AES and added MIFARE Ultralight EV1 updated section 2.1.2 Transaction Speed, added section 2.1.2.1 FAST_READ Time Saving, added section 4 MIFARE Ultralight EV1 Counters, added section 5 MIFARE Ultralight EV1 Password and PACK, added section 6 MIFARE Ultralight EV1 Anti-cloning based on Originality Check, added section 7 MIFARE Ultralight EV1 Tearing Application Implementation
2.1	20070826	Example has been corrected
2.0	20061019	Security features (section 2.2) and example (annex 6.3) added
1.0	20020515	First release

Contact information

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

© NXP B.V. 2018. All rights reserved.

1. Introduction

1.1 Purpose and Scope

This application note is intended to describe the features of the MIFARE Ultralight MF0ICU1 and MIFARE Ultralight EV1 MF0ULx1 (see [MF0ICU1] and [MF0ULx1]), and the comparison to the MIFARE Classic product family.

It addresses some security mechanisms which may be used to protect the data stored in the card and demonstrates the implementation of MIFARE Ultralight and MIFARE Ultralight EV1 into an existing MIFARE Classic application and shows the necessary modification over the existing environment for NFC readers for MIFARE ICs. The changes apply to all relevant NXP Semiconductors NFC readers for MIFARE ICs.

Being MIFARE Ultralight EV1 MF0ULx1 backward compatible with MIFARE Ultralight MF0ICU1, the following chapter are applicable to both MIFARE Ultralight MF0ICU1 and MIFARE Ultralight EV1 MF0ULx1, if not explicitly stated otherwise.

1.2 How to use this document

This document contains a collection of hints and features that could be of interest for those, who plan to use the MIFARE Ultralight and MIFARE Ultralight EV1.

None of this information is intended to replace any of the relevant datasheets or design guides.

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

Any data, value, cryptogram are expressed here as hex string format if not other mentioned.

In this document the term „MIFARE Classic card“ refers to a MIFARE Classic IC-based contactless card, the term „MIFARE Ultralight card“ refers to a MIFARE Ultralight IC-based contactless card.

1.3 Reference documents

[MF0ULx1]	MIFARE Ultralight EV1, Contactless Single-trip Ticket IC MF0ULx1, Product data sheet, Doc. No: 2345**[1]
[AN10922]	AN10922 Symmetric Key Diversifications, Doc. No: 1653**
[AN11093]	AN11093 Card Coil Design Guide, Doc. No.: 0117**
[SI070010]	Temperature Management, Inlet Design.
[MFRC530]	MFRC530; Reader solution for MIFARE products, Doc. No. 0574**
[MFRC531]	MFRC531; Standard ISO/IEC 14443 A/B reader solution
[CLRC632]	CLRC632; Standard multi-protocol reader solution.
[AN11341]	AN11341 MIFARE Ultralight EV1 Originality Signature Validation, Doc. No: 2591**
[ISO/IEC 14443-3]	Identification cards- Contactless integrated circuit(s) cards-proximity cards- Part 3: Initialization and anticollision
[FIPS46-3]	Data Encryption Standard.

[1] ** ... document version number

© NXP B.V. 2018. All rights reserved.

[ISO/IEC 9797-1] Information technology Security techniques Message Authentication Codes.

2. MIFARE Ultralight application hints

2.1 Memory features

In addition to the user memory area the MIFARE Ultralight and MIFARE Ultralight EV1 offers the features of an OTP[2] area and lock bytes to lock the OTP and user area. The usages of LOCK bits are described in [MF0ICU1] and [MF0ULx1].

A MIFARE Ultralight and MIFARE Ultralight EV1 dedicated 4-byte WRITE-command provides a high transaction speed. All the add-on features are dedicated to support special application functionality e.g. ticketing.

2.1.1 Using OTP memory for multiple ticketing

The 4 OTP bytes, which are pre-set to “0” at the delivery, can be set to “1” only once. This gives the possibility to interpret them as an irreversible counter. Therefore, the number of “1” bits in Page 3 can be considered as counter value beside the MIFARE Ultralight EV1 counters (see chapter 4). This counter can be incremented by changing a bit from “0” to”1”. The counter cannot be decremented due to a “1” bit of Page 3 cannot be cleared. In this case this 4-byte offers a number of 32 states that could be used to allow a certain number of passing turn-styles.

Example:

An example of a four rides ticket is shown in figure 1. For this application a ticket issuing the OTP memory of the MIFARE Ultralight has to be pre-set to “FFFFFFF0”.

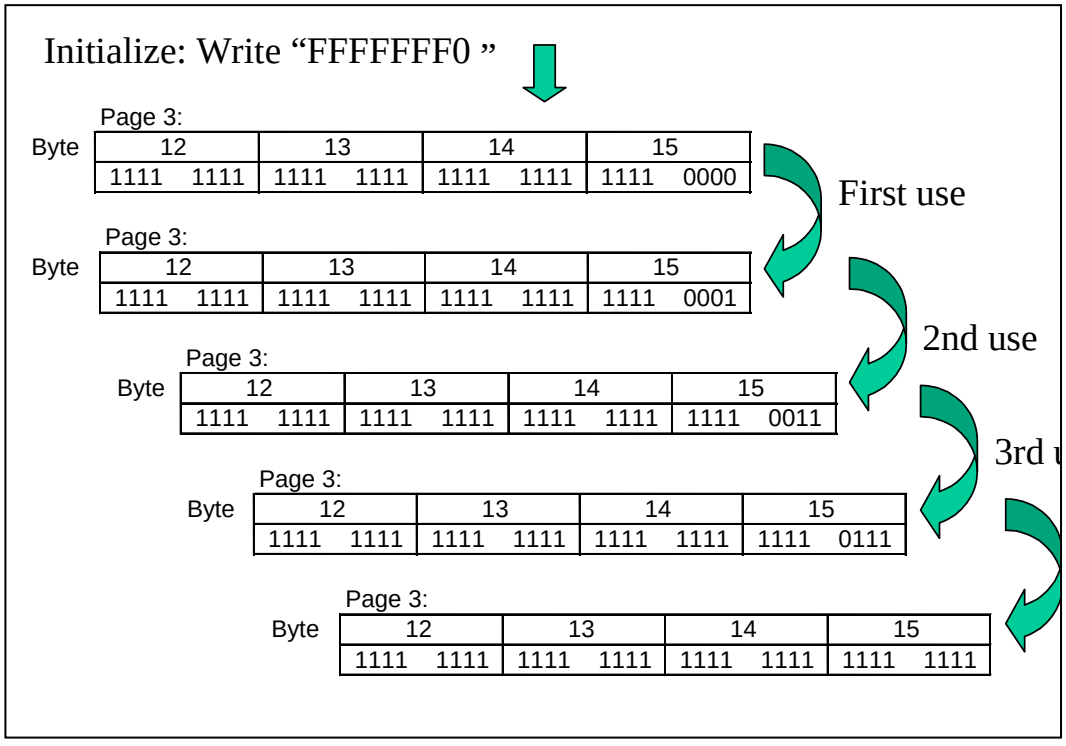


Figure 1: Example of a four rides ticket

[2] **O**ne **T**ime **P**rogramming

© NXP B.V. 2018. All rights reserved.

For every access, the 4-byte OTP (page 3) has to be checked and updated (as shown in Figure 2) to ensure the validity of the tickets. Using the same command flow the counter might be extended to a number of 31 times just by changing the initial memory content.

E.g. “FFFFFC00” or “FC00FC00” might be used for a 10 times counter initial value, or “FFF00000” for a 20 times counter initial value, which has to be written into the OTP memory while issuing the ticket.

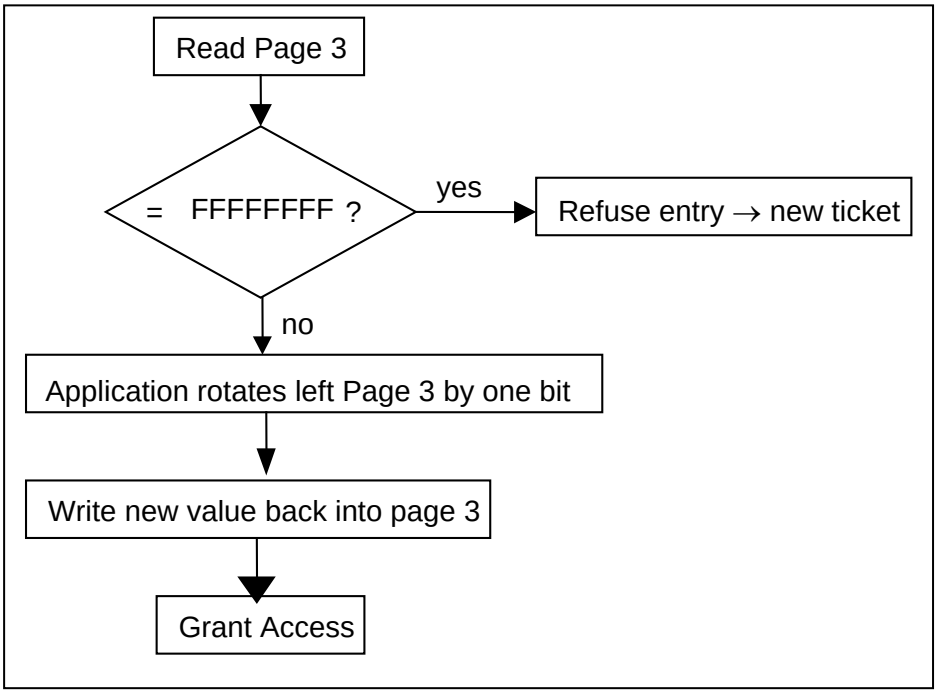


Figure 2: Ticket Counter Command flow

Remark:

With the initial value “00000000” for 32 times counter, the rotate left at the very first access check after selling the ticket has to be exchanged into another initial WRITE command.

© NXP B.V. 2018. All rights reserved.

2.1.2 Transaction Speed

Although the MIFARE Ultralight offers the 16-byte Compatibility Write command to be compatible with the MIFARE Classic environment, the use of the 4 byte Write command is recommended, if the application requires a fast transaction. The Write saves approximately 20% of the transaction time[3] compared to the Compatibility Write, as can be seen in Table 1 and Figure 3.

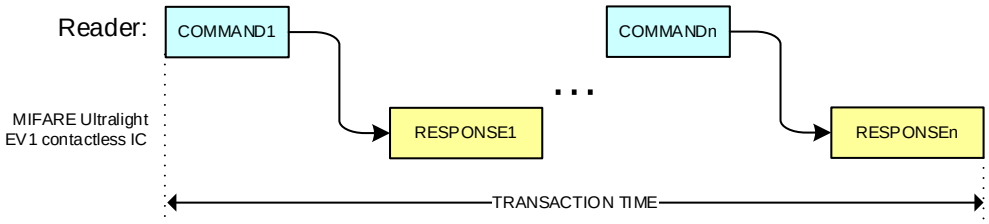


Figure 3: Transaction Time

Command	4 pages		12 pages	
REQA	0.4 ms		0.4 ms	
Anticollision level 1 + 2 & Select	3.7 ms		3.7 ms	
Read	2.1 ms		6.3 ms	
Write	19.7 ms		59.2 ms	
Compatibility Write	25.4 ms		76.4 ms	
Halt	0.5 ms		0.5 ms	
Transaction Time (approximately)	26.4 ms	32.1 ms	70.1 ms	87.3 ms

Table 1: Transaction time

The transaction time[2] as shown in Table 1 and Figure 3 counts the time needed for

- the communication from the reader to the MIFARE Ultralight,
- the response time of the MIFARE Ultralight and
- the communication from the MIFARE Ultralight back to the reader.

2.1.2.1 FAST_READ Time Saving

MIFARE Ultralight EV1 (See [MF0ULx1]) introduces the FAST_READ command. The FAST_READ command has a variable frame length depending on the start and end address parameters. The maximum frame length supported by the PCD needs to be taken into account when issuing this command.

The table below shows the comparison in term of timing between READ and FAST_READ. The FAST_READ command is able to speed up the reading compared with the READ command in case of amount of data that is smaller than 4 pages but also bigger than 4 pages. Only in case of 4 pages reading the READ command is faster than the FAST_READ.

[3] This doesn't include the time requires the host computer for the application itself (e.g. calculating & checking ticket values, displaying results, opening gates, etc.).

© NXP B.V. 2018. All rights reserved.

Command	1 page	4 pages	12 pages	32 pages
READ	2.1 ms	2.1 ms	6.3 ms	16.8 ms
FAST_READ	1.2 ms	3.7 ms	3.7 ms	11.8 ms
FAST_READ Time Saving	+43%	-6%	+21%	+30%

Table 2: READ and FAST_READ Timing Comparison

2.2 Proposed Security Mechanism

MIFARE Ultralight has been designed to support the faster application with the cheapest solution. Therefore, it does not address any security feature except:

- the unique identity (UID),
- the Password protection (MIFARE Ultralight EV1 only, see)
- the Originality Signature Validation (MIFARE Ultralight EV1 only, see [MFULEV1SIGNVA])

From the application point of view this means, no authentication has to be performed and no key is needed. Performing a MIFARE authentication generates an error, and the MIFARE Ultralight goes back to the Idle (or Halt) state.

But if requires, smarter secured application using MIFARE Ultralight can be created using an intelligent reader with the respective application software. A lot of cryptographic technologies are available to assist you. One successfully implemented approach is demonstrated in the following sections. This approach uses a 16-byte Master key (Mk), which is used to add the confidentiality and integrity of the storage data.

MIFARE SAM EV1 (secure access module) can be used in the sub-section below to encrypt or decrypt the data.

2.2.1 Confidentiality of stored Data

As the MIFARE Ultralight pages can be read without any authentication, anyone can read the pages using any standard reader. But if the stored data is encrypted with a secured key then these are just some bytes to one who does not have the secret key and information regarding the encryption method. Therefore, by storing the encrypted data in MIFARE Ultralight memory, one can add the confidentiality to the data itself.

Note that the password verification method available in the MIFARE Ultralight EV1 does not offer a high security protection. It is an easy and convenient way to prevent unauthorized memory access. If a higher level of protection is required, cryptographic methods on application layer can be used to increase overall system security.

As an example, AES may be used for the encryption of data in MIFARE Ultralight memory.

$$Data_{stored} = f\left(key, data_{origin}, UID\right)$$

The 7-byte UID of the MIFARE Ultralight has to be incorporated in the security system to confirm the uniqueness of the tickets and a 16-byte Master Key (Mk) has to be defined by the application provider. To decrease the threat on the Master key (Mk), for each card, a Card key (Ck) is derived from the Master Key (Mk) using the well-known key diversification. The steps to be followed are indicated in [AN10922] section 2.2 “AES-128 key” where the inputs to the 128-bit AES key diversification are:

- M, the diversification input equals to the 7 bytes UID, and

© NXP B.V. 2018. All rights reserved.

- K, the 16 bytes AES 128 bits Master Key (Mk)

And the output is:

- Diversification Key, the 16 bytes AES 128 bits Card Key (Ck)

After this step the plain data is encrypted using the Card Key (Ck) in CBC (send) mode.

- Use 16 bytes initial vector (IV) of all '00's, IV= "00...00"
- As AES 128 works with 16-byte block wise, organize the data in multiple of 16 by adding the standard padding [ISO/IEC 9797-1] with all zeros '00'. As example ('xx' is the data bytes):

```
10 padding bytes: xxxxxxxx xxxx0000 00000000 00000000
15 padding bytes: xx000000 00000000 00000000 00000000
```

The complete scheme is shown in the following figures (figure 3 and figure 4):

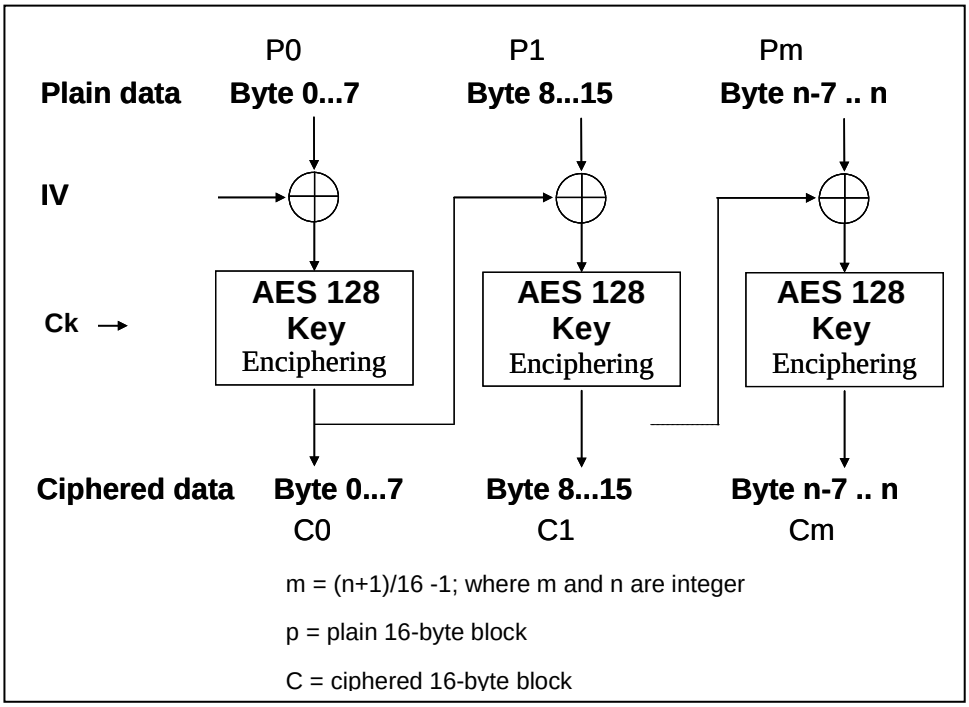


Figure 4: Data encryption scheme

Therefore, the data has to be decrypted after reading it from the card.

© NXP B.V. 2018. All rights reserved.

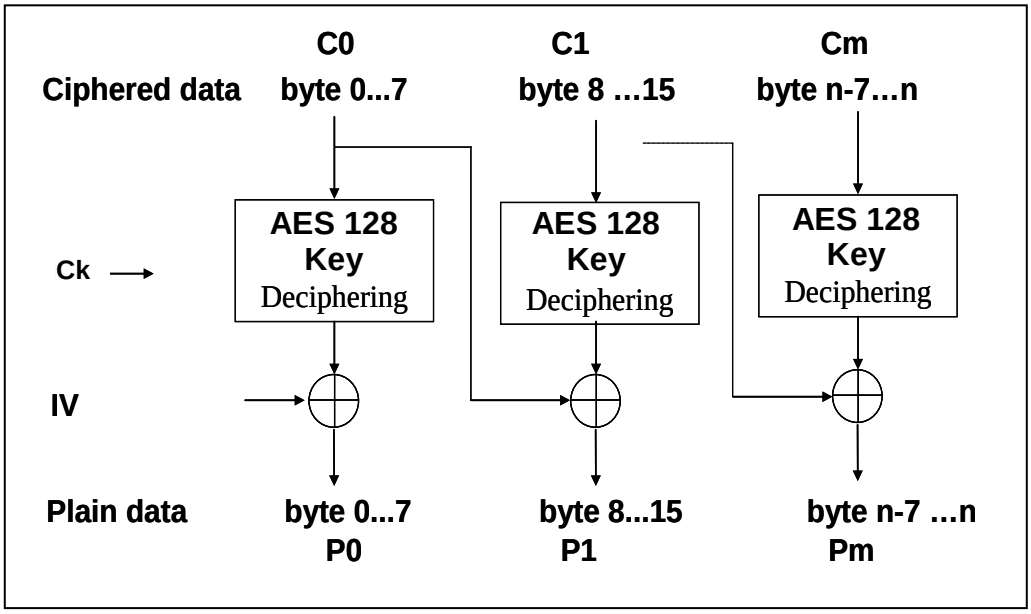


Figure 5: Data decryption scheme

2.2.2 Integrity of stored Data

The content of the MIFARE Ultralight or the MIFARE Ultralight EV1 memory lacks guaranteed integrity. To avoid this inconvenience, we propose a security checksum which has to be calculated over the bytes in pages 2 to used memory end and has to be appended with the data. For this purpose, MAC (Message Authentication Code) [ISO/IEC 9797-1] may be a good choice. The complete scheme is shown in figure 5:

$$Checksum = f\left(key, usedmemory(inc.page2 \,\&\, 3), UID\right)$$

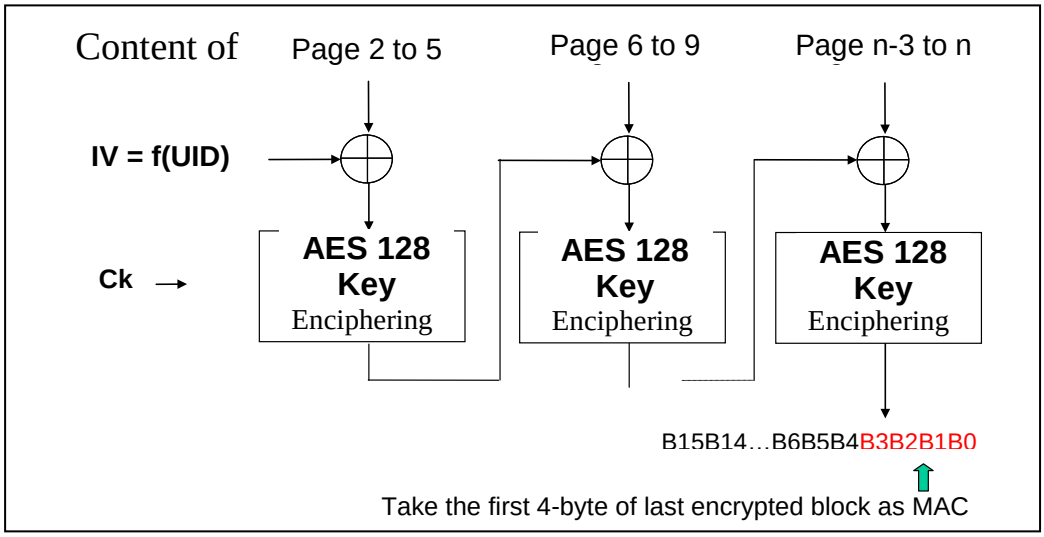


Figure 6: MAC calculation

- Use the same Initial Vector (IV) and Card Key(Ck) used in the previous section. The IV contains the 7 bytes UID that has to be padded to reach 16 bytes. The padding can be done as described in the next item

© NXP B.V. 2018. All rights reserved.

- If number of used pages is odd or number of data bytes is not a multiple of 8-byte, add standard padding with all '00'. As example ('xx' is the data bytes):

```
3 padding bytes: xxxxxxxx xx000000 00000000 00000000
6 padding bytes: xxxx0000 00000000 00000000 00000000
```

Either the encrypted data and/or the checksum can be stored in the MIFARE Ultralight user memory. In this case **the data is protected against damage and being copied** from one MIFARE Ultralight to another one, as long as the key is kept secret.[4]

If high level security features are required, other members of the MIFARE card IC family can be used in the application, e.g. the MIFARE Plus or the MIFARE DESFire EV1.

Please note, the DES operation may be performed using a MIFARE DESFire SAM module. This SAM module will facilitate the system in the following ways:

- The key can be stored once
- The key cannot be read back
- The module provides one step functions for calculation of AES
- The Cryptographic operations are fast enough for real time operations.

3. Using MIFARE Ultralight in an existing MIFARE Classic application

The MIFARE Ultralight offers the feature to be used in an existing MIFARE Classic application. Therefore, the MIFARE Ultralight command structure is compatible to the MIFARE Classic one.

Because the MIFARE Ultralight addresses a different application category (single trip ticketing, fast and cheap transactions), some application changes have to be done anyway, but the existing MIFARE transaction command structure and the NFC reader for MIFARE ICs may be used with the MIFARE Ultralight (as shown in Figure 7).

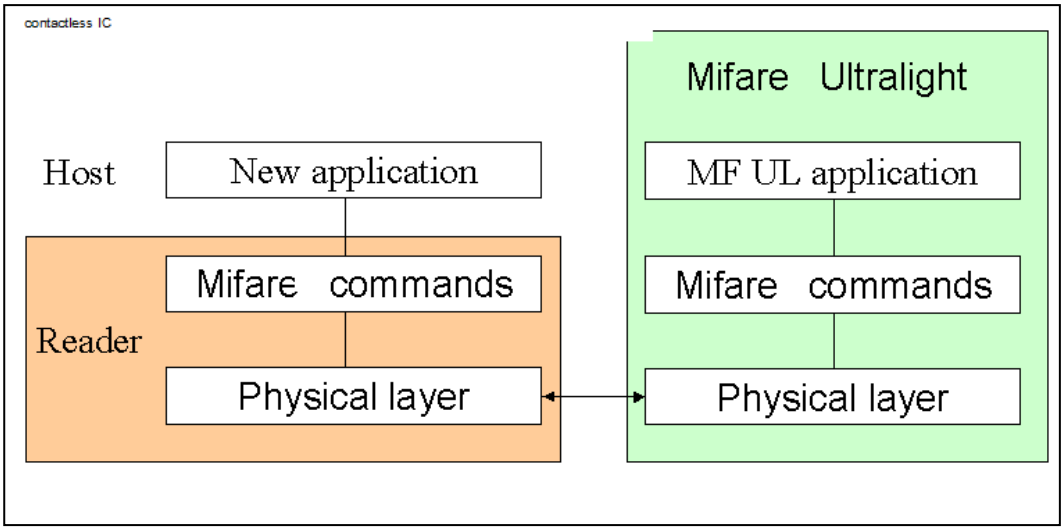


Figure 7: MIFARE Ultralight in an existing MIFARE Classic application

[4] However, a complete security against replay attack can not be provided with the MIFARE Ultralight alone, here also an appropriate application software has to ensure reply attack resistance.

© NXP B.V. 2018. All rights reserved.

Differences: MIFARE Classic - MIFARE Ultralight

The basic differences between MIFARE Classic and MIFARE Ultralight are:

I) The MIFARE Ultralight uses a **7-byte UID** instead of 4-byte (like MIFARE Classic). There are 2 possibilities to select a MIFARE Ultralight card. It suggests using the anti-collision cascade level 2 (as specified in the ISO14443A - 3) to get the complete UID and select one MIFARE Ultralight (see 3.1.1).

If the reader does not support the ISO cascade level 2 anti-collision and there is no chance to update the reader to do so, a combination of anti-collision cascade level 1 (using the first 3 bytes of the UID) and a read of block 0 after the Select can be used instead. Please note that this workaround has the limitation that there is no chance to fully RESOLVE a collision between two cards in case of the unlikely event, that the first part of the UID is equal. The collision can only be DETECTED, allowing the reader to inform the user to present only one card to the reader (see 3.1.2).

II) The MIFARE Ultralight doesn't need the authentication and no keys, as it uses **no encryption**.

Note that MIFARE Ultralight EV1 has the password authentication feature anyhow different from MIFARE Classic authentication.

III) The MIFARE Ultralight only uses **"Read", "Write" and "C.Write"** (MIFARE Classic compatible write command with 16 Bytes).

Note that MIFARE Ultralight EV1 is backward compatible with MIFARE Ultralight supporting additional command set.

No Value-commands are used.

© NXP B.V. 2018. All rights reserved.

3.1 Transaction Command Flows

3.1.1 Transaction flow using Cascade Level 2

The anti-collision cascade level 1 and 2 (as described in the ISO14443-A part 3) should be used to select a MIFARE Ultralight. This command sequence gives back the complete 7-byte UID of the MIFARE Ultralight, and allows selecting only one card (as shown in Figure 8).

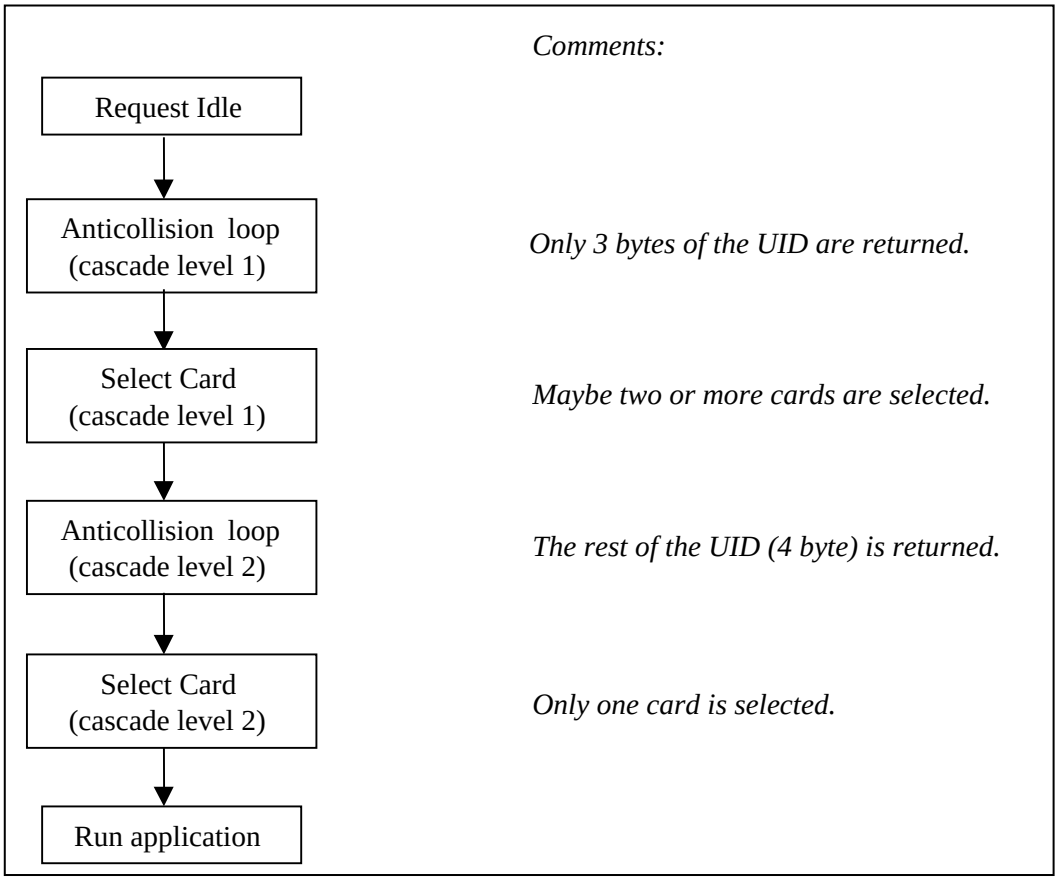


Figure 8: Transaction flow with Cascade Level 2

© NXP B.V. 2018. All rights reserved.

3.1.2 Transaction flow using Cascade Level 1 and Read Block 0

If the reader does not support the anti-collision cascade level 2, only the anti-collision cascade level 1 (ISO14443A-3) can be used to select a MIFARE Ultralight. This is the "Classic Anti-collision and it returns 3 significant bytes of the UID. In this case the complete UID shall be checked after selection with a read of block 0 to make sure, that only one card is selected. **If a collision is detected** during that read of block 0, the user has to be informed, that **only one card** has to be presented to the reader (see Figure 9).

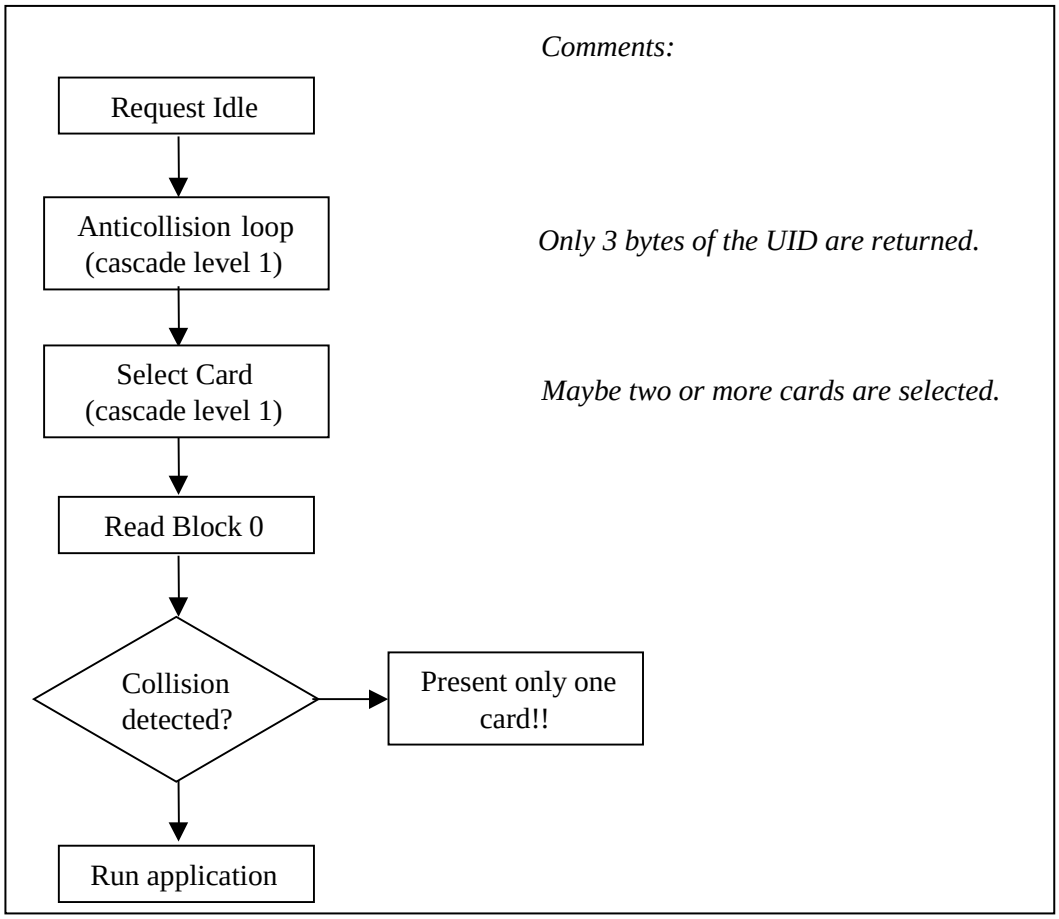


Figure 9: Transaction flow with Cascade Level 1 & Read Block 0

Remark:

This command flow as given in 3.1.2 doesn't follow the ISO14443, and only offers a compatible command flow to work with some old reader environment. If possible, the use of the complete anti-collision cascade level 1 and 2 is recommended.

© NXP B.V. 2018. All rights reserved.

3.2 MIFARE Ultralight and NFC Readers for MIFARE ICs

The MIFARE Ultralight can be selected, read, and written by every NFC reader for MIFARE Classic.

The MIFARE Classic authentication has to be skipped, and the selection of the MIFARE Ultralight has to be done as shown either in Figure 8 or Figure 9.

A MIFARE Classic Read command can be used. In this case only the first 4 Bytes contain valid data according to the addressed page; the other 12 bytes refer to the next 3 pages (see the related datasheet of the MIFARE Ultralight).

To write data into the memory of the MIFARE Ultralight, either the (4-byte) WRITE or the COMPATIBILITY WRITE can be used (see the related datasheet of the MIFARE Ultralight).

Reader Modules:

Reader	Anti-collision	WRITE	Comment
MF CM200	cascade level 2 possible, but LLL[5] has to be adapted[6]	possible, but LLL has to be adapted	supports MIFARE Ultralight
MF CM500	cascade level 2 possible, but LLL has to be adapted[7]	possible, but LLL has to be adapted	supports MIFARE Ultralight

Reader Devices:

Reader	Anti-collision	WRITE	Comment
MF RD260	only cascade level 1, no firmware update or extension possible	only COMPATIBILITY WRITE, no firmware update or extension possible	supports MIFARE Ultralight only in compatibility mode
MF RD560	only cascade level 1, no firmware update or extension possible	only COMPATIBILITY WRITE, no firmware update or extension possible	supports MIFARE Ultralight only in compatibility mode

Reader ICs:

[5] **L**ow **L**evel **L**ibrary

[6] example see 9.2

[7] example see 9.2

© NXP B.V. 2018. All rights reserved.

Reader	Anti-collision	WRITE	Comment
MFRC171	full cascade level 2 possible, but LLL has to be adapted	possible, but LLL has to be adapted	supports MIFARE Ultralight[8]
MFRC500	BFL[9] contains the full cascade level 2 support	BFL contains the full 4 byte WRITE support	supports MIFARE Ultralight
MFRC530	BFL contains the full cascade level 2 support	BFL contains the full 4 byte WRITE support	supports MIFARE Ultralight
MFRC531	BFL contains the full cascade level 2 support	BFL contains the full 4 byte WRITE support	supports MIFARE Ultralight
CLRC632	BFL contains the full cascade level 2 support	BFL contains the full 4 byte WRITE support	supports MIFARE Ultralight
MFRC522	BFL contains the full cascade level 2 support	BFL contains the full 4 byte WRITE support	supports MIFARE Ultralight
MFRC523	BFL contains the full cascade level 2 support	BFL contains the full 4 byte WRITE support	supports MIFARE Ultralight
CLRC663	NFC Reader Library contains the full cascade level 2 support	NFC Reader Library contains the full 4 byte WRITE support	supports MIFARE Ultralight

4. MIFARE Ultralight EV1 Counters

The MIFARE Ultralight EV1 MF0ULx1 features three independent 24-bit one-way counters (see [MF0ULx1]). The counters are initialised to 000000h. They can be read using the READ_CNT command and increased using the INCR_CNT command.

An example is indicated in **Figure 10**.

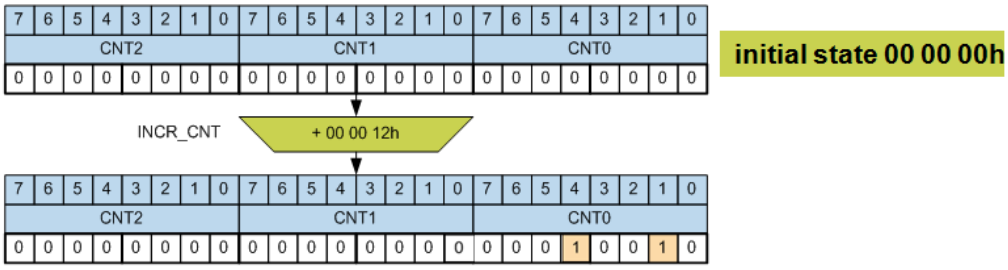


Figure 10: Counter increased of 18 (12h)

MIFARE Ultralight EV1 counters have anti-tearing protection that means that either the old value or the new (just written) value is present. It is recommended to use the following steps when increasing the counter for e.g. ticketing purposes:

[8] example see 9.1

[9] BFL means Basic Function Library

© NXP B.V. 2018. All rights reserved.

I) Read the counter using READ_CNT in order to check the current counter value

II) Increase the counter using INCR_CNT:

a. If the INCR_CNT response is equal to NAK5/7, this is a tearing event, repeat steps I) to III),

b. If the INCR_CNT response is equal to NAK6, the counter is corrupted or unusable, invalidate the ticket, or

c. In case of timeout, repeat steps I) to III),

d. If the INCR_CNT response is equal to ACK, execute step III)

III) Read again the counter using READ_CNT check that the new expected counter value has been correctly stored.

a. If the value is not correctly stored, repeat steps I) to III) up to N times

b. If after N times the value is still not correct invalidate the ticket

4.1 Re-loadable counter

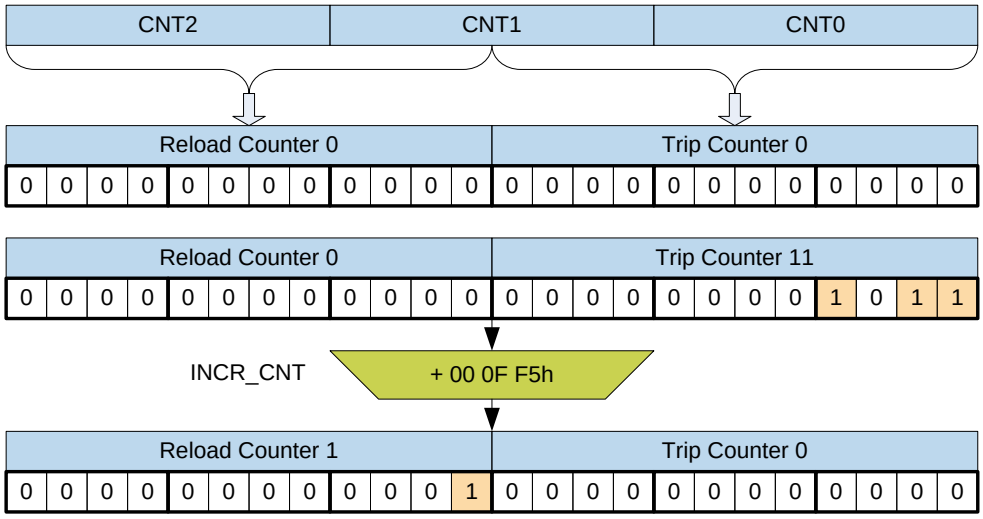


Figure 11: Re-loadable counter

The 3-bytes counter can be used for re-loadable tickets, as indicated in the **Figure 11**. The counter is split in 2 parts: a *Reload Counter* and a *Trip Counter*. The *Trip Counter* is counting the number of trips. In **Figure 11** the ticket has been used for 11 trips and then reloaded (e.g. reset). At the same time when reloading the *Reload Counter* is increased by 1.

© NXP B.V. 2018. All rights reserved.

4.2 Value counter protected by means of MAC

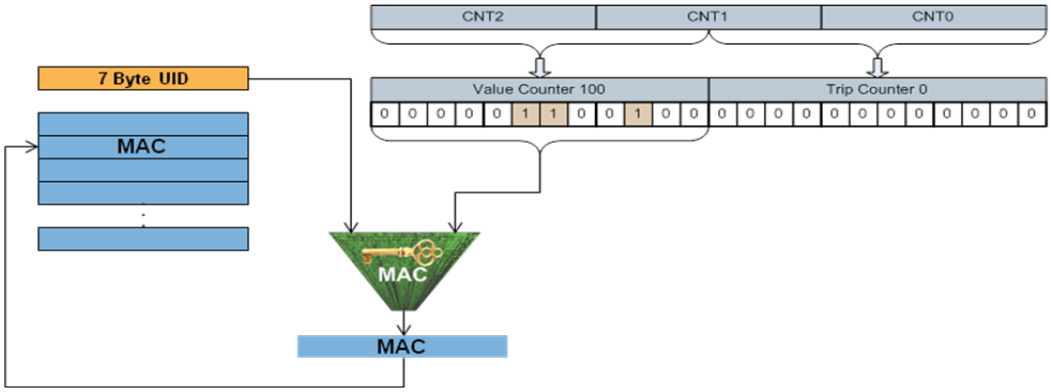


Figure 12: Value (e.g. the value- and the trip- counter) counter protected by MAC.

A value counter e.g. the counter is split into a *Value Part* and the *Trip Counter* (i.e. the ticket is storing a value in it), needs to be protected by a MAC stored in the ticket itself (an example is described in section 2.2.2) and described in **Figure 12**.

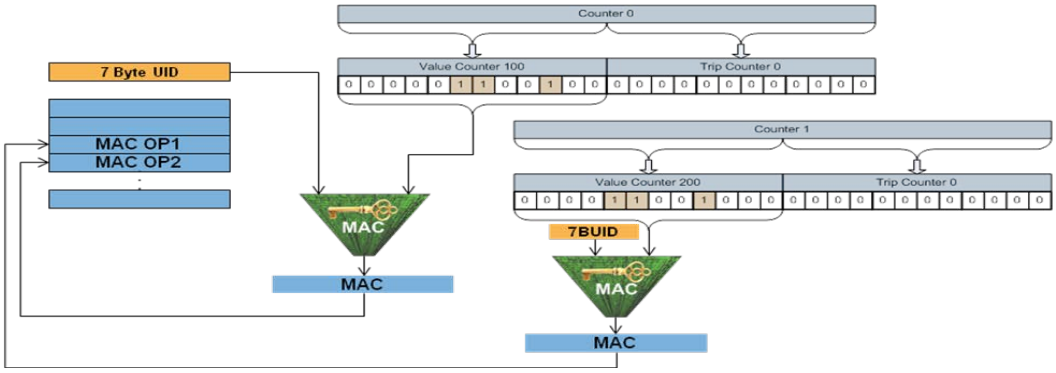


Figure 13: Value counters for multiple operator support.

The MIFARE Ultralight EV1 MF0ULx1 supports up to 3 independent counters. **Figure 13** shows a typical example of 2 value counters assigned to 2 different operators.

4.3 Counter for zone/mileage based scheme

The MIFARE Ultralight EV1 MF0ULx1 counter can also be used in a zone- or mileage-based tariff scheme in a check-in – check-out system. In this case the counter is split into 2 parts: one part containing the *Accumulated Total Fare* of the ticket and a second part containing the *Temporary Fare* as indicated in **Figure 14**.

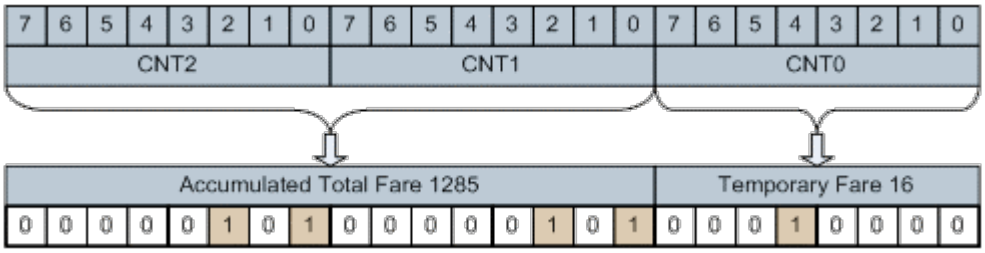


Figure 14: Counter used in a zone- or mileage- based tariff scheme at check-in.

© NXP B.V. 2018. All rights reserved.

At the check-in the full-fare (16 credits) is stored in the *Temporary Fare*. **Figure 14** shows this case and the storage of the value 16 (full fare) in the *Temporary Fare*.

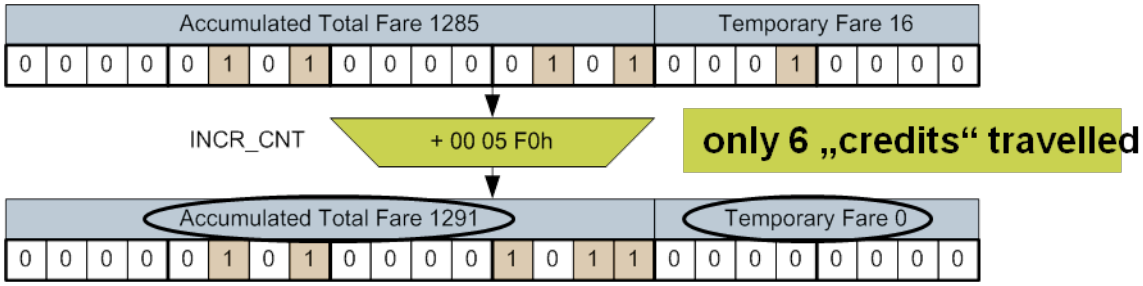


Figure 15: Counter used in a zone- or mileage- based tariff scheme at check-out where only 6 credits have been travelled.

At the check-out the real travelled credits are accumulated in the *Accumulated Total Fare* and the *Temporary Fare* is reset, see **Figure 15**.

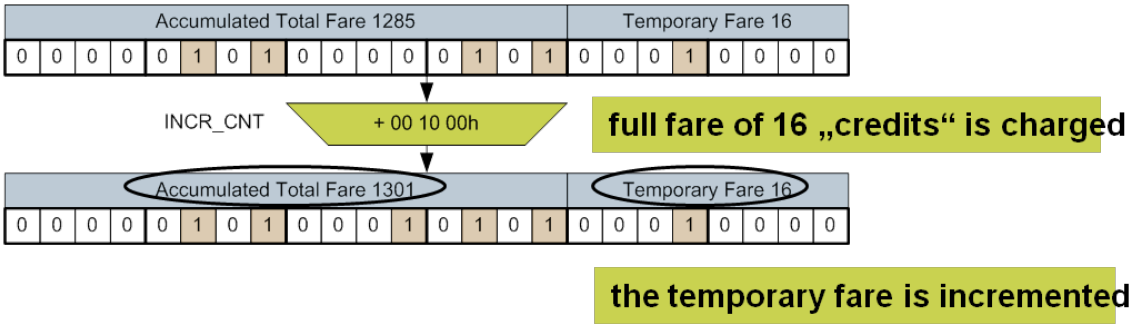


Figure 16: Counter used in a zone- or mileage- based tariff scheme at check-in in case the passenger did not check-out.

If the passenger does not check-out, at the next check-in the full fare (16 credits) is charged in the *Accumulated Total Fare* and the *Temporary Fare* is set again to 16 credits.

5. MIFARE Ultralight EV1 Password and PACK

The MIFARE Ultralight EV1 MF0ULx1 provides a password authentication to limit a part of the memory area for being accessed either in writing or reading and writing (see [MF0ULx1]).

Although the password verification method available in MIFARE Ultralight EV1 MF0ULx1 does not offer a high security protection, it can be as well (beside the originality signature check described in Table Document information) used to verify the originality of the ticket/card. Please note that the password and the PACK are sent in plain and this needs to be considered when assessing the system security.

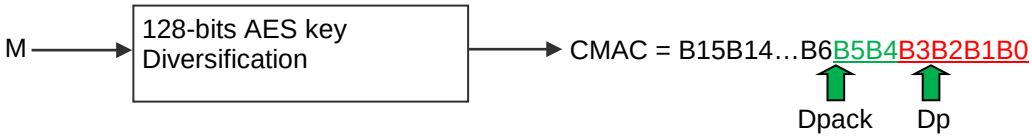
5.1 Password and PACK diversification

In case the password authentication is used, it is recommended to diversify the Password and the PACK to reduce the risk of compromise password/PACK. The diversification is done similarly to the key diversification described in [AN10922] section 2.2 “AES-128 Key”. In this case the following items are defined:

© NXP B.V. 2018. All rights reserved.

- K: a 16 bytes AES 128 bits Master Key
- M: the 7 bytes UID of the MIFARE Ultralight EV1, also called diversification inputs
- CMAC: the output from the 128-bits AES key Diversification called “diversified key” as indicated in Section 2.2 and Figure 2 of [AN10922]
- Dp: diversified Password
- Dpack: diversified PACK

The figure below describes the diversification scheme and how to obtain the diversified Password and PACK.



From the figure the Dp is obtained from the 4 LSB of the CMAC indicated as B3…B0, and the Dpack is derived from the next 2 bytes indicated as B5B4.

6. MIFARE Ultralight EV1 Anti-cloning based on Originality Check

The MIFARE Ultralight EV1 MF0ULx1 supports the originality function based on a 32-byte ECC signature (see [MF0ULx1]). The application note [MFULEV1SIGNVA] describes how to validate the signature (retrieved from the MIFARE Ultralight EV1 using the READ_SIG command) using the MIFARE Ultralight EV1 UID (Unique IDentifier) and the ECC public key provided by NXP Semiconductors.

To check that the card has not being cloned it is sufficient to verify the signature of the ticket. If the signature verification is correct, it means that the ticket-IC is a genuine NXP Semiconductors IC. Note that the signature contains the UID that is so implicitly verified during the signature verification.

The purpose of originality check shall be to protect customer investments by identifying mass penetration of non NXP originated MF0EV1 ICs into existing infrastructure scheme. The purpose of originality check shall NOT be to completely prevent HW copy or emulation of individual MF0EV1 ICs.

7. MIFARE Ultralight EV1 Tearing Application Implementation

The MIFARE Ultralight EV1 implements anti-tearing for OTP, Lock bits and counters (see [MF0ULx1]). This means that in case of a tear event either the old value or the new (just written) value is present. This section describes how it is possible to generically implement in MIFARE Ultralight a tearing application i.e. how to store application data that provide a protection against tearing events.

For the tearing application implementation 2 memory areas having the same size are needed see **Figure 17**.

© NXP B.V. 2018. All rights reserved.

Figure 17: tearing application implementation.

The application data is stored in 2 memory locations. The application data also contains a timestamp indicated in white and a CMAC (that can be calculated as indicated in section 2.2). Every time a new update is needed i.e. new data has to be written, only the set of data with the older timestamp is updated. The CMAC is added to guarantee the integrity of the written application data.

In particular, the **Figure 17** shows a typical update of the Application Data done on the older Application Data set (timestamp = t-1). As soon as the new application data is written the timestamp is updated (timestamp = t+1) and the CMAC is also written.

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

8. MIFARE Ultralight Coil design hints

The MIFARE Ultralight chip is available in two versions: either the **"standard" version MF0ICU10** with an input capacitance of approximately 17pF or a **high capacitance version MF0ICU11** with approximately 50pF. For a complete coil design please refer to the "MIFARE (Card) Coil Design Guide" [M011731].

Using the standard version of the MIFARE Ultralight chip it's recommended to use **the same coil design for the MIFARE Ultralight as for the MIFARE Classic**. Although the MIFARE Ultralight has a slightly higher capacitance than the MIFARE Classic (by 0.5pF), the same coil design should be used to result in a slightly lower resonance frequency. This lower resonance frequency increases the overall performance of cheap antennas and ensures a similar performance compared to MIFARE Classic – but has its limitation, if multiple cards operate simultaneously in the field.

For coil design issues it's recommended to use the Application notes "MIFARE (Card) Coil Design Guide" [M011731] and "Temperature Management, Inlet Design" [SI070010].

© NXP B.V. 2018. All rights reserved.

9. Appendix

9.1 MF RC171 low level library extension: Cascade Anticollision

LLL adaptation to execute cascade level 2, see section 3.2

```
/*****************************************************************************/
int CALL_CONV MfPiccCascAnticoll (unsigned char select_code,
                                  unsigned char bcnt,
                                  unsigned char *snr)
/*****************************************************************************/
{
  int           status;
  unsigned char snr_chk = 0;
  int           i;

  if (MfAssertMode(select_code,0x93|0x95|0x97))
      return (MI_WRONG_PARAMETER_VALUE);

  MfOutp(ENABLE, _PEN | _PRE);         // CRC-disable, Parity enable
  MfOutp(MODE  , __mode);              // __mode preset
  MfOutp(BCNTS ,(unsigned char)(bcnt + 16));           // 16 + number of bits
  MfOutp(STACON, (unsigned char)(__stacon|_AC));        // anticollision-mode
  MfDelay50us(4);                                  // BUS-access not allowed
                                                   // for 35us
  MfOutp(DATA, select_code);           // "SELTYPE" of MIFARE1
  MfOutp(DATA, (unsigned char)(((2 + (bcnt >> 3)) << 4) | (bcnt & 0x07)));
                                                    // bytecount higher nibble
                                                    // bitcount  lower  nibble
                                                    // incl. first 2 bytes!!
  for (i = 0; i < (bcnt + 7)/8; i++)
  {
    MfOutp(DATA, snr[i] );
  }
  MfOutp(TOC, TIMEOUT_14443_3); // set timeout
  while (!((status = MfInp(STACON)) & _DV));
  MfOutp(TOC, 0);         // reset timer

  if ((status = MfInp(STACON)) & (_TE | _BE))       // any error
  {
     if (status & _TE)
       return (MI_NOTAGERR);
     if (status & _BE)
     {
       MfDelay50us(10);            // delay 500us
       return (MI_BITCOUNTERR);
     }
  }
  for (i = 0; i < 4; i++)
  {
     snr[i] = MfInp(DATA);
     snr_chk ^= snr[i];
  }
  snr_chk ^= MfInp(DATA);
  // serialnumber check
  if (snr_chk)
    return (MI_SERNRERR);
  return (MI_OK);
}
```


© NXP B.V. 2018. All rights reserved.

9.2 MF CM200 / CM500 low level library extension: Cascade Anticollison

LLL adaptation to execute cascade level 2, see section 3.2

```
/*****************************************************************************/
int CALL_CONV MfPiccCascAnticoll (unsigned char select_code,
                                  unsigned char bcnt,
                                  unsigned char *snr)
/*****************************************************************************/
{
  int           status;
  unsigned char snr_chk = 0;
  int           i;

  if (MfAssertMode(select_code,0x93|0x95|0x97))
      return (MI_WRONG_PARAMETER_VALUE);

  MfOutp(ENABLE, _PEN | _PRE);         // CRC-disable, Parity enable
  MfOutp(MODE  , __mode);              // __mode preset
  MfOutp(BCNTS ,(unsigned char)(bcnt + 16));           // 16 + number of bits
  MfOutp(STACON, (unsigned char)(__stacon|_AC));        // anticollision-mode
  MfDelay50us(4);                                 // BUS-access not allowed
                                                  // for 35us
  MfOutp(DATA, select_code);           // "SELTYPE" of MIFARE1
  MfOutp(DATA, (unsigned char)(((2 + (bcnt >> 3)) << 4) | (bcnt & 0x07)));
                                                   // bytecount higher nibble
                                                   // bitcount  lower  nibble
                                                   // incl. first 2 bytes!!
  for (i = 0; i < (bcnt + 7)/8; i++)
  {
    MfOutp(DATA, snr[i] );
  }
  MfOutp(TOC, TIMEOUT_14443_3); // set timeout
  while (!((status = MfInp(STACON)) & _DV));
  MfOutp(TOC, 0);         // reset timer

  if ((status = MfInp(STACON)) & (_TE | _BE))       // any error
  {
     if (status & _TE)
       return (MI_NOTAGERR);
     if (status & _BE)
     {
       MfDelay50us(10);            // delay 500us
       return (MI_BITCOUNTERR);
     }
  }

  for (i = 0; i < 4; i++)
  {
     snr[i] = MfInp(DATA);
     snr_chk ^= snr[i];
  }
  snr_chk ^= MfInp(DATA);
  // serialnumber check
  if (snr_chk)
    return (MI_SERNRERR);
  return (MI_OK);
}
```


© NXP B.V. 2018. All rights reserved.

9.3 Worked out example of proposed security mechanism

An example application flow diagram is shown in the following:

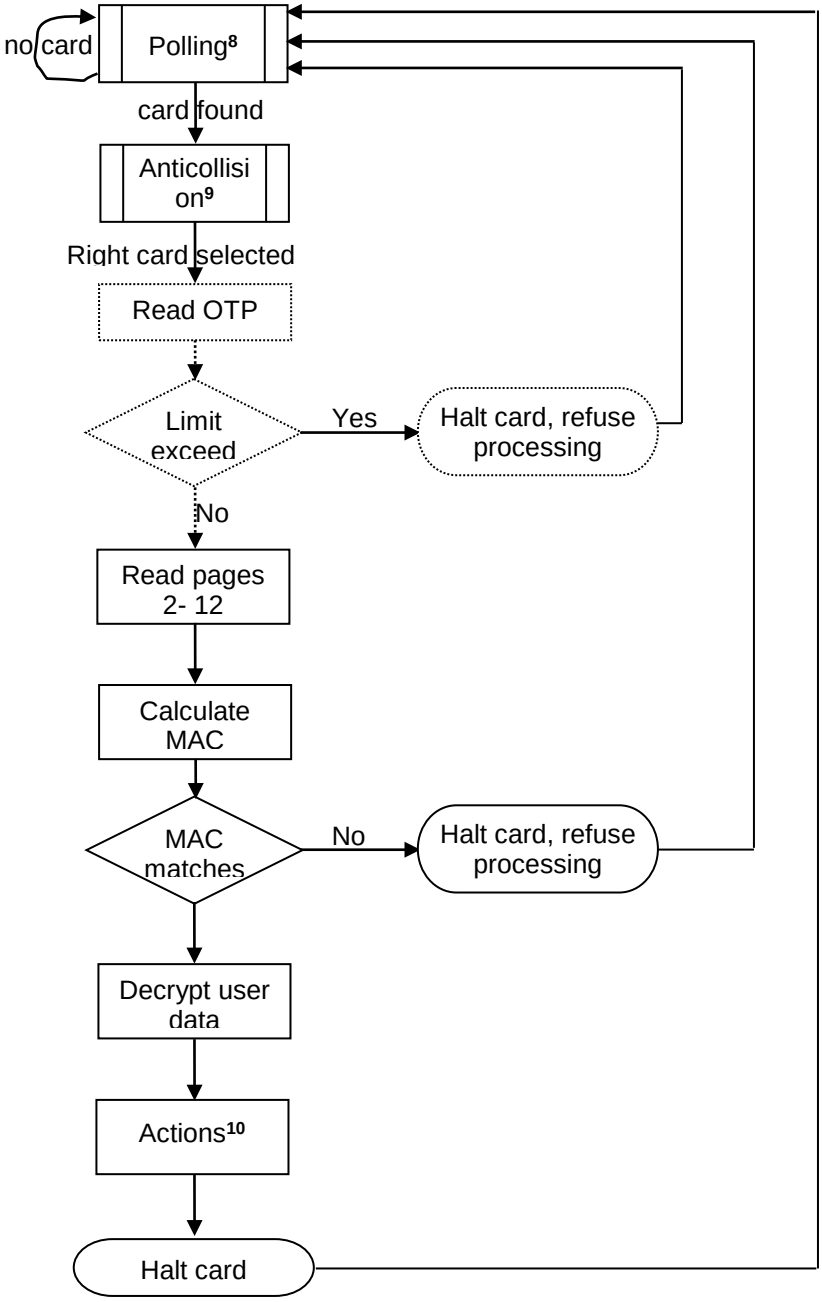


Figure 18: Example application flow diagram

Dotted blocks may be avoided if the OTP bytes are not used

[8] Pre-defined process for card detection, reader sends always REQA and check if there is any answer.

[9] Standard anticollision [ISO/IEC 14443-3], which includes the selection of the right card (also from the multiple cards).

[10] If OTP or any memory content is updated, MAC has to be recalculated and rewritten.

© NXP B.V. 2018. All rights reserved.

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. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

10.3 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.

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

© NXP B.V. 2018. All rights reserved.

11. Contents

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. 2018. All rights reserved.

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

Date of release: 9 July 2018 **073131**

Document identifier: AN11340