AN12344

MIFARE DESFire Light Target Applications and Usage

Rev. 1.0 — 31 January 2019 Application note
522710 COMPANY PUBLIC

Document information

Information	Content
Keywords	MIFARE, MIFARE DESFire Light, MIFARE DESFire EV2, ISO/IEC 7816, AES, LRP, Applications, Access Management, Micropayment, Loyalty, Event Management
Abstract	This application note gives examples of how MIFARE DESFire Light can be used in selected target applications. It explains how the file system can be utilized and configured in the optimum way for fitting to different kind of environments and infrastructures like access management systems, micropayment / loyalty applications or the event sector.

Revision History

Revision history

Rev	Date	Description
1.0	20190131	Initial version

1 Abbreviations

Table 1. Abbreviations

Acronym	Description
AID	Application Identifier
APDU	Application Protocol Data Unit
ATQA	Answer to Request-A, according to ISO/IEC 14443
ATS	Answer to Select, according to ISO/IEC 14443
C-APDU	Command-APDU
DF Name	Dedicated File Name. The DF Name to be used for the ISO Select command as defined in ISO 7816-4. According to EMV, the DF Name is also called AID (application identifier). To avoid confusion with the MIFARE DESFire Application ID (AID) this document uses the term DF Name for the ISO Select command.
DIV	Diversification
ISO Select	The ISO Select command used with the DF Name (ISO Select by DF Name), including the FCI response. Definition according to ISO/IEC 7816-4.
LRP	Leakage Resilient Protocol
NFC	Near Field Communication
PCD	Proximity Coupling Device (reader, terminal)
PDCap	Physical Device Capabilities
PICC	Proximity Integrated Circuit Card
PTO	Public Transport Operator
R-APDU	Response-APDU
SAK	Select Acknowledge
UID	Unique IDentifier

2 Introduction

MIFARE DESFire Light is a versatile contactless smart card platform targeting the single-application usage. It mainly addresses the needs of limited use application as well as simple extended use applications, offering a powerful mix between performance, security, privacy and flexibility.

This application note focuses on elaborating about potentially used target applications for MIFARE DEFire Light. Three of the mainly interesting purpose applications, namely access management, micropayment and event management have been picked out and selected to be further analyzed within this application note. An application structure mapping fitting to MIFARE DESFire Light is proposed inside this document.

2.1 About the content of this document

This document addresses developers, project leaders and system integrators who have a general technical understanding or are already familiar with the MIFARE DESFire product family. Knowledge of the reader terminal infrastructure or complete service infrastructure is good to have.

Note that this document does not cover the general working principle of the MIFARE DESFire Light. Read Ref [1] in order to get the full overview and description of MIFARE DESFire Light and the associated command set.

This application note is a supplementary document for implementations using MIFARE DESFire Light. Should there be any confusion, check the MIFARE DESFire Light data sheet Ref [1] or the Features and Hints application note. The best use of this application note is achieved by reading the mentioned data sheet in advance.

2.2 Structure of this document

This document describes the relevant information for realizing and structuring a MIFARE DESFire Light application. It is elaborated how different files can be used for different purposes and how the command sequences and interaction with the card via a contactless reader can be clustered.

Chapter Section 3 describes how the default application layout of MIFARE DESFire Light is structured and how the default configuration is set up.

The following subchapters, Section 3.1, Section 3.2 and Section 3.3, give specific suggestions on how to use the MIFARE DESFire Light IC for the mentioned applications and what is the best practice for utilizing the existing file system.

3 MIFARE DESFire Light Target Applications

MIFARE DESFire Light has a fixed memory structure according to the ISO/IEC 7816-4 file system, offering six pre-installed files. Details to the file system setup and the configurations which are already done in NXP´s manufacturing can be read in the MIFARE DESFire Light data sheet, [1].

The default configured file system of the chip is depicted in Figure 1.

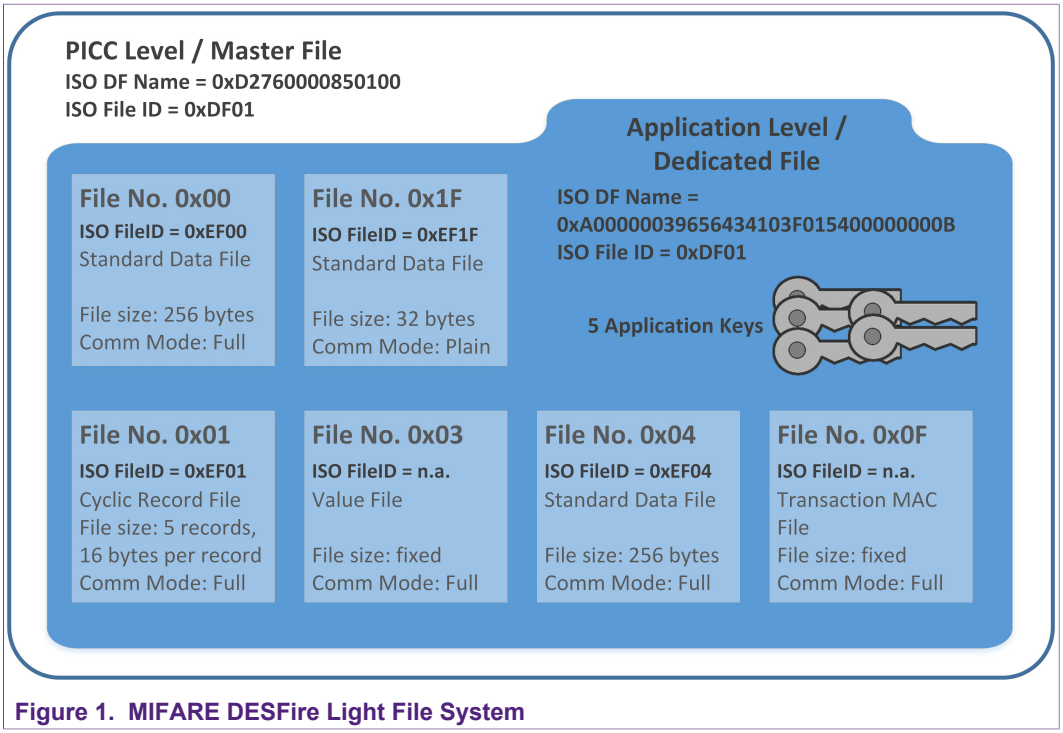

Figure 1. MIFARE DESFire Light File System

Also the access conditions and communication modes are already pre-configured for each file inside the application. The pre-assigned keys for the access rights can be seen in Figure 2. The 5 available application keys come with the default key type AES 128 bit and default key value (0x00000000000000000000000000000000).

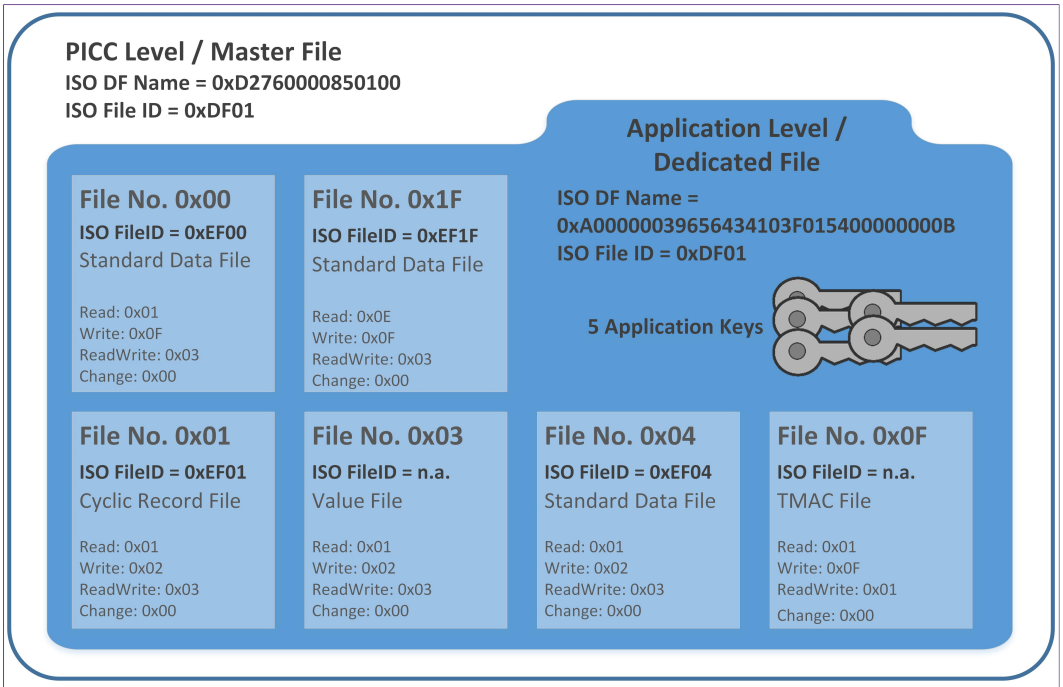

Figure 2. MIFARE DESFire Light File System and pre-configured Access Rights

This pre-configured file system builds a great basis for very easy, fast configurations and deployment of applications in the field.

MIFARE DESFire Light is easy to use and it is extremely straightforward to fill the available file system structure with the needed user data. Small configurations like setting keys, changing access rights, or changing general file parameters, can be done very fast and convenient during the IC personalization.

The outlined application architecture builds the backbone of all application examples which are suggested inside this document in the next chapters.

3.1 Access Management

MIFARE DESFire Light can be ideally used for any kind of access management application, be it corporate access, residential access or any other kind of access system.

It is also common to combine access management with identity management and realize a so-called IAM (Identity and Access Management) system, combining a user identity and access credential on the same device and managing it in the corresponding infrastructure. Also therefore, MIFARE DESFire Light is the perfect fit.

3.1.1 Application Structure

The pre-configured file system can be utilized in different ways for realizing an access management system with MIFARE DESFire Light.

In this section we focus on creating an access application for employees, accessing a corporate environment.

Depending whether the access system is operating online, offline or semi-online, different data items are required to be stored on the access card.

Mandatory functionalities, that the card shall be able to perform in this example:

- Authenticity - proof that the card belongs to the access control system
- Ownership - proof of the card owner, contain information about the card owner

Basically the mandatory items are enough for realizing a pure online system with only checking the owner and the related access rights in the server backend, once the card owner is tapping his card to the access reader.

Optional functionalities, that the card shall be able to perform in this example:

- Access rights / access profile - allowing access to an area also for an offline system
- Transaction logs - maintaining the usage history, showing when the user was accessing which area of the access management system
- Transaction counters - adding additional security

For realizing a basic corporate access management system that can work online as well as offline, the following example uses all of the above mentioned items / functionalities and incorporates them in the application layout.

Usage of the pre-configured files inside the application

- **Standard Data File - File No 0x00** - used for storing the employee`s access rights or access profile. The coding of access rights is up to the access control system. E.g., a general access right for managing access to the full building and then access levels

depending on specific areas inside the building. Another possibility would be to manage access inside the building for each room separately, or just to restrict access to certain separate zones and leaving access to all other rooms open.
Instead of File No 0x00 also File No 0x1F could be chosen to store the access rights, depending on the needed size and preference.

- **Cyclic Record File - File No 0x01** - used for storing a transaction log, showing when the employee was entering which access area. This file is used for realizing a complete access history. The point of entry (in this case the reader number / terminal id) can be included in the log, to have a detailed track of when / where / how the employee was moving around the building.

- **Standard Data File - File No 0x04** - used for storing the employee data. This can be only the employee identifier without any other personal information, or it can contain further information like name, date of birth, profession, desk number, etc. in case the card is personalized.

- **Transaction MAC File - File No 0x0F** - this file is used for enabling the Transaction MAC feature. It can be used for proofing the authenticity of executed transactions and adds an additional layer of security to the overall system. More details on the TMAC feature can be found in [4].

The other two available files remain unused for realizing the access application. Of course, they can be utilized in any way in case the access application structure should look different in the actual end application.

The access conditions of the files are left unchanged, so the default access conditions are a good fit for realizing this access application. Also the communication mode stays enabled to fully encrypted, as it was already pre-configured.

A visual representation of the access application can be seen in Figure 3.

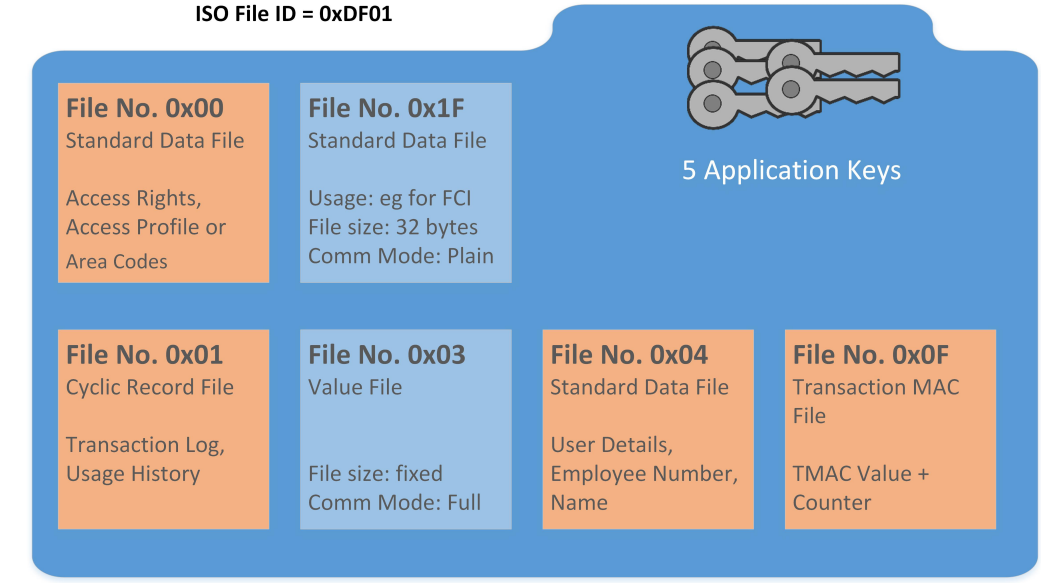

Figure 3. Access Management Application Structure

A sample transaction, showing the command flow for an easy micropayment application is available in Figure 4.

The transaction shows the following steps:

1. Activating the card on ISO/IEC 14443-Layer 3
2. Activating the card on ISO/IEC 14443-Layer 4
3. Selecting the application by using the ISOSelect command
4. Authenticating with Key 0x01 which acts as read key for File 0x00, File 0x01 and File 0x04
5. Checking the user data from File 0x04. Here information related to the user, employee ID, phone number, etc. might be stored and evaluated on the terminal side
6. Retrieving a transaction history from File 0x01, showing the last records that indicate when the employee was entering which room / area of the building. Suspicious movements in the building could be detected here
7. The data which has been read out from the card (user data and transaction history) can be checked on the reader, if needed. It also can be sent to the backend system to store everything there for further analysis purposes
8. Reading the employees access rights or access profile from File 0x00. Depending on this information, the reader might open or not after finishing the transaction
9. Authenticating with Key 0x03 which acts as read/write key for File 0x01
10. If the user has access to the room / area, a transaction record proofing that the employee was entering a certain door at which time (timestamp) and at which terminal (reader ID) is added to the record file, File 0x01. This helps for having a complete history of executed transactions and employee movements stored on the card, not only on the terminal. That is especially important in semi-online systems or systems where the online connection from the reader to the backend might get lost. In this case, the transaction history from the card can be read out at the next tap and reported to the backend for tracking purposes

11. Finalizing the transaction, the transaction is committed by using the CommitTransaction command. This concludes the transaction and triggers the TMAC calculation. The TMAC counter and TMAC value are returned to the terminal as a response to the CommitTransaction command
12. As the terminal can`t do anything meaningful with the TMAC counter and TMAC value, they need to be passed on (together with the log of executed commands during the transaction) to the related backend for transaction evaluation and TMAC checking

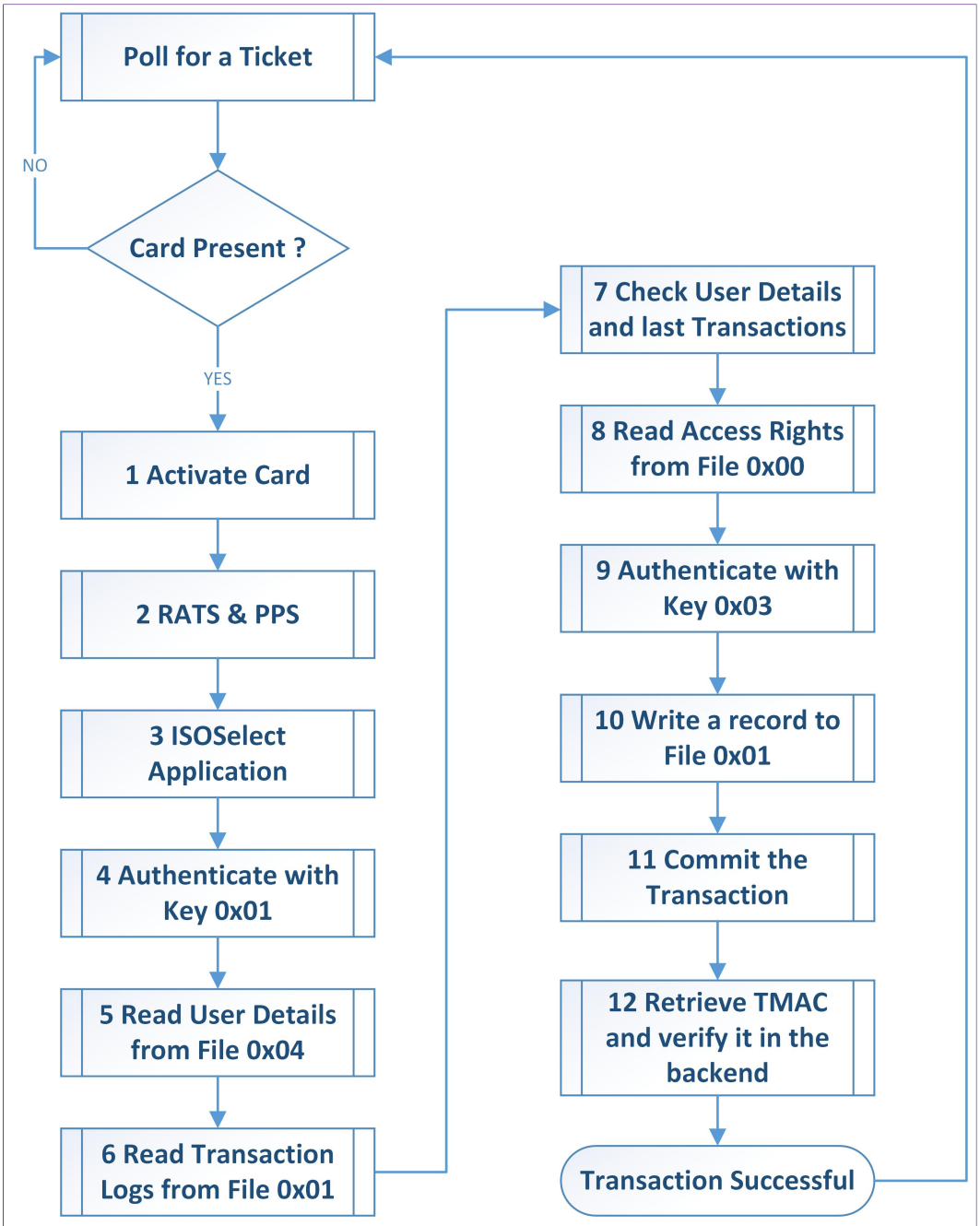

Figure 4. Access Management Transaction

3.2 Micropayment

Realizing micropayment and / or loyalty solutions based on MIFARE DESFire Light becomes very easy and straightforward by re-using the already configured file system.

The essential parts which are needed for implementing a micropayment application are the possibilities to securely store a value on the card as well as to have a money spending history available. Both of these things can be achieved with MIFARE DESFire Light by using on the one hand the value file for storing the value. On the other hand the cyclic record file for keeping track on the spendings.

3.2.1 Application Structure

In the following, one application structure for a micropayment application is suggested. This is only a proposal and the card can be utilized in countless alternative ways as well.

Usage of the pre-configured files inside the application

- **Value File - File No 0x03** - used for storing the currently available money value. At card issuance, this value is set to zero and the end user can upload money by himself via any provided service station of the micropayment infrastructure or at the cash desk directly (depending on how the infrastructure is designed). The value file is also limited by a minimum and a maximum value, meaning a certain value range can`t be exceeded.

- **Cyclic Record File - File No 0x01** - used for storing a transaction log, showing when money was uploaded to the card and when money was spent. This file is used for realizing a complete money usage history. The point of usage (in this case the reader number / terminal id) can be included in the log, to have a detailed track of when / where / how much money was accessed.

- **Standard Data File - File No 0x04** - used for storing the user data in case the card is personalized. User data like a member id, name, postal address, phone number, date of birth and much more can be stored in this file, for offering a personalized card and also personalized advertisement and user interaction. In case of unpersonalized / anonymous cards, this file can remain empty.

- **Transaction MAC File - File No 0x0F** - this file is used for enabling the Transaction MAC feature. It can be used for proofing the authenticity of executed transactions. More details on the TMAC feature can be found in [4].

The other two available files remain unused for realizing the micropayment application. Of course, they can be utilized in any way in case the micropayment application structure should look different in the actual end application.

The access conditions of the files are left unchanged, so the default access conditions are a good fit for realizing this micropayment application. Also the communication mode stays enabled to fully encrypted, as it was already pre-configured.

A visual representation of the micropayment application can be seen in Figure 5.

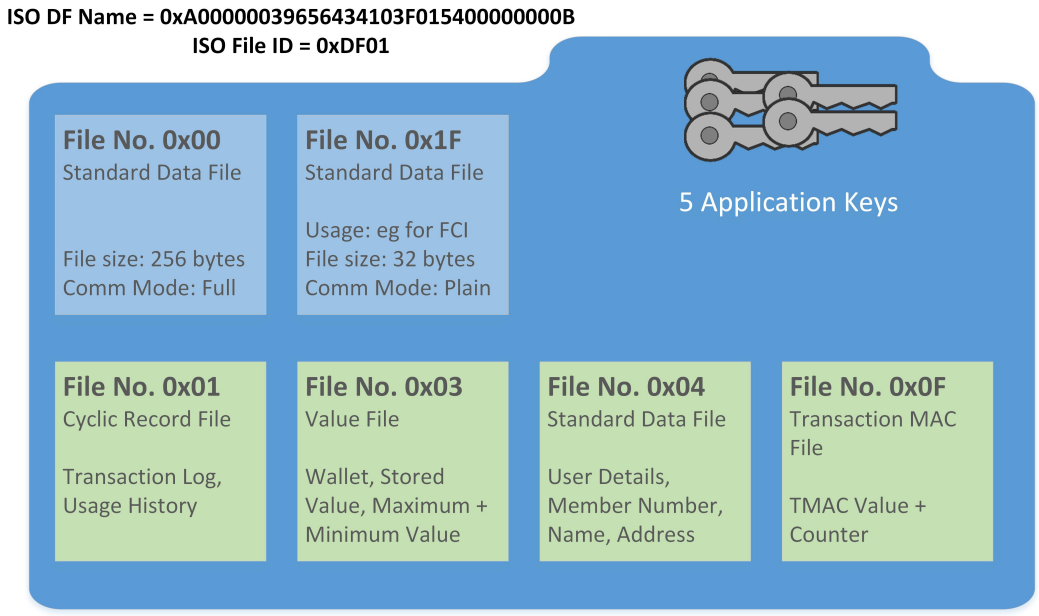

Figure 5. Micropayment Application Structure

A sample transaction, showing the command flow for an easy micropayment application is available in Figure 6.

The transaction shows the following steps:

1. Activating the card on ISO/IEC 14443-Layer 3
2. Activating the card on ISO/IEC 14443-Layer 4
3. Selecting the application by using the ISOSelect command
4. Authenticating with Key 0x01 which acts as read key for File 0x01, File 0x03 and File 0x04
5. Checking the user data from File 0x04. Here information related to the user, membership ID, special concessions, etc. might be stored and evaluated on the terminal side
6. Retrieving a transaction history from File 0x01, showing the last records that indicate when money was deducted and also increased on the card
7. The data which has been read out from the card (user data and transaction history) can be checked on the reader, if needed. It can also be sent to the backend system to store everything there for further analysis purposes
8. Reading the currently stored value from File 0x03
9. Authenticating with Key 0x03 which acts as read/write key for File 0x01, File 0x03 and File 0x04
10. Depending on the selected user action, either a credit or a debit command can be performed now, affecting File 0x03. Meaning either that money is deducted for a standard micropayment operation, or that money gets loaded onto a card because the user did some value reloading to stock up his balance
11. After modifying the value stored in File 0x03, a transaction record proofing which kind of command (credit / debit) has been executed at which time (timestamp) and at which terminal (reader ID) is added to the record file, File 0x01. This helps for

having a complete history of executed transactions stored on the card, not only on the terminal. That is especially important in semi-online systems or systems where the online connection from the reader to the backend might get lost. In this case, the transaction history from the card can be read out at the next tap and reported to the backend for tracking purposes.

12. Finalizing the transaction, the transaction is committed by using the CommitTransaction command. This concludes the transaction and triggers the TMAC calculation. The TMAC counter and TMAC value are returned to the terminal as a response to the CommitTransaction command
13. As the terminal can`t do anything meaningful with the TMAC counter and TMAC value, they need to be passed on (together with the log of executed commands during the transaction) to the related backend for transaction evaluation and TMAC checking

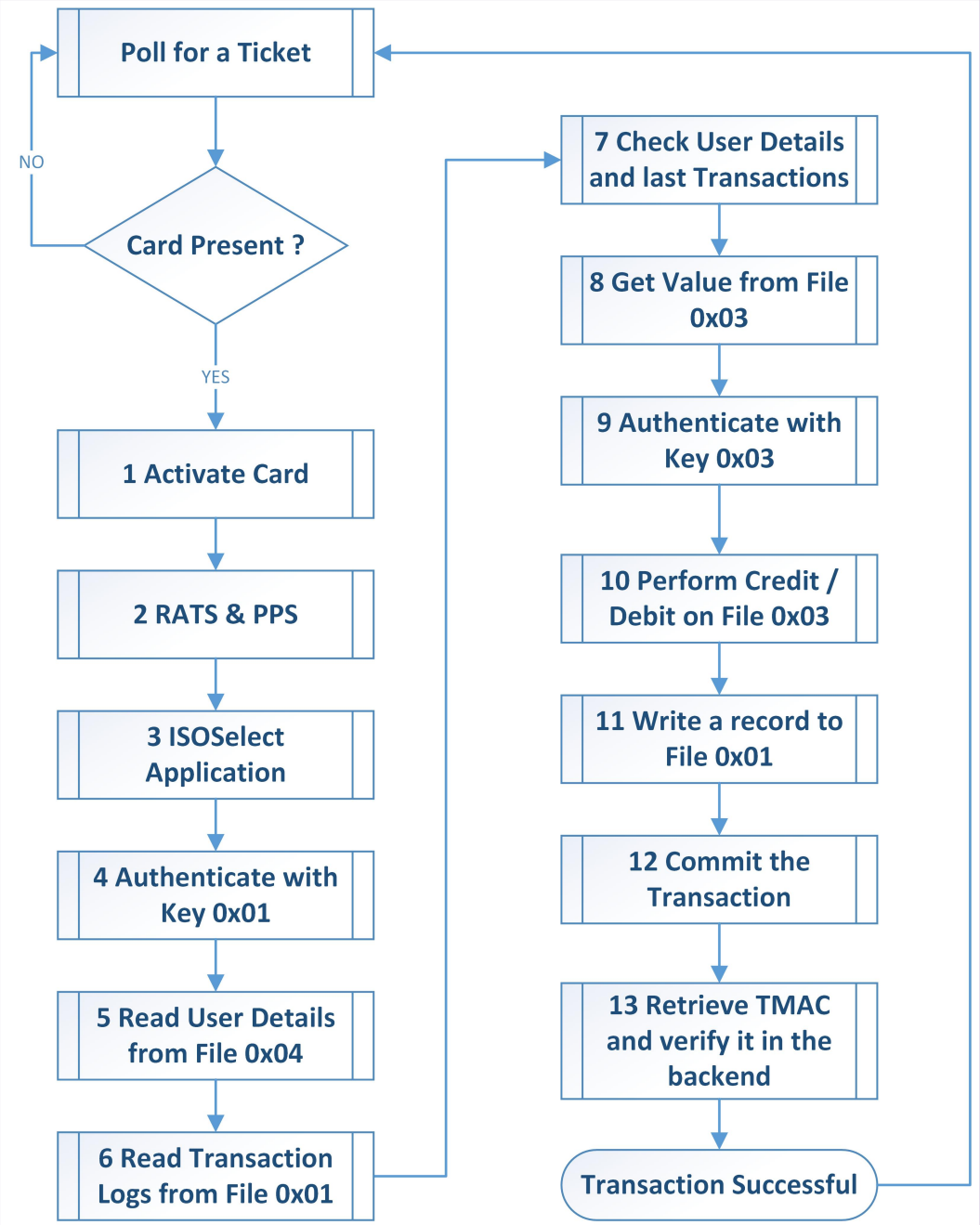

Figure 6. Micropayment Transaction

3.3 Event Management

MIFARE DESFire Light is the perfect product fitting different kinds of event management systems and event infrastructures.

Vital for realizing event visitor badges is the combination of different purposes which can be any of:

- Event location access
- In-location access area differentiation (e.g. VIP area, special guest area, different event areas for different visitor groups, ...)
- Micropayment at the event
- Mobile top-up for loading currency which can be used at the event

Note: For privacy reasons it might be advisable to enable Random ID, for event access or public events in general, on the chip, to limit tracking possibilities of the card holder. By enabling Random ID of the card, the card shows a different 4 byte RID on each card activation (each card tap to the reader), which makes a tracking of the user (card holder) impossible.

Analyzing the different possible purposes of an event badge, it can be seen that access and micropayment purposes can be combined in the same application for realizing an event application.

As access management usage and micropayment usage were already described in the previous chapters, this section tries to combine the outlined approaches into one event application.

3.3.1 Application Structure

In this section, a simple application structure usage for an event management application is suggested, by re-using some of the suggestions that were already made in the access and micropayment application.

Usage of the pre-configured files inside the event application

- **Standard Data File - File No 0x00** - used for storing the visitor`s access rights or access profile. The coding of access rights is up to the access control system. E.g., a general access right for managing access to the full building and then access levels depending on specific areas inside the building / event area. Another possibility would be to manage access inside the building for each room separately, or just to restrict access to certain separate zones (e.g. VIP area) and leaving access to all other rooms open.

  Important for event badge is also a timely limitation of access to the event location. Depending on the available options like 1-hour pass, 1-day pass, 3-day pass, 1-week pass, etc., a timestamp signalizing the end validity of this event badge can be evaluated once the card is tapped on the reader terminal.
  Instead of File No 0x00 also File No 0x1F could be chosen to store the access rights, depending on the needed size and preference.

- **Value File - File No 0x03** - used for storing the currently available money value for on-event payment. At card issuance, this value is set to zero and the end user can upload money by himself via any provided service station of the infrastructure or at the cash desk directly (depending on how the infrastructure is designed), or with over-the-air topup services if available. The value file is also limited by a minimum and a maximum value, meaning a certain value range can`t be exceeded.

- **Cyclic Record File - File No 0x01** - used for storing a transaction log, showing all transactions that are done at the event. It includes access transaction as well as monetary transactions. This file is used for realizing a complete card usage history. The point of usage (in this case the reader number / terminal id) can be included in the log, to have a detailed track of when / where / how much money was accessed and also when / where / how the event visitor was moving around in the event location.

- **Standard Data File - File No 0x04** - used for storing the user data in case the card is personalized. User data like a user id, name, postal address, phone number, date of birth and much more can be stored in this file, for offering a personalized card and also personalized advertisement and user interaction. In case of unpersonalized / anonymous cards, this file can remain empty.

- **Transaction MAC File - File No 0x0F** - this file is used for enabling the Transaction MAC feature. It can be used for proofing the authenticity of executed transactions. More details on the TMAC feature can be found in [4].

In this suggested layout, nearly all files of the event management application are used, having no more space left for storing any kind of other data. Of course, the files can be utilized also in any other way in case the event application structure should look different in the actual end application.

The access conditions of the files are left unchanged, so the default access conditions are a good fit for realizing this access application. Also the communication mode stays enabled to fully encrypted, as it was already pre-configured.

A visual representation of the event management application can be seen in Figure 7.

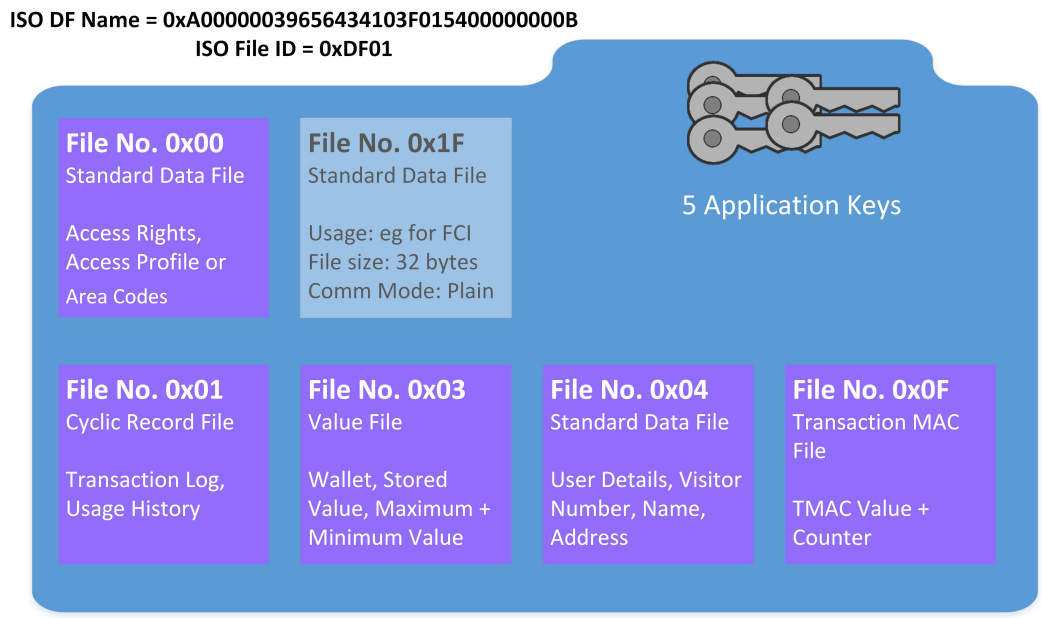

Figure 7. Event Management Application Structure

As the event application realizes two use cases, namely event access and in-event micropayments, the transaction that is executed depends on the use case. It depends on the reader on which the card is tapped by the user to decide whether an access transaction or a monetary transaction should be executed.

When entering the event location and moving across the location area, changing rooms and floors, the access system readers perform an access management transaction which can be similar to the one that is described in Figure 4.

When using the card for making in-event payments for food, drinks, goodies, etc., or loading value on the event badge directly at the event, the payment readers perform a micropayment transaction which can be similar to the one that is described in Figure 6.

4 References

[1] Product data sheet - MIFARE DESFire Light contactless application IC, document number 4307xx[1], available in NXP DocStore and on the NXP website https://www.nxp.com/docs/en/data-sheet/MF2DL_H_x0.pdf

[2] Product data sheet - MIFARE DESFire EV2 contactless multi-application IC, document number 2260xx, available in NXP DocStore

[3] Application Note - AN10922 Symmetric key diversifications, document number 1653xx, available in NXP DocStore and on the NXP website https://www.nxp.com/docs/en/application-note/AN10922.pdf

[4] Application Note - AN12343 MIFARE DESFire Light Features and Hints, document number 5225xx, available in NXP DocStore and on the NXP website https://www.nxp.com/docs/en/application-note/AN12343.pdf

[1] xx … document revision number

5 Legal information

5.1 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.

5.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors. In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory. Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

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

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

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification. Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products. NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

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

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer. In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages. Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

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

5.3 Trademarks

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

MIFARE — is a trademark of NXP B.V.

DESFire — is a trademark of NXP B.V.

Tables

Figures

Contents

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

Date of release: 31 January 2019
Document identifier: AN5227
Document number: 522710