AN12113

Over-the-Air top-up with MIFARE DESFire and MIFARE Plus

Rev. 1.2 — 2 July 2020 Application note
456712 COMPANY PUBLIC

Document information

Information	Content
Keywords	Over-the-Air, OTA, Top-up, MIFARE DESFire, MIFARE Plus, Stored Value, Transport Ticketing, Loyalty, Closed-loop Payment
Abstract	This application note describes, how OTA top-up can be realized in combination with MIFARE DESFire and MIFARE Plus ICs.

Revision History

Revision history		
Rev	**Date**	**Description**
1.2	20200702	Removed specific product versions and made the document generally applicable to MIFARE DESFire and MIFARE Plus products.
1.1	20191028	Section for MIFARE Plus EV1 added
1.0	20180108	Initial version of the document

1 Introduction

In this document the usage of over-the-air (OTA) services, especially over-the-air top-up functionality, in combination with MIFARE DESFire and MIFARE Plus, will be discussed.

As over-the-air services are quite convenient for both the system operator as well as the end user, OTA gets more and more important nowadays. OTA can be used in many different applications and numerous purposes, the most common one being probably transport ticketing and update of stored values.

Section 2 talks about over-the-air services in general and describe the concept behind OTA. Later on, Section 3 illustrates, how OTA can be used in combination with MIFARE DESFire and finally Section 4 section explains, how OTA can be used in combination with MIFARE Plus.

1.1 About this document

This document addresses developers and people who already have general know-how of the MIFARE DESFire ICs or the MIFARE Plus ICs and its command sets.

Please note that this document does not cover the general working principle of the MIFARE DESFire products or the MIFARE Plus products. Please read [1] in order to get the full overview and description of the MIFARE DESFire EV3, and [2] in order to get the full overview of the MIFARE Plus EV2.

This application note is a supplementary document for implementations using the MIFARE DESFire or MIFARE Plus family products. Should there be any confusion, check out the related product data sheets. The best use of this application note will be achieved by reading the mentioned corresponding documents data sheet in advance.

Note: This application note does not replace any of the relevant functional specifications, data sheets or design guides.

2 Over-the-Air (OTA) services and applications

Over-the-air services are referring to different kind of methods how to distribute new updates, configuration settings, keys or any other kind of data to devices of the end-customer. Important when speaking about the OTA is that there is one central instance which is contacted for requesting and distributing updates.

When using OTA in combination with an IC like MIFARE DESFire or MIFARE Plus, there needs to be a medium in between the smart card and the server backend, that is responsible for building up the communication between the smart card and the server / the central instance which distributes updates. This can be a mobile phone application, a self-service terminal (kiosk), a desktop application with attached reading device or a service station.

In this application note, the focus is put on the mobile phone application which supports the OTA feature and connects the smart card with the server backend. This is a very convenient solution for the end-user as nowadays nearly everybody possesses an NFC capable smartphone and can install the application that is required for the OTA service very easily.

The system operator, how it is called in this document, can be anyone running the smart card infrastructure, e.g. a public transport operator, a shop, a loyalty scheme operator, and many more.

2.1 Common OTA applications

Basically, there are no limitations to OTA – everything that can easily be managed and updated remotely via a central server could be a potential OTA application.

Examples for well-known OTA applications are:

- Public transportation
- Stored value applications
- Gift cards / Voucher cards
- Parking
- Closed-loop payment

2.2 Benefits of using OTA top-up services

Using the over-the-air as an additional feature for sure is very attractive for the end-customer but for the system operator as well.

For the end-customers, the benefits of using OTA services via a mobile phone application are countless. Among them are for example:

- Convenience
- Easy to use
- No cash needed
- No queuing at a self-service terminal or service station needed
- Topping up / Recharging can be done on demand
- Mobile application can be used to not only top-up, but also to keep the overview of currently available value / tickets / balance on the smart card

Benefits for the system operator, respectively the OTA service provider are:

- Customer satisfaction
- Cost reduction through
  - Less cost for ticket issuance
  - Less self-service terminals needed (if OTA mobile phone application is used)
  - Less staff for service stations needed
- Payment settlement in the backend
- Less cash flow due to backend / online payment

2.3 Working principle of OTA services

If the system operator wants to offer OTA top-up services to his customers, he needs to make sure to develop the needed infrastructure. What will be needed is for sure one mobile application and the corresponding server backend implementation.

The mobile application needs to be able to communicate with the smart card and forward information from the IC to the server backend and vice-versa.

The end-customer can then easily install the mobile application on his smartphone and via tapping the smart card, he can make use of the OTA service that the system operator offers to him.

Involvement of payment operators is out of scope of this document and is depending on the overall system setup. How exactly the payment gateway and the payment processor backend need to be included into the infrastructure is individual for each system operator.

An example OTA system is depicted in Figure 1.

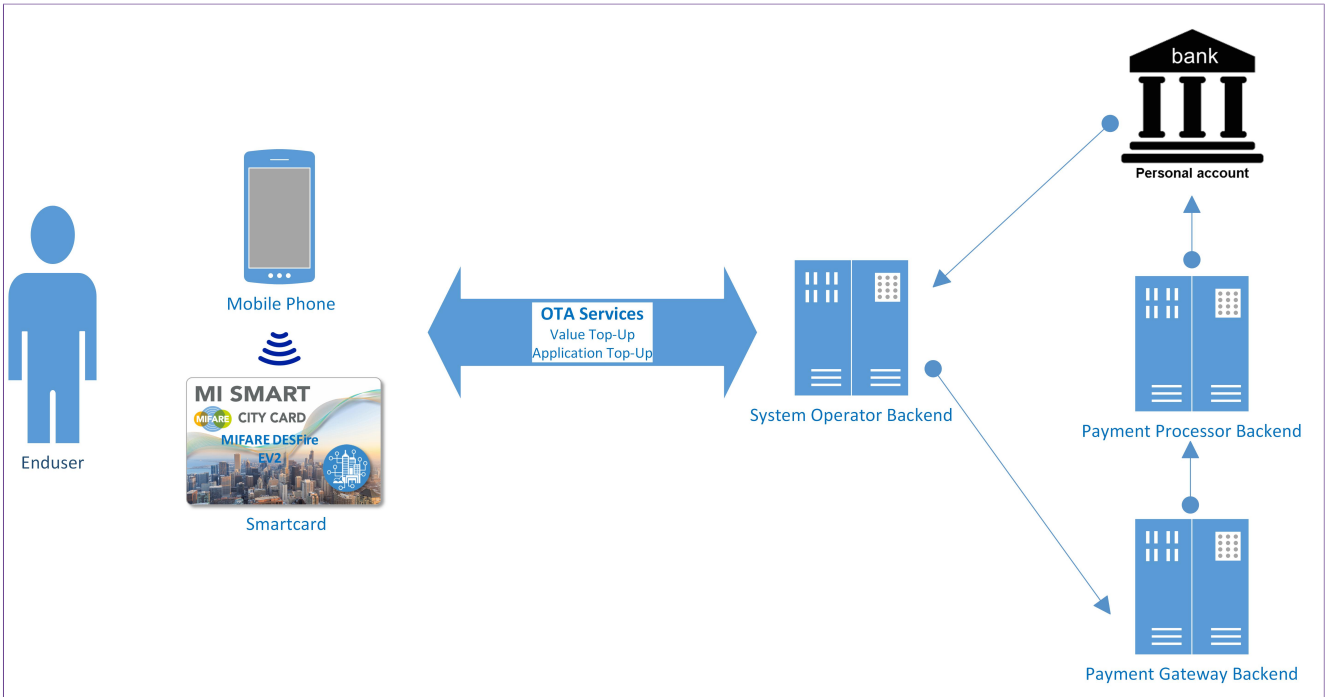

Figure 1. OTA System Infrastructure

3 OTA in combination with MIFARE DESFire

An over-the-air service, like for example a value top-up which will be the main discussion point of this document, can be realized for already existing MIFARE DESFire infrastructures and of course for new, upcoming MIFARE DESFire system installations.

If an over-the-air service is going to be implemented, the most important thing to consider is that the full communication of the top-up transaction, will be executed between the MIFARE DESFire IC and the server backend. The NFC capable mobile phone acts as a forwarding device and needs to connect the IC to the backend, so that all required information can be exchanged.

This means that all command APDUs which normally are sent from the reader or terminal to the IC, need to be implemented in the server backend and sent to the IC via the mobile phone. The following sections will focus in detail on the selection of the right MIFARE DESFire commands to make a secure end-to-end communication possible.

3.1 MIFARE DESFire features and functionalities that are recommended for OTA

For all kind of ongoing communication, it is very important to ensure the secure exchange of data between the MIFARE DESFire EV3 or EV2 IC, the mobile phone and the server backend. For both offline (MIFARE DESFire IC ← → mobile phone) and online (mobile phone ← → server backend) it needs to be ensured, that no real user data leaks or can be retrieved somehow.

Recommendations regarding usage of features and functionalities:

- Authentication
  - For authentication, it is recommended to use Cmd.AuthenticateAES and the EV1 secure messaging or the new Cmd.AuthenticateEV2First and the EV2 secure messaging.
  - Both suggested authentication methods are based on AES and therefore guarantee the strongest possible data encryption during the ongoing transaction.
  - The new Cmd.AuthenticateEV2First additionally improves the command binding inside one transaction through a transaction identifier, making it even harder to try any kind of attacks.

- Data integrity and data encryption
  - When accessing and exchanging data it is highly recommended to ensure the data integrity by using MACs / CMACs and at least the offered communication mode CommMode.MACed.
  - To ensure confidentiality during data exchange, it is highly recommended to add data encryption by using the communication mode CommMode.Full. This ensures the full data encryption, including MACing.
  - Transferring data in CommMode.Plain without any protection is highly not recommended.
  - The communication mode can be set for each file on the MIFARE DESFire IC individually which offers full flexibility and customizability.

- Transaction MAC (TMAC)

- In order to ensure that all executed commands really reached the card and were executed successfully, the Transaction MAC feature of MIFARE DESFire EV3 or EV2 can be used.
- By generating a MAC over the full transaction (all involved commands) on the IC, the backend can re-generate this MAC and verify if everything was executed as expected.
- Using the TMAC ensures
  - that all the commands were really transmitted between the server backend and the IC
  - that there was no man-in-the-middle or someone malicious trying to manipulate the communication
  - that there was no command inserted or replayed

- File Access Rights
  - MIFARE DESFire offers the flexibility to set the access rights for each access condition (read, write, read/write, change configuration). Each access right is associated with either a key or it can be set to free or never access.
  - MIFARE DESFire EV3 and EV2 offer a new feature, the multiple access condition sets of file access rights. Each application can have up to 8 access condition sets, with their own access rights associated with any key in the application. This offers the possibility to e.g. associate one set of keys for card interaction to the real terminal where the IC will be used, and separate keys for reading and writing to the server backend, which will be only used for the top-up transaction.
  - Using of multiple file access condition sets ensures the protection of the terminal keys, as only the keys which are needed for the OTA top-up need to be stored in the server backend.
  - A simple example for multiple file access condition sets is illustrated in Table 1.

Table 1. Example of multiple file access condition sets for OTA

Access Right	Key number used on Terminal	Key number used on Server
	Access Condition Set 1	Access Condition Set 2
Read	0x01	0x04
Write	0x02	0x04
Read / Write	0x02	0x04
Change Configuration	0x03	0x03

3.2 Timing efficient implementation of OTA with MIFARE DESFire

When implementing an OTA top-up solution for MIFARE DESFire, it is important to think about the application structure in detail and to use an optimized set of commands.

As all the APDUs need to be transferred from the server backend via the mobile phone to the IC and then back from the IC via the mobile phone to the server backend, there will be necessary a delay introduced. Depending on the network connection, the time that is needed to exchange a command-response pair between the server backend and the IC can be significantly larger than exchanging a command-response pair offline between a reader terminal and the IC.

Recommendations regarding timing efficiency of the OTA transaction:

- Authentications

If possible, reduce the number of needed authentications to a minimum. The less authentications are needed, the faster the overall transaction will be.

- File access rights

Assign meaningful file access rights to all files which need to be accessed from the server backend. E.g. only one key (and therefore also only one authentication) is needed in the backend to do most file manipulations.

- Value file

Use the value file for storing and topping up a numerical value. Don`t store values in normal standard data files or backup data files. MIFARE DESFire offers a set of value file operation commands (e.g. Cmd.GetValue, Cmd.Credit, Cmd.Debit) which make it very easy to access and modify numerical values, without the need to issue the standard read and write commands.

- Big frame size support

When reading or writing large amount of data, configure the IC to support bigger frame sizes during the pre-personalization. This reduces the needed command-response pairs when exchanging large chunks of data.

- Cyclic record file

For any logging purposes, use cyclic record files rather than linear record files. The benefit of using cyclic record files is the automatic overwrite of the oldest record entries, so an additional command which would be needed for erasing the file content, is not needed. However, be careful not to overwrite unsaved log entries!

3.3 Example OTA top-up application structure for MIFARE DESFire

In the next paragraphs, an example application representing a stored value application will be illustrated, and its usage in combination with OTA top-up will be explained.

The application structure and content are only exemplary and completely dynamically to design, depending on the system requirements.

In Figure 2 the example application which will be used for a detailed analysis in this chapter is depicted.

The application could have any Application ID that is used at the system operator, however for this example we will use the Application ID (AID) = 0x014499.

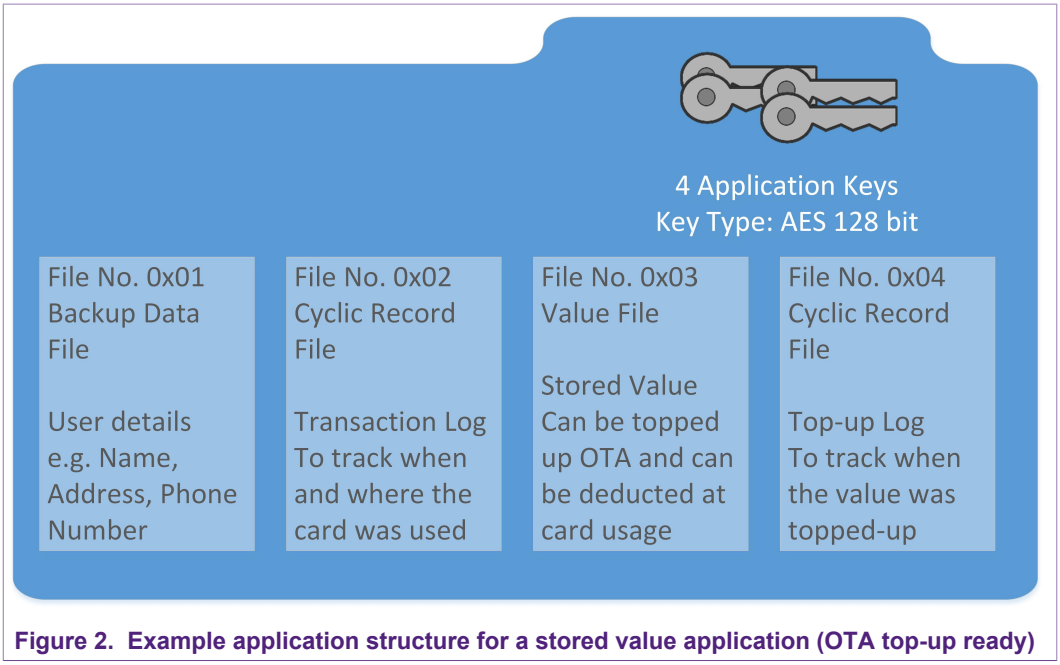

Figure 2. Example application structure for a stored value application (OTA top-up ready)

In this example application with AID = 0x014499, the following 4 files are used:

- File No 0x01 – Backup Data File
  - Stores some general information of the card header
  - File Size: 128 bytes
- File No 0x02 – Cyclic Record File
  - Stores the transaction logs. One log entry contains the following parts:
    - Usage location / Reader ID (4 byte)
    - Usage timestamp / date (8 byte)
    - Deducted value during usage (4 byte)
  - Record Size: 16 bytes, maximum 10 records per file
- File No 0x03 – Value File
  - Stores the currently available value as well as lower and upper limit
    - Current value (4 byte)
    - Minimum value (4 byte)

  - Maximum value (4 byte)
- File No 0x04 – Cyclic Record File
  - Store the top-up history log. One log entry contains the following parts:
    - Top-up timestamp / date (8 byte)
    - Top-up value (4 bytes)
  - Record Size: 16 bytes, maximum 10 records per file

In this example application, the following 4 keys are used. An overview of the access rights of the single files and the needed authentication keys is also depicted in Figure 3.

- Key No 0x00 – Application Master Key
  - Not diversified. Administration of the application, not used to access the application data, only used for configuring application and file settings.
- Key No 0x01 – Application Read Key
  - Diversified. Read access to all files inside the application and write access to the transaction log.
- Key No 0x02 – Application Write Key
  - Diversified. Write access to the user datafiles.
- Key No 0x03 – Application Top-up Key
  - Diversified. Read and write access to the value file and the top-up transaction log file.

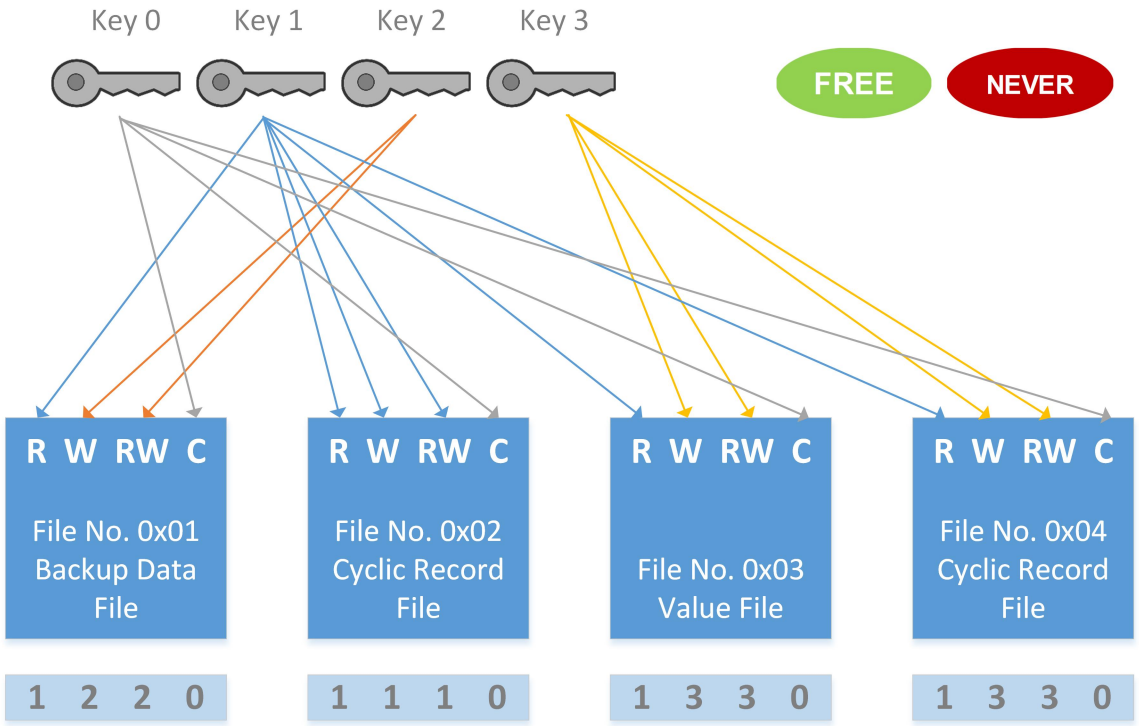

Figure 3. Example access rights for a stored value application (OTA top-up ready)

3.3.1 Command sequence

The complete sequence of commands that is needed for the following two scenarios is discussed:

- Top-up command sequence that is needed for topping up the value that is stored on the card over-the-air via a corresponding server backend (illustrated in Figure 4).
- Use the card at a transport reader or any other kind of terminal where an amount of the stored value is deducted (illustrated in Figure 6).

Both transactions only show how a potential top-up as well as usage of a value based card could look like in the field. There is definitely no need to stick to the given command sequences, but they rather shall demonstrate how a simple OTA transaction could be realized.

In Figure 4 the transaction which does the OTA top-up of a value that is stored on the card, is depicted. All the commands which are shown are triggered from the server backend and sent to the IC via the mobile phone. For the full transaction, the IC needs to remain tapped to the mobile phone, so that the connection between IC and server backend remains established. The mobile phone cannot read or modify any of the exchanged data during reading / writing as everything is transmitted in an encrypted way, using the session keys that are generated after a successful authentication.

Additionally, Figure 5 shows the involved system parts and the devices which are involved in the communication.

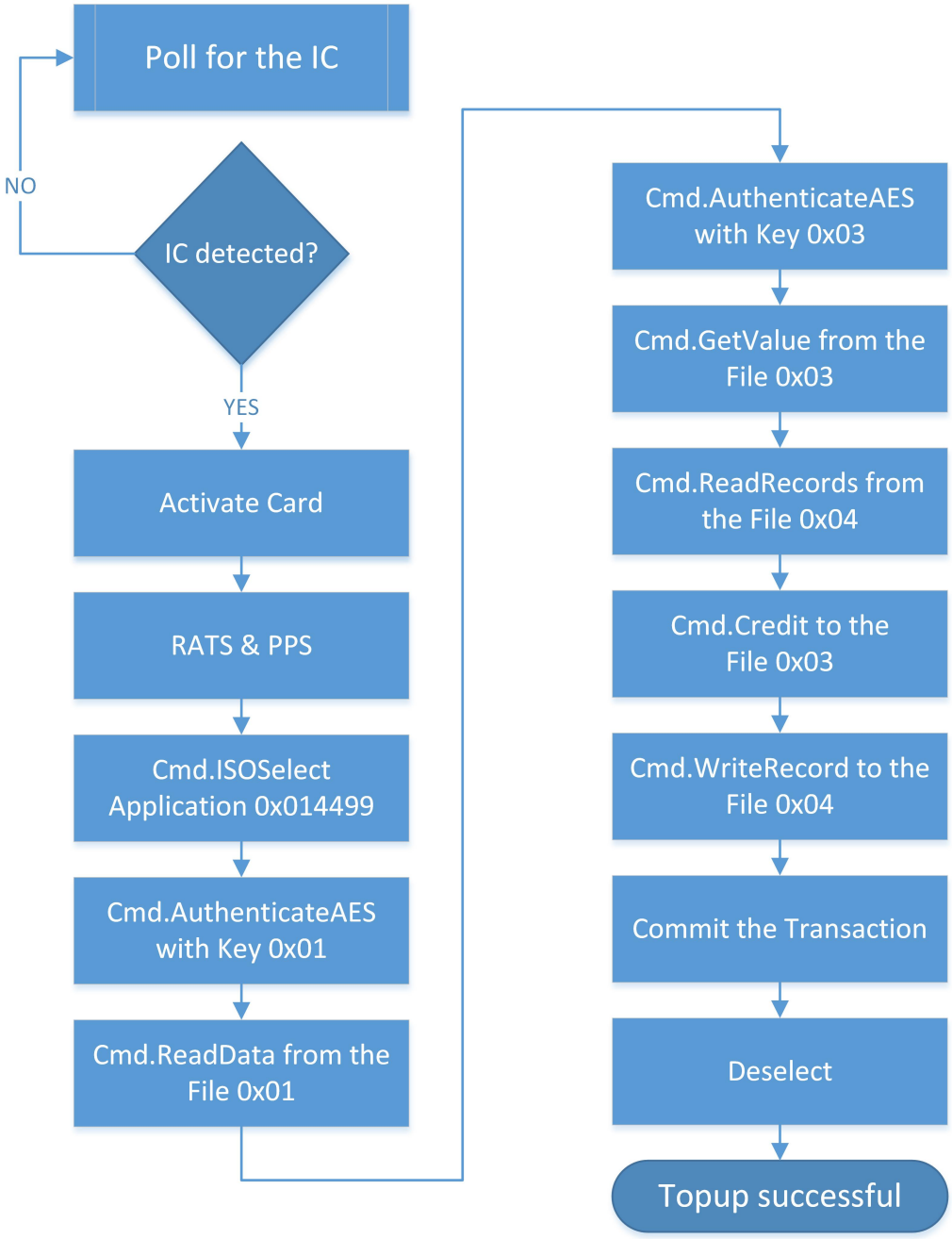

Figure 4. Top-up command sequence for the example stored value application

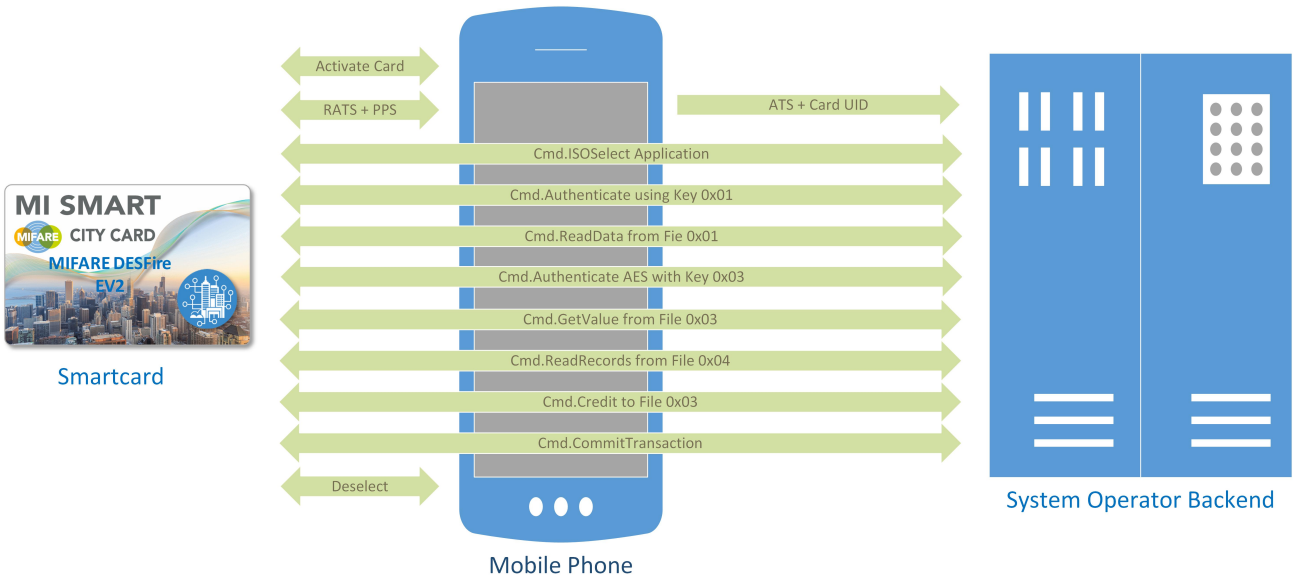

Figure 5. Top-up command sequence for the example stored value application showing the main involved parties / devices

In Figure 6 the transaction which actually uses some value that is stored on the card and deducts it, is depicted. All the commands which are shown are triggered from a reader terminal that can be either online or offline. The commands are sent directly from the reader to the IC, without the need to have the mobile phone present.

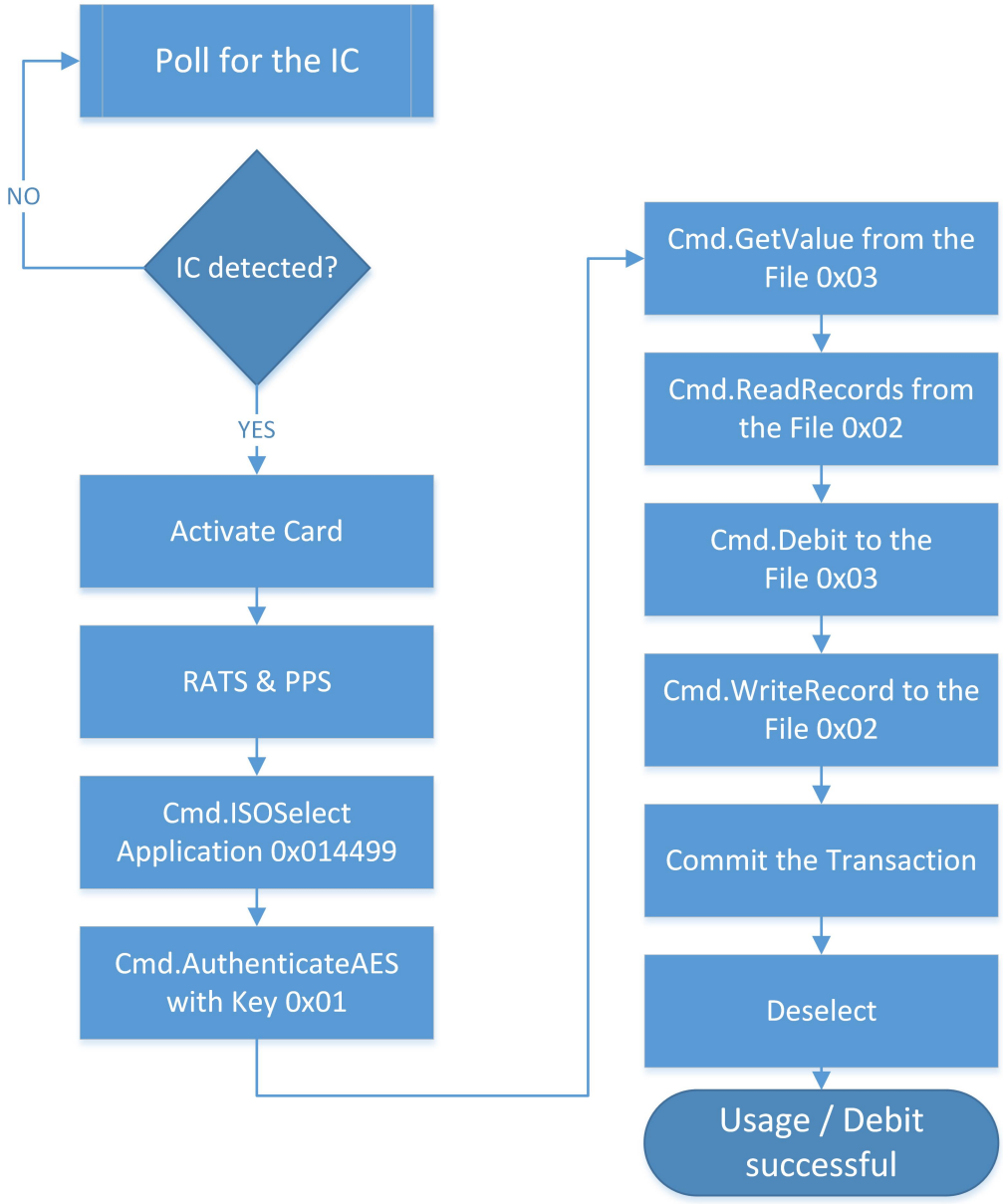

Figure 6. Top-up command sequence (value deduction) for the example stored value application

4 OTA in combination with MIFARE Plus

For MIFARE Plus EV1 and EV2, a dedicated mode exists to allow mobile OTA value top-up in a legacy infrastructure. This mode is called SL1SL3Mix Mode, and basically combines the features of SL1 and SL3 together. It allows on the one hand to operate a user memory sector on the card or the entire card in the MIFARE Classic backwards compatibility mode (SL1/Crypto-1), whereby the value top-up over-the-air (OTA) is secured by AES authentication with a key length of 128-bit.

4.1 Configuration of MIFARE Plus

To activate the SL1SL3Mix Mode for the MIFARE Plus EV1 and MIFARE Plus EV2 the following step has to be performed. The switch is done using a multiple key authentication, called AuthenticateSectorSwitch, addressing the L1L3MixSectorSwitchKey(90 07h), and the AESSectorKeyB of the targeted sector. During the AuthenticateSectorSwitch, the targeted sectors are addressed by their AESSectorKeyB. These keys are also used during the authentication. Details on how to switch user memory sectors into SL1SL3Mix Mode can be found in [2] .

Once the targeted sectors are switched to SL1SL3Mix Mode, those sectors can be used using the SL1 command set, as well as the SL3 command set.

IMPORTANT NOTE FOR MIFARE PLUS EV1: Both command sets use the same access conditions defined in the sector trailer of the targeted sector. This means, that it is not possible to restrict read / write to a command set of a specific security level. The only way restrictions can be applied, is by using, e.g., KeyA only for read/decrement, and KeyB only for write/increment operations. A random value for KeyB in the sector trailer would then restrict writing/increment operations to an authentication with the AES-KeyB of the targeted sector. A feature to avoid this is added in MIFARE Plus EV2, and described in section Section 4.3.

From this point onwards, OTA top-up works analogous to MIFARE DESFire.

4.2 Example Configuration of one MIFARE Plus sector

In this example, all the steps needed to configure a sector of a MIFARE Plus EV1 / EV2 card from SL1 to SL1SL3Mix Mode. This can be applied in any installation working on SL1.

Starting from a MIFARE Plus EV1 / EV2 card in SL1, with all necessary keys already written during personalization (esp. key no. 0x9007, which is the SL1SL3SectorSwitchKey), first the access conditions need to be updated accordingly.

Also, to enable SL1SL3Mix Mode sector switching, it must be ensured that byte 4 in the MFPConfigurationBlock (0xB000) is set to 0xAA, to allow the AuthenticateSectorSwitch command.

In this example, we want to

- allow write/increment with KeyB **only**
- allow read/decrement with KeyA|B

on all data blocks in the targeted sector, and

- never allow any key read
- key and AC write with KeyB **only**
- AC read with KeyA|B

for the Sector Trailer.

KeyA is now the key used for read/decrement, and KeyB for write and increment.

Note: In the case that value top-up should only be allowed using the AES-128 authentication, KeyB could be set as random value in the Sector Trailer.

This results in a Sector Trailer for the targeted sector as:

Table 2. Example - Sector Trailer

KeyA	Access conditions	KeyB
AA BB CC DD EE FF	08 77 8F 69	UU VV WW XX YY ZZ

The Sector Trailer can be written with a standard WRITE command in SL1. Once the Sector Trailer is written, the sector can now be switched to SL1SL3Mix Mode, using the AuthenticateSectorSwitch command. The Keys targeted for this command are the SL1SL3SectorSwitchKey (90 07h), and the KeyB of the targeted Sector. With one AuthenticateSectorSwitch command, N sectors can be switched to SL1SL3Mix Mode, where N is the number of sectors available on the MIFARE Plus card.

The block number of the AES sector KeyB can be obtained using following formula:

KeyB = 0x4000 + (sector number * 2) + 1)

Detailed information on the command can be found in [2] .

Once the sector is switched into SL1SL3Mix Mode, it is ready to be used with AES-128 authentication for OTA top-up. If the key values have not been configured earlier, it can be done now[1].

4.3 Additions in MIFARE Plus EV2

In MIFARE Plus EV2, two new features that improve the SL1SL3Mix Mode have been added:

- SL1UpdateRestriction
- ValuePairConfiguration

Both new features help to even more restrict the usage of CRYPTO-1 based access on data and value blocks. The configuration of both features is realized by 2 new VCSystemData configuration blocks available on MIFARE Plus EV2. Both features overrule the access conditions in the sector trailer.

SL1UpdateRestriction:

This setting can be activated per block for each data/value block, as well as for sector trailers. Once activated, the update (Write/Transfer) of the blocks content is disabled when using a SL1 authentication. This allowes a configuration, where the cards contents can be read in SL1, but not written or changed. A change of the block content is only possible with a secure SL3 authentication using AES128 keys.

ValuePairConfiguration:

The ValuePairConfiguration allows configure exceptions from the SL1UpdateRestriction in a way, that pairs of value blocks (source and target) can be defined, for which the

1 Note: In case the KeySystemDataWrite has not been explicitly allowed in the MFPConfigurationBlock, it is **only** possible to change the AESSectorKeys after switching a sector to SL1SL3Mix Mode. Also, the Access conditions also apply here for changing the AES-128 Keys. For this example, in Order to change KeyA, one needs to be authenticated with KeyB.

transfer command is allowed, even if the SL1UpdateRestriction is switched on for a block which is part of this pair. Additional to that, the transfer operation can be restricted to decrement only, by setting a flag in the ValuePairConfigurationBlock.

If all of above configurations are applied, the value in the resulting block can only be decremented in SL1, but never incremented. Also, the decrement can only come from a block that is allowed (this could also be the block itself). Only by using the SL3 AES128 authentication, the content of the value block can be written/incremented.

More details can be found in the MIFARE Plus EV2 datasheet

5 References

1. **Product Data Sheet —** MIFARE DESFire EV3 contactless multi-application IC, document number 4870xx, available in NXP DocStore
2. **Product Data Sheet –** MIFARE Plus EV2 mainstream contactless smart card IC, document number 5223xx, available in NXP DocStore

6 Legal information

6.1 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.

6.2 Disclaimers

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

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

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

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

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

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

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

Security — While NXP Semiconductors has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP Semiconductors accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

6.3 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.

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

NXP — 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: 2 July 2020
Document identifier: AN12113
Document number: 456712