MIFARE DESFIRE EV3C

TECHNICAL DETAILS OF THE EV3C PRODUCT VERSION

SANDRA FUCHS
CUSTOMER APPLICATION SUPPORT
JUNE 2020



COMPANY CONFIDENTIAL

NXP

SECURE CONNECTIONS
FOR A SMARTER WORLD

ALMOST 2 DECADES OF EVOLUTION TO MIFARE DESFire EV3



MIFARE DESFire EV3



MIFARE DESFire EV2

2020

NEW
- Faster performance
- AppXplorer support with preloaded keys
- SDM (NTAG DNA)



MIFARE DESFire EV1

2016

NEW
- AppXplorer support
- Common Criteria EAL5+



MIFARE DESFire

2008

NEW
- Open standard AES crypto
- Common Criteria EAL4+

2002

NEW
- Open standard 3DES crypto
- Flexible file system Multi-application mgmt

	MIFARE DESFire EV1	MIFARE DESFire EV2	MIFARE DESFire EV3	MIFARE DESFire EV3C
ISO/IEC 14443 A 1-4	Yes	Yes	Yes	Yes
ISO/IEC 7816-4 support	extended	extended	extended	extended
EEPROM data memory	2/4/8KB	2/4/8/16/32KB	2/4/8KB	2/4/8KB 1KB (MFC)
Flexible file structure	Yes	Yes	Yes	Yes
NFC Forum Tag Type 4	Yes	Yes	Yes	Yes
Unique ID	7B UID or 4B RID	7B UID or 4B RID	7B UID or 4B RID	7B UID or 4B RID and 4B NUID
Number of applications	28	unlimited	unlimited	unlimited
Number of files per app	32	32	32	32
High data rates support	Up to 848 Kbit/s	Up to 848 Kbit/s	Up to 848 Kbit/s	Up to 848 Kbit/s
Crypto algorithms support	DES/2K3DES/ 3K3DES/ AES128	DES/2K3DES/ 3K3DES/ AES128	DES/2K3DES/ 3K3DES/ AES128	DES/2K3DES/ 3K3DES/ AES128/ Crypto-1
CC certification (HW+SW)	EAL 4+	EAL 5+	EAL 5+	EAL 5+
MIsmartApp (AppXplorer)	-	Yes	Yes, preloaded	Yes, preloaded
Secure Dynamic Message	-	-	NTAG DNA	NTAG DNA
Transaction MAC	-	Yes	Yes	Yes
Multiple keysets per app	-	Up to 16 keysets	Up to 16 keysets	Up to 16 keysets
Multiple file access rights	-	Up to 8 keys	Up to 8 keys	Up to 8 keys
Inter-app files sharing	-	Yes	Yes	Yes
Transaction timer	-	-	Yes	Yes
Virtual Card Architecture	-	Yes	Yes	Yes
Proximity Check	-	Yes	Yes	Yes
Delivery types	Wafer, MOA4 & MOA8	Wafer, MOA4 & MOB6	Wafer, MOA4 & MOA8	Wafer, MOA4 & MOA8

MIFARE DESFire EV3 – 2 product variants! New Features!

MIFARE DESFire EV3



MIFARE DESFire EV3C



- Secure Dynamic Messaging (SDM as known from NTAG DNA)
- Delegated Application Management support extended to 2 sets of DAM Keys (Card issuer owned DAM Keys, NXP owned DAM Keys)
- Transaction Timer to limit the transaction time and to avoid potential MitM attacks
- Seamless drop-in replacement for existing DESFire infrastructures
- Increased operating distance (~15% compared to MF DESFire EV1)
- Increased transaction speed (up to 1.6x faster compared to MF DESFire EV1)

- 1KB MIFARE Classic memory
- MIFARE Classic to DESFire data mapping
- Crypto-1 support
- Support of ISO/IEC 14443-3 command exchange for MF Classic and support of ISO/IEC 14443-4 command exchange for MF DESFire

MIFARE DESFire EV3C with MIFARE Classic migration support

One card – supporting two card protocols

Key features

- MIFARE DESFire & Classic on one chip
- MIFARE Classic 1KB memory
- Mapping Classic blocks to DESFire files
- MIFARE Classic keys & AC management via DESFire
- Dual-mode operation on same data
- Dual UID supported (7BUID + 4BNUID)
- MIFARE Classic “Kill” option supported
- MIFARE Classic Data & Value operation restriction

Migration Scenario (steps)

1. Mapping Classic card data to DESFire files
2. Deploy new DESFire EV3C on existing infra
3. Upgrading TVM with DESFire to improve security
4. Add mobile Top-Up service with DESFire protocol
5. Add new readers with DESFire support
6. Progressively upgrade existing readers to DESFire
7. Kill Classic on DF EV3C when upgrading is completed
8. Turn off Classic supports on readers ASAP



MIFARE DESFIRE EV3C
MIFARE CLASSIC FUNCTIONALITY

SECURE CONNECTIONS
FOR A SMARTER WORLD

MIFARE DESFire EV3C – GetVersion information

- MIFARE DESFire EV3C supports additionally 1KB MIFARE Classic memory and functionality, next to the MIFARE DESFire functionality on the same dvice
- MIFARE Classic 1KB functionality is compliant to the standalone MIFARE Classic 1KB product

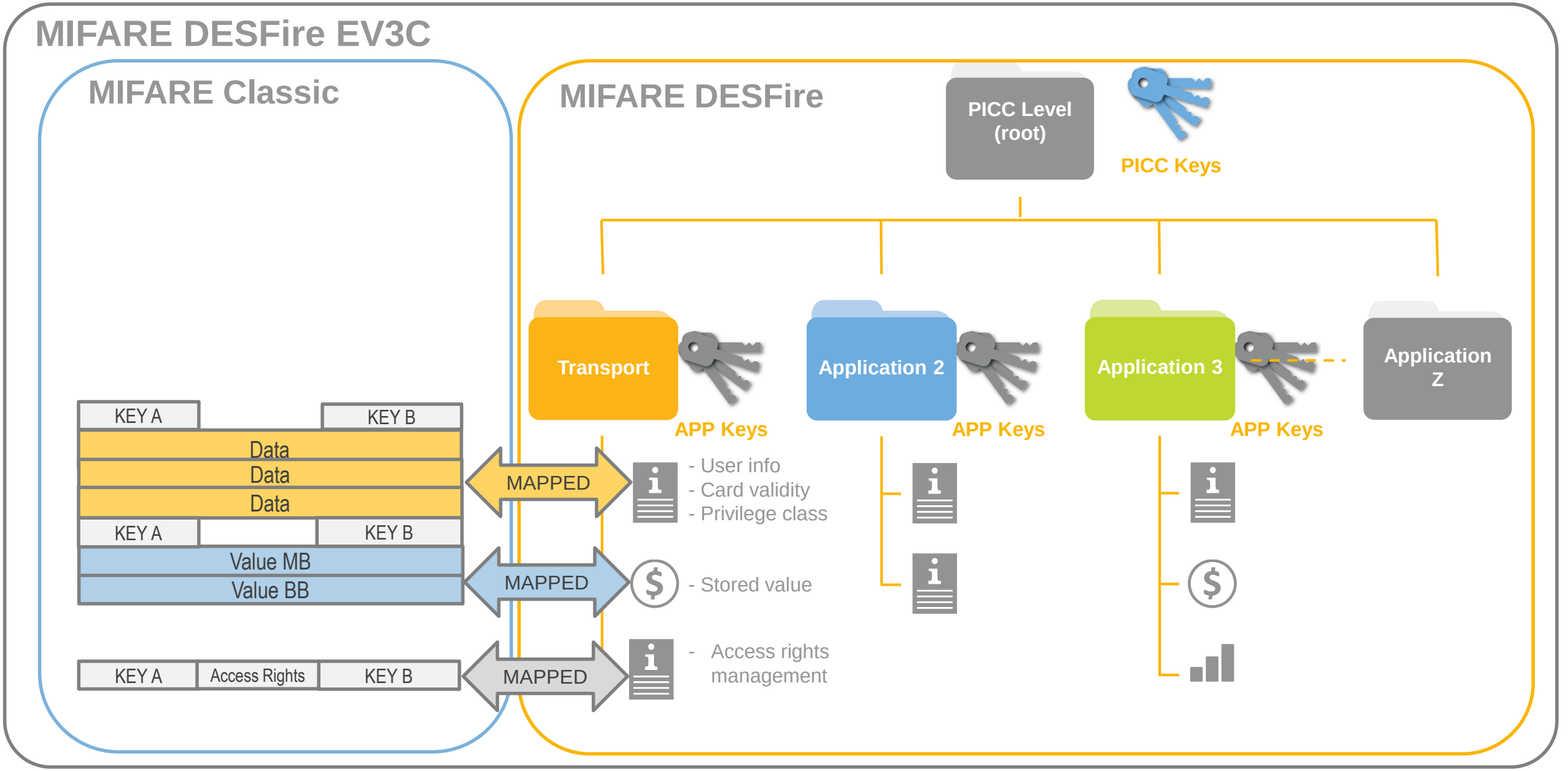
- The support for MF Classic functionality will be reflected in the SWSubType in the GetVersion response
  - 0x01 reflects MIFARE DESFire standalone (without MIFARE Classic support)
  - 0x02 reflects MIFARE DESFire with MIFARE Classic support

MIFARE Classic Functionality

- MIFARE DESFire EV3C supports MIFARE Classic 1KB functionality next to MF DESFire functionality on the same device
  - MIFARE Classic memory can be accessed using ISO/IEC 14443-3 commands
  - MIFARE DESFire EV3C memory can be accessed using ISO/IEC 14443-4 commands

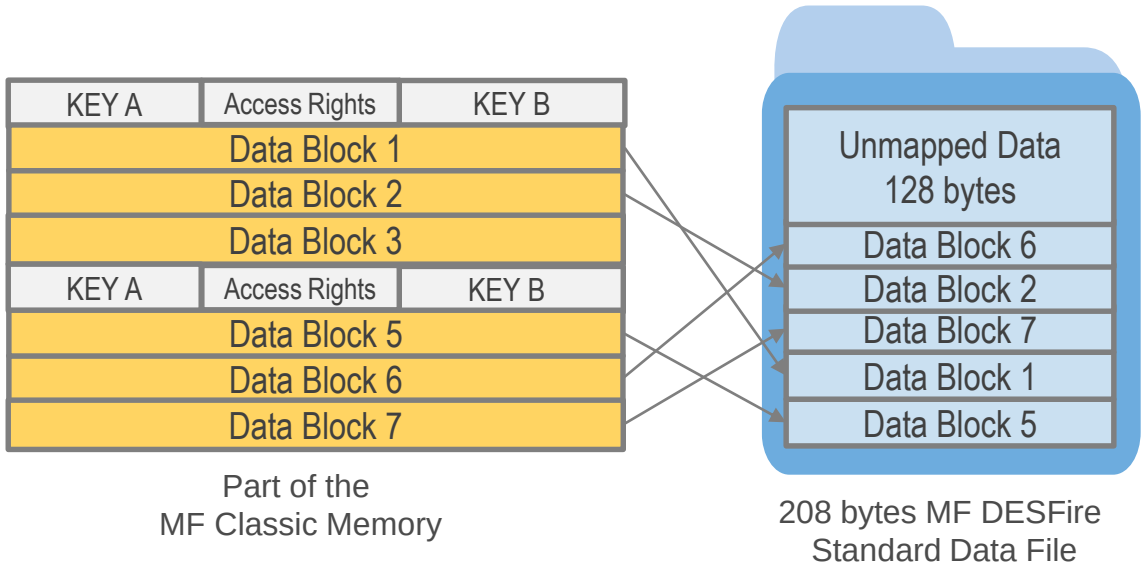
- Memory mapping from MIFARE Classic blocks to the MIFARE DESFire file system is possible
  - Mapping of MF Classic data blocks, value blocks and sector trailers
  - Mapped data can be accessed via both MF Classic and MF DESFire interface (using either 14443-3 or 14443-4 commands)
  - Mapped data is kept in synch between both memory partitions
  - MIFARE Classic update restrictions can be set (e.g. specific data update / change only via the MF DESFire interface, but reading still via MF Classic interface)

Mapping MIFARE Classic Blocks to MIFARE DESFire Files



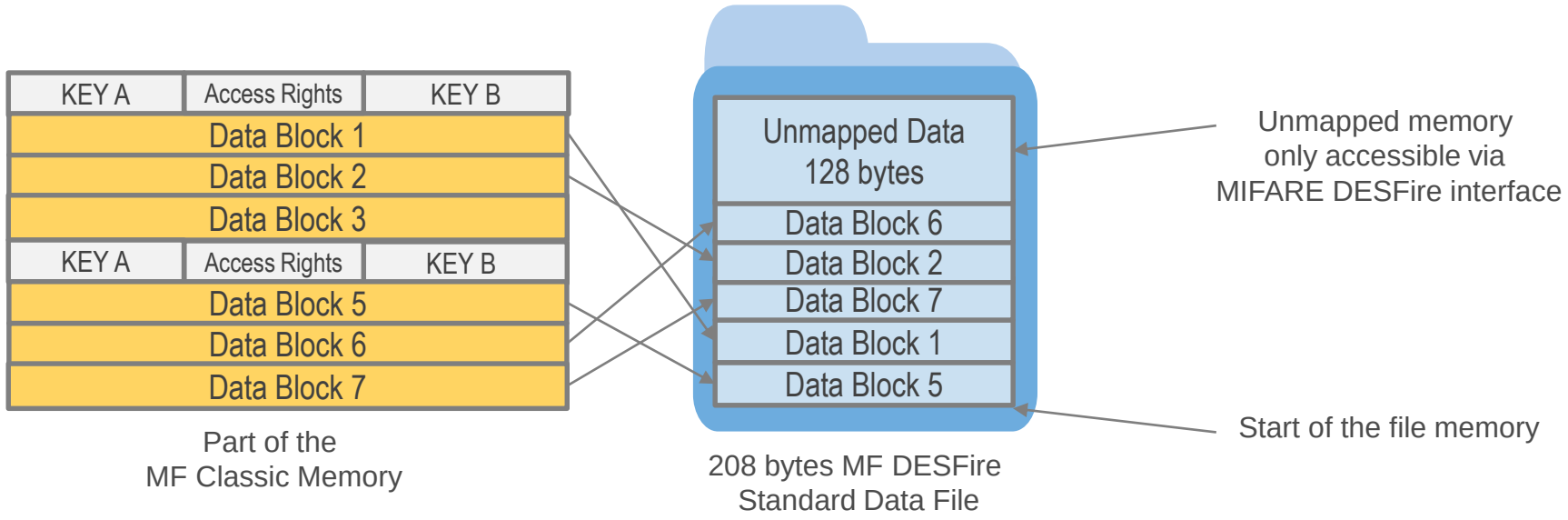
MIFARE Classic Functionality – Memory Mapping

- Memory mapping from MIFARE Classic blocks to the MIFARE DESFire file system
  - MF Classic blocks can be mapped to
    - MF DESFire Standard Data / Backup Data / Value Files
  - Data is always mapped to the beginning of the MF DESFire file
  - Several MF Classic blocks can be mapped in any order to one or more MIFARE DESFire files
  - If the file is bigger than the memory that`s required by the mapped data, the remaining (unmapped) data can only be accessed via the MF DESFire interface



MIFARE Classic Functionality – Memory Mapping

- Memory mapping from MIFARE Classic blocks to the MIFARE DESFire file system
  - Once data was mapped, updates to the MF Classic blocs or to the MF DESFire file data will be reflected on both the MF Classic and the MF DESFire interfaces
    - Independent on which interface the update was done
    - The data will be kept in synch between both interfaces



Memory Mapping Enablement

- Memory mapping must be enabled at application level, already during MIFARE DESFire application creation
  - On application level using the commands
    - CreateApplication
    - CreateDelegatedApplication
  - Parameter to be set → KeySettings3 Bit3 set to 1, indicating MFC Mapping enabled

- Afterwards, memory mapping needs to be enabled for the specific files, that shall contain mapped data, during the file creation
  - On file level using the commands
    - CreateStandardDataFile
    - CreateBackupDataFile
    - CreateValueFile
  - Parameter to be set → FileOption Bit5 set to 1, indicating MFC Mapping enabled

Memory Mapping Enablement and Configuration

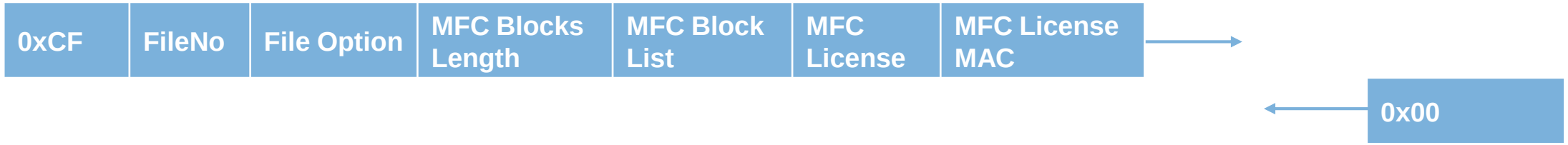
- The command that`s needed to establish the memory mapping between MIFARE Classic and MIFARE DESFire is CreateMFCMapping
  - Mapped blocks are specified as MFCBlockList

- All MF Classic blocks that are mentioned in the MFCBlockList need to be covered by the MFCLicense
  - For mentioned blocks, BlockOptionX bit0 needs to be set
  - For Sector Trailers, also BlockOptionX bit3-5 will be checked

- The MFCLicense and MFCLicense MAC is needed
  - So that the MF DESFire application owner can prove that he is authorized to create the mapping
  - MFCLicenseMAC is generated by the MIFARE Classic application owner, as he is controlling the access to the MIFARE Classic memory section
  - CreateMFCMapping command will fail, in case the MAC verification fails

Memory Mapping
Cmd.CreateMFCMapping

PCD PICC



- FileNo: 1 byte long unique MF DESFire File ID inside the application
- FileOption: MIFARE Classic mapping options
  - Bit5-3: Updating Key A / Key B / Access conditions enabled
  - Bit2: Enabling Cmd.RestoreTransfer (for Value Files)
  - Bit0: Mapping Data Block or a Sector Trailer
- MFC Blocks Length: Amount of MFC blocks to be mapped to the MF DESFire File (for Value File only 1 MFC Value block can be mapped)
- MFCBlockList: List of MFC block numbers that shall be mapped to the MF DESFire File
- MFCLicense: „License“ generated by the MF Classic memory owner that allows / restricts the mapping of MF Classic blocks
  - Can be used to restrict the MF DESFire application owner to access only blocks that he is allowed to
- MFCLicenseMAC: CMAC calculated by the MF Classic memory owner including the MFCLicense and affected MFC sector keys

MFC License

- A license through which MIFARE Classic mapping rights and update rights can be defined
  - E.g. a MIFARE Classic application owner can restrict the mapping rights towards the MIFARE DESFire application owner

MFCLicense = BlockCount || *BlockNr1* || *BlockOption1* || *BlockNr2* || *BlockOption2* || … || *BlockNrN* || *BlockOptionN*

Block Count	Defining the number of blocks covered by the MFCLicense
BlockNrX	A specific block for which the following BlockOption applies for
BlockOptionX	Defining the granted rights towards the MF DESFire interface

MFC License – Block Options

- Restrict MFC Update operation allowed / not allowed
  - Defines whether the RestrictMFCUpdate command is allowed for the specified block or not
- Update of MFC Key A
  - Can only be set for sector trailers
  - Defines whether MFC Key A can be updated via the MF DESFire interface
- Update of MFC Access Conditions in Sector Trailer
  - Can only be set for sector trailers
  - Defines wheter MFC access conditions can be updated via the MF DESFire interface
- Update of MFC Key B
  - Can only be set for sector trailers
  - Defines whether MFC Key B can be updated via the MF DESFire interface
- Mapping MIFARE Classic Block content
  - Defines whether the specified block can be mapped or not

MFC License MAC

- A CMAC calculated by the MIFARE Classic application owner
  - CMAC calculation as per EV2 secure messaging using the zero byte IV and the MFCLicenseMACKey

- Needed by the MIFARE DESFire application owner to create the MF Classic to MF DESFire memory mapping
  - Mandatory needed for the Cmd.CreateMFCMapping
  - Prevents unauthorized access to MIFARE Classic blocks

$$MFCLicenseMAC = CMAC(0x01 \,||\, MFCLicense \,||\, MFCSectorSecrets)$$

MFCLicense … As defined before (specifying MF Classic mapping and update rights)

MFCSectorSecrets … MIFARE Classic KeyA (and optionally KeyB) of the sector

MFC License MAC Key – Enablement of the Key

- MFCLicenseMACKey is needed for calculating the MFCLicenseMAC that's needed to execute the CreateMFCMapping command
- It´s disabled by default
- Needs to be enabled using the ChangeKey or ChangeKeyEV2 command
- Is always of keytype AES
- It`s key number 0x32 on PICC level
- This key is not available for authentication, just for MFCLicenceMAC verification on the card



MIFARE Classic Functionality – 4byte NUID supported

- MIFARE DESFire EV3C can be set to support a 4byte NUID
  - In this case the 4b NUID will be applied for card activation
  - It will be written to Block0 of the MF Classic memory
  - The real 7byte unique VCUID stays still intact and can be retrieved on the MF DESFire interface via the Cmd.GetCardUID
  - The real 7byte unique VCUID will still be used for specific MF DESFire features
    - Transaction MAC
    - Virtual Card Selection
    - Secure Dynamic Messaging and Mirroring

Setting a 4byte NUID on the card

Cmd.SetConfiguration



- Option: 0x00 – PICC Configuration update
- Data: Different SetConfiguration specific options to choose from
  - Bit6: Set to 1 to enable the 4byte NUID for the IC

After enabling the 4byte NUID, this UID will be…

- Used during ISO/IEC 14443-3 activation
- Reflected in the ATQA
- Reflected in MFC Block 0 (Manufacturer block in the MIFARE Classic memory section)

Restricting MIFARE Classic Updates

- Additionally to the MF Classic access conditions configuration that can be mapped and managed via the MF DESFire interface, it`s possible to further restrict MF Classic access rights
  - MIFARE Classic blocks can be restricted from being updated over the MF Classic interface

- Command to be used for this is Cmd.RestrictMFCUpdate

- Possible to allow any update capabilities to the MIFARE DESFire interface only, without the need to permanently lock the MF Classic blocks
  - Might be done after the first personalization was done via the MF Classic interface
  - Allows configurations where MF Classic can only execute Read or Decrement operations whereas the high secure MF DESFire interface can execute also Write and Increment operations
  - Possibility to also only restrict Write operations, but still accept Value Transfer operations by configuring so-called MFCValuePairs

Restricting MIFARE Classic Updates

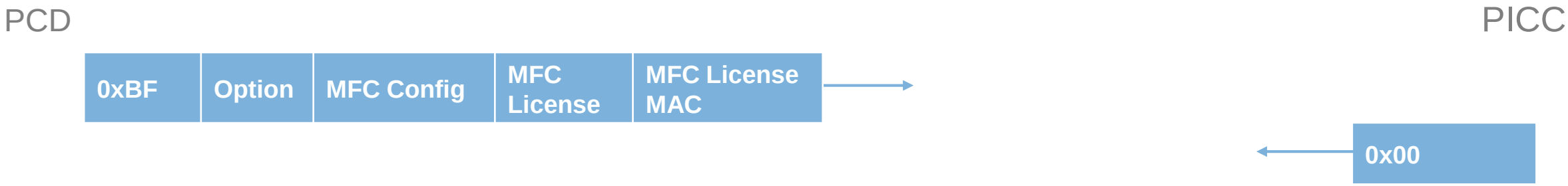
Cmd.RestrictMFCUpdate

- Option 0x00 ***MFCRestrictConfiguration***
  - MFC_Write and MFC_Transfer commands targeting the restricted block will be rejected independent of the configured MFC access conditions in the sector trailer

- Option 0x01 ***MFCValuePairConfiguration***
  - MFC_Transfer commands targeting the restricted block will still be allowed, if MFCValuePairs are configured. However the source block is restricted to be either the complementary block of the value pair, or the targeted block itself.
  - It´s possible to restrict value pair operations to allow decrement only

Restricting MIFARE Classic Updates

Cmd.RestrictMFCUpdate



- Option: MIFARE Classic update restriction option
  - 0x00: Configure MFCRestrictConfiguration
  - 0x01: Configure MFCValuePairs
- MFCConfig:
  - For option 0x00: 8 bytes of MFCRestrictConfiguration
  - For option 0x01: dynamic list of MFCValuePairConfiguration (between 1 and 7 value pairs can be configured at the same time)
- MFCLicense: „License“ generated by the MF Classic memory owner that allows / restricts the mapping of MF Classic blocks
  - Can be used to restrict the MF DESFire application owner to access only blocks that he is allowed to
- MFCLicenseMAC: CMAC calculated by the MF Classic memory owner including the MFCLicense and affected MFC sector keys

MFC Value Pair Configuration

- Up to 7 value pairs can be configured on the card (in form of a bitfield)

- These 7 pairs are shared across all MF DESFire application
  - If two blocks are configured to form a value pair, the source block of the commands Restore, Increment, Decrement needs to be the complementary block of the value pair (or the targeted block itself), otherwise the Transfer command will fail
  - Each application can issue the command and configure value pairs, but maximum 7 pairs can be configured in total
  - When the command is executed again with the same application, old configured value pairs will be released and replaced by the new list
  - Application deletion will release all value pairs configured for the application

- Formatting the card will release the configured value pairs

Disabling MIFARE Classic – MFC Kill

- In case the MIFARE Classic functionality is not necessarily needed any more, and the infrastructure was upgraded to work with the MIFARE DESFire interface only, it`s recommended to disable the MIFARE Classic functionality

- Disabling the MIFARE Classic functionality means
  - MFC memory part will be not usable any more
  - MFC commands on ISO/IEC 14443-3 will not be accepted any more
  - MFC functionality can`t be enabled any more, MFC Kill is a one-time action that`s not reversable

- Disabling the MIFARE Classic functionality works with authentication using the MFCKillKey

MFC Kill Key – Enablement of the Key

- MFCKillKey is needed for permanently disabling the MIFARE Classic functionality and memory part
- It´s disabled by default
- Needs to be enabled using the ChangeKey or ChangeKeyEV2 command
- Is always of keytype AES
- It`s key number 0x31 on PICC level
- This key can be used for authentication using AuthenticateEV2First or AuthenticateEV2NonFirst to permanently kill the MIFARE Classic functionality





SECURE CONNECTIONS
FOR A SMARTER WORLD

NXP and the NXP logo are trademarks of NXP B.V. All other product or service names are the property of their respective owners. © 2018 NXP B.V.