Newland

SCANNING MADE SIMPLE

FM3281 NFC Description Guide

User Guide

Version: 1.0.0

Catalog

1. Introduction

Near Field Communication (NFC) is a new technology. Devices (such as mobile phones) using NFC technology can exchange data when they are close to each other. It evolved from integrating non-contact radio frequency identification (RFID) and interconnection technology. By integrating induction card reader, induction card, and point-to-point communication functions on a single chip, mobile terminals are used to realize mobile payment, electronic ticketing, access control, mobile identity identification, anti-counterfeiting and other applications.

Near-field communication is a kind of short-range wireless communication technology based on RFID technology. Like RFID, near-field communication information is transmitted through electromagnetic induction coupling in the wireless frequency part of the spectrum, but there is still a big difference between the two. The transmission range of near-field communication is smaller than that of RFID, whose transmission range can reach 0~1m. However, because of the unique signal attenuation technology adopted by near-field communication, compared with RFID, near-field communication has the characteristics of low cost, high bandwidth, and low energy consumption.

The main characteristics of near-field communication technology are as follows:

1) Wireless communication technology for short-range (within 10 cm) secure communication.
2) RF frequency: 13.56MHz.
3) RF compatible: ISO 14443, ISO 15693, Felica standard.
4) Data transfer speed：106kbit/s，212 kbit/s，424kbit/s.

2. Supported RF Standards

FM3281 can support four RF standards as below:

1、 ISO 14443-A
2、 ISO 14443-B
3、 ISO 15693
4、 Felica

3. Card types are supported

Basic ISO 14443-a, ISO 14443-B, ISO 15693, Felica protocol standard, there are many different functions and cards for different industries. The FM3281 supports the following types of cards

1、 Mifare series cards
2、 NTAG213/215/216 cards
3、 Ultralight/C/EV1 Card
4、 NTAG 42x DNA/TT Card
5、 ICODE2 card
6、 Felica card

3.1 Mifare Series Cards

Supports Mifare Desfire, Mifare Classic, mifare ultralight, Mifare Plus card types.

3.1.1 Mifare Desfire

The MIFARE DESFire product family provides highly secure microcontroller-based ICs based on global open standards for RF interfaces and encryption methods. The name DESFire indicates that DES, 2K3des, 3K3des, and AES hardware encryption engines are used to protect the transmitted data.

This series is ideal for solution developers and system operators to build reliable, interactive, and scalable contact-free solutions. MIFARE DESFire products can be seamlessly integrated into mobile scenarios and support multi-application smart card solutions for authentication, access control, loyalty and micropayment applications as well as transport ticketing facilities.

FM3281 supports Mifare Desfire EVx family, Mifare Desfire light,

Products	Explain	ISO/IEC	Security	Certification
Mifare Desfire Evx	High security IC for contact-free smart urban services.	ISO/IEC 14443 A 1-4&ISO/IEC 7816	DES/2K3DES/3K2DES/AES Encryption Algorithm	CC EAL5+
Mifare Desfire light	Safe, easy to integrate, cost-effective, contact-free, integrated circuit	ISO/IEC 14443 A 1-4&ISO/IEC 7816	AES 128-bit and LRP authentication and secure messaging	CC EAL4

3.1.2 Mifare Desfire EVx

	MIFARE DESFire EV3	MIFARE DESFire EV2	MIFARE DESFire EV1
ISO/IEC 14443 A 1-4	Have	Have	Have
Supports ISO/IEC 7816-4	Expanded	Expanded	Expanded
EEPROM data memory	2/4/8KB	2/4/8/16/32KB	2/4/8KB
Flexible file structures	Have	Have	Have
NFC Forum Class 4 Tags	Have	Have	Have
Unique ID	7 B UID or 4 B RID	7 B UID or 4 B RID	7 B UID or 4 B RID
Number of applications	Supported Memory Size	Supported Memory Size	28
Number of files per app	32	32	32
Data transfer rates supported	Up to 848 Kbit/s	Up to 848 Kbit/s	Up to 848 Kbit/s
Supported encryption algorithms	DES/2K3DES/ 3K3DES/ AES128	DES/2K3DES/ 3K3DES/ AES128	DES/2K3DES/ 3K3DES/ AES128
Delegated Application Management (Multiple Applications)	Yes, preloaded key	Have	-
SUN (Secure Unique NFC Message)	Yes, compatible with NTAG DNA	-	-

	MIFARE DESFire EV3	MIFARE DESFire EV2	MIFARE DESFire EV1
Transaction MAC per application	Have	Have	-
Multiple sets of keys per application	Up to 16 keys	Up to 16 keys	-
Multiple files can be accessed	Up to 8 keys	Up to 8 keys	-
File sharing between applications	Have	Have	
Transaction timer	Have	-	-
Virtual Card Architecture	Have	Have	-
Near-field verification	Have	Have	-
Transport type	Wafer, MOA4 and MOA8	Wafer, MOA4 and MOA6	Wafer, MOA4 and MOA8

Table 1

3.1.3 Mifare Desfire light

The MIFARE DESFire Light is a contactless IC designed for easy integration into new and existing systems. Its predefined file system and 640 bytes of total available memory (equivalent to 1 kB of MIFARE Classic) make it an excellent choice for single-application designs in a variety of use cases. The MIFARE DESFire Light is compatible with MIFare DESFire EV2.

MIFARE DESFire Light are designed for limited and extended use applications and include appropriate protection mechanisms to support trusted services. Depending on the use case, up to five AES 128-bit keys can be used to manage access, while secure messaging options enhance data and privacy protection. All hardware and software security features of the chip have been externally reviewed, tested and certified in accordance with Common Criteria EAL4.

3.1.4 Mifare classic

Can be used in applications such as public transport ticketing; Major cities have chosen MIFARE as their electronic ticketing solution. Applications:

- Public transport
- Access control
- Event ticketing
- Gaming and Authentication

The Smart Anti-Jamming feature allows multiple cards to be operated simultaneously in the field. The anti-tamper algorithm selects each card separately and ensures that the selected card performs the transaction correctly and does not cause data corruption due to the presence of other cards.

Mifare classic is designed for easy integration and convenience. This allows a complete ticketing transaction to be completed within 100 ms. As a result, users using the Mifare class card will not spend a lot of time on ticket transactions, thereby avoiding congestion at the entrance and reducing the boarding time of the bus. The MIFare card can be placed in the wallet to conduct transactions, even if there are coin-like metal objects in the wallet, there will be no impact on communication.

Products	ISO/IEC	Bit Rate	Security	UID Type	Write Operation Tolerance	EEPROM
MIFARE Classic EV1	14443-Type 3A	106	MIFARE CRYPTO1	4NUID & 7 UID	100000	1kB - 4kB

3.1.5 Mifare plus

The MIFARE Plus product family adds security to smart urban services by enabling a seamless migration from contact-free infrastructure to higher security, and its backward compatibility allows a cost-effective upgrade of the security level of native smart card applications. Special features, such as support for mobile and wireless recharge, make the use of smart urban services more convenient.

Products	Explain	ISO/IEC	Security
MIFARE Plus EVx	The MIFARE Plus EVx card serves as a gateway for new smart city applications, while also greatly enhancing the security and connectivity of existing deployments.	ISO/IEC 14443 A 1-4 & ISO 7816-4	48-bit Crypto-1, 128-bit AES
MIFARE Plus SE	The MIFARE Plus SE solution is an entry-level version of the MIFare Plus product family.	ISO/IEC 14443 A 1-4	48-bit Crypto-1, 128-bit AES

3.1.5.1 Mifare plus EVx

Memory	MIFARE Plus EV2	MIFARE Plus X
Memory configuration	Block/sector structures	Block/sector structures
Memory size	2 kB / 4 kB	2 kB / 4 kB
ISO/IEC	ISO/IEC 14443 A 1-4 ISO/IEC 7816	ISO/IEC 14443 A 1-4 ISO/IEC 7816
UID/ONUID	7 B UID or 4 B ONUID	7 B UID or 4 B ONUID
Data transfer rate	Up to 848 kbps according to ISO/IEC 14444 -4	Up to 848 kbps according to ISO/IEC 14444 -4
Algorithm	AES 128-bit, Secure Messaging, Legacy Encryption1	AES 128-bit, Secure Messaging, Legacy Encryption1
Security level concept	Sector-by-sector or card-by-card	Card only
SL1SL3MixMode	Secure backend connected to SL1 sector	-
Transaction MAC (TMAC)	Secure verification of back-end transactions	-

Memory	MIFARE Plus EV2	MIFARE Plus X
Transaction timer	Mitigating man-in-the-middle attack	-

3.1.5.2 MIFARE Plus SE

MIFARE Plus is the only mainstream smart card family compatible with MIFARE Classic 1K and MIFARE Classic 4K to provide pre-issued cards before infrastructure security upgrades. Upgraded to Level III security, MIFARE Plus uses the Advanced Encryption Standard (AES) for authentication, data integrity, and encryption.

The MIFARE Plus SE is the entry-level version of the proven and reliable MIFare Plus product family. It is fully compatible with MIFARE Classic 1K features, providing complete support for MIFARE Classic value blocks.

MIFARE Plus SE is the first choice for customers who are ready to make the switch to higher security, introducing ready-to-use cards for AES security into their existing systems environment, ready for the future.

The MIFARE Plus SE card is easy to distribute and runs the MIFARE Classic system because it uses a linear memory structure that is compatible with MIFARE Classic. And because the MIFARE Plus SE supports all MIFARE Classic value block operations in security levels SL1 and SL3. The MIFARE Plus SE stores its 128-bit AES key at the top of the block. The optional AES authentication in SL1 effectively detects cards that are not part of the system.

3.1.6 Mifare Ultralight

Ideal for low-cost, high-volume applications such as public transportation, loyalty cards, and event tickets.

Products	Explain	ISO/IEC	Security
MIFARE Ultralight AES (new)	Secure, easy to integrate, contact-free IC for limited applications	ISO/IEC 14443 A 1-3	128-bit AES authentication for data and counter protection and CMAC data integrity protection via RF interface
MIFARE Ultralight C	Contact-free IC supports 3DES encryption in limited applications	ISO/IEC 14443 A 1-3	112-bit 3DES

Products	Explain	ISO/IEC	Security
MIFARE Ultralight EV 1	Contact-free IC with password protection for limited smart paper tickets and cards	ISO/IEC 14443 A 1-3	32-bit password + password confirmation

3.2 NTAG Series Cards

NTAG is the market-leading portfolio of NFC tag IC solutions for IoT consumer and industrial applications. These powerful NFC tags offer varying levels of security and functionality to meet a wide range of applications and customer requirements. Companies can now introduce smart, digitally connected products that drive new value throughout the lifecycle. They also enable a new user experience with a more dynamic and higher level of personalization, thanks to the extensive NTAG feature set.

Passive-powered NFC tag and label IC solutions include the NTAG 21x, NTAG 213 tag tamper detection, and cryptographically secure NTAG DNA series. NTAG products are fully NFC Forum compliant, covering both Category 2 and Category 4 tags. NTAG ICs store NDEF (NFC Data Exchange Format) data, making them fully compatible with any NFC device.

Products	Memory Access Protection	Encrypted Communication	NFC Hit Counter	Originality Signature	Secure NDEF (SUN) Message	Tamper Detection
NTAG 424 DNA Tag Tamper Detection	AES-128 bit key (5)			56 bytes		Primary open and current status
NTAG 424 DNA	AES-128 bit key (5)			56 bytes		-
NTAG 213 Tag Tamper Detection	32-bit password	-		32 bytes (programmable)	-	Once-open and current state
NTAG 213/215/216	32-bit password	-		32 bytes	-	-

Products	Standard Edition	User Memory [bytes]	Data Retention Period [years]	Write Operation Endurance [times]	Label Certification
NTAG 424 DNA Tag Tamper Detection	ISO /IEC 14443-A NFC Forum T4T	416, including 128b security data files	50	200,000	AES-128 bit key (5)
NTAG 424 DNA	ISO /IEC 14443-A NFC Forum T4T	416, including 128b security data files	50	200,000	AES-128 bit key (5)
NTAG 213 Tag Tamper Detection	ISO 14443A 1-3 NFC Forum T2T	144	10	100,000	-
NTAG 213/215/216	ISO 14443A 1-3 NFC Forum T2T	144 / 504 / 888	10	100,000	-

3.3 ICODE

The short-range RFID solution is compliant with ISO/IEC 15693 and ISO/IEC 18000-3 standards and follows the NFC Forum Tag Type 5 specification. ICODE offers a working range of up to 1.5 meters and a remote reader, ICODE also offers an additional read range compared to the ISO/IEC 14443 and a standard ISO/IEC 15693 reader for an ultra-small form factor and NFC handset readability: The ICODE product specification has a range of valuable features. Such as protection of EAS, Application Family Identifier (AFI), memory access, and privacy.

ICODE products are used in a wide range of applications, as listed below:

- Library management
- Identification of consumables and accessories
- Brand protection and anti-counterfeiting
- Supply chain visibility and control
- Industrial use

Products	Standard Edition	User memory [bit]	EPC Code Size [Bits]	EAS protection	EAS Selectivity
ICODE SLIX 2	ISO 18000-3M1 NFC Forum T5T	2528	-	32-bit password	
ICODE SLIX	ISO 18000-3M1 NFC Forum T5T	896	-	32-bit password	-
ICODE SLIX-L	ISO 18000-3M1 NFC Forum T5T	256	-	32-bit password	
ICODE SLIX-S	ISO 18000-3M1 NFC Forum T5T	1280	-	32-bit password	

	AFI	AFI Protection	Keep quiet	Original signature	Memory protection
ICODE SLIX 2		32-bit password			32-bit password
ICODE SLIX		32-bit password	-	-	-
ICODE SLIX-L		32-bit password	-	-	32-bit password
ICODE SLIX-S		32-bit password	-	-	32-bit password

3.4 Felica

FeliCa has the same technology that is applicable to cash or identification cards as regular IC cards, but the instruction set is specialized for applications that require high-speed processing features (automatic recharge devices, building access controls, etc.) or checkout (convenience stores), etc. Therefore, it is not compatible with the basic instructions of ISO 7816-3. In addition, the internal memory of the IC chip is fixed as 16-byte records, which is incompatible with the file structure stipulated by ISO 7816-3.

In terms of encryption processing, Triple DES is used for mutual authentication and DES or Triple DES for communication. There is no public key encryption specification. Dual model (contact/non-contact) is only used for contact communication, although it can be encrypted with a public key.

In mutual authentication, the indented code is used as the password of the solution. Instead of saying that each item is authenticated individually, it is encrypted by a complex access code. The key generated is called a flinch code, and this flinch code can be used by up to 16 items. The indented code does not produce the original password. In this way, high speed processing is achieved without compromising the level of security

Applies to:
Public transport
Automatic recharge device
Building access control
Checkout (Convenience Store)

4. General function

4.1 Working mode setting

FM3281 supports command mode, reporting mode and service mode to meet different application requirements.

i. Reporting mode: The equipment is always in the process of finding the card. When the card enters the valid range, the UID is automatically output, and other business instructions are not allowed.
ii. Command mode: card request, anti-collision, card selection and other operations are controlled by the upper computer software.
iii. Business mode: When the equipment is initialized in the process of card searching, the UID will be automatically output when the card enters the valid range. It can also be operated by other business instructions. After the operation is completed, it is necessary to re-control the access to the card search according to its own needs.

Commands:

NFCMOD0 Repoting mode
NFCMOD1 Command mode
NFCMOD2 Business model

4.2 Setting of time interval for card searching

Set the time interval for card searching, and the unit is ms. The smaller the time interval for card searching is, the higher the power consumption of the device is, and the faster the card is read.

Command:

NFC DUR $ $range 10-1000

4.3 Return to card type settings

Whether to return card type and output ASCII character string when automatically outputting UID. Card type includes protocol type (2 bytes) + ATQA (4 bytes) + SAK (2 bytes). The protocol type contains:
01：ISO14443-A
02：ISO14443-B
03：ISO15693

04： FELICA
05: Chinese 2nd generation ID card

NFCCTP0: Prohibites Output Card Type
NFCCTP1: enables output card type

4.4 Enable switch

NFC enable switch.
Commands:

NFCENA0: Disable NFC function
NFCENA 1: Enable NFC function

4.5 Reread Latency

NFC reread delay switch
Commands:

NFC RDE0: NFC reread delay is invalid. The same NFC tag can be read continuously at any time.
NFC RDE1: If an NFC tag is read and the NFC tag is read for the secondary consecutive time within the NFC re-read delay time, the NFC tag read for the secondary time will be ignored and will not be output.

4.6 Reread Delay Time Settings

Reread delay.
Command:

NFCRDT $ $Range 1-3600000 time units ms

5. Card operation

5.1 Instruction format

Host->Scanner

CMD	Function Code	Data
NFCCMD	4 byte	N byte

- CMD: Unified Command.
- Function Code: Card Operation Function Code.
- Data: Host sends to the card data in hexadecimal string format. For example, HEX: 0x41, ASCII character 41. (Can be empty).

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	4 byte	N byte	2 byte	N byte

- CMD: The unified command NFC CMD is consistent with that sent by Host.
- Function Code: Card operation function code is consistent with that sent by Host.
- Data: Host sends to the card data in hexadecimal string format. For example, HEX: 0x41, ASCII character 41. (Can be null) is consistent with that sent by Host.
- Status: 00, 71 for success, others for failure. 2 bytes.
- Card Data: Data returned by the card. (In hexadecimal string format, for example, if the data obtained from the card is HEX: 0x41, then the output of the scanner is ASCII character 41, which can be null).

5.2 Select Protocol

If the reader supports cards of multiple protocols, the corresponding card protocol needs to be selected before card operation. Applies to the command mode.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0010	2 byte

Data: the protocol type to be selected, and the value is 00--TypeA; 01--TypeB; 02--ICODE2; 03 – Felica 2 bytes.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0010	N byte	2 byte	NULL

5.3 Find the card again

When the working mode is set as the business mode, you need to use this command to actively find a card again after finding the card

Host->Scanner

CMD	Function Code	Data
NFCCMD	0011	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0011	NULL	2 byte	NULL

5.4 FormatKeyEntry

Format a key entry to a new KeyType into KeyStore.

Host->Scanner

CMD	Function Code	Data
NFCCMD	002B	8 byte

Data:
The following length and contents are before conversion to hexadecimal strings.
The data packet data is wKeyNo (2 bytes, LSB first) and wKeyType (2 bytes, LSB first).
wKeyNo: Key number of the key to be loaded. Range: from 0 to 3.
wKeyType: New Key type of the KeyEntry (predefined type of KeyType).
The values are as follows:

0x00 --AES 128 Key [16 bytes]
0x04 --2 Key Triple Des.[16 bytes]
0x05 --3 Key Triple Des.[24 bytes]

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	002B	8 byte	2 byte	NULL

5.5 SetKey

Change a key entry at a given version in Key Store.

Host->Scanner

CMD	Function Code	Data
NFCCMD	002C	48/64 byte

Data:
The following length and contents are before conversion to hexadecimal strings.
Packet data is wKeyNo, wKeyVersion, wNewKeyType, pNewKey, wNewKeyVersion
wKeyNo: Key number of the key in Key Store, 2 bytes and LSB first.
The value is from 0 to 3.
wKeyVersion: Key version of the **wKeyNo** in Key Store, 2 bytes and LSB first.
wNewKeyType: New Key type of the **wKeyNo**, 2 bytes and LSB first.
The values are as follows:

0x04 --2 Key Triple Des [16 bytes].
0x05 --3 Key Triple Des [24 bytes].

pNewKey: The new key to be changed, 16 bytes or 24 bytes.
wNewKeyVersion: New Key Version of the **wKeyNo** to set, 2 bytes and LSB first.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	002C	48/64 byte	2 byte	NULL

Call **Format Key Entry** and Set Key to load the key into Key Store to authenticate the Desfire EV1/EV2/EV3/Light or NTag42X card key.

5.6 ISO14443-A series card activation instruction

5.6.1 Request Card

Host->Scanner

CMD	Function Code	Data
NFCCMD	0012	2 Byte

Data: request mode, 2 bytes
00: With the halt command, the card can only be requested to be placed on the scanner once. If it is operated again, it needs to be removed and placed again to prevent multiple operations on the card.
01: The request has been successful.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0012	2 Byte	2 byte	4 Byte

Card Data: ATQA Data.

5.5.2 Anti-collision

Host->Scanner

CMD	Function Code	Data
NFCCMD	0013	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0013	NULL	2 byte	8 Byte

Card Data: default serial number of card.

5.5.3 Select a card

Host->Scanner

CMD	Function Code	Data
NFCCMD	0014	8 Byte

Data: serial number of the card returned for anti-collision.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0014	8 Byte	2 byte	2 Byte

Card Data: SAK Data.

5.5.4 Secondaryary anti-collision

Secondaryary anti-collision and secondaryary card selection are required for the 7-byte UID card.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0015	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0015	NULL	2 byte	8 Byte

Card Data: the secondaryary serial number returned by the card.

Note: 0 byte is removed from the card serial number returned by the first anti-collision, and the remaining 3 bytes and the 4 bytes returned by the secondary anti-collision finally form a 7-byte UID.
For example, the serial number returned by the first anti-collision is { 0x88, 0x04, 0xFD, 0xF4 }.
The sequence number returned by the secondary anti-collision is { 0xEA, 0 xA9, 0x6A, 0x80 },
The final card serial number is { 0x04, 0 xFD, 0 xF4, 0xEA, 0xA9, 0x6A, 0x80 }

5.5.5 Secondaryary selection card

Host->Scanner

CMD	Function Code	Data
NFCCMD	0016	8 byte

Data: serial number of card returned by secondaryary anti-collision

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0016	8 byte	2 byte	2 Byte

Card Data: SAK Data

5.5.6 Stop Card

Host->Scanner

CMD	Function Code	Data
NFCCMD	0017	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NCMD	0017	NULL	2 byte	NULL

5.6 Contactless CPU card operation (ISO14443 TypeA)

Operation process of non-connected CPU card:

- Command mode: request-> anti-collision-> select card-> secondary anti-collision-> secondary select card-> reset-> send command-> stop
- Service mode: UID received-> reset-> send command-> stop-> find card

5.6.1 Reset

Host->Scanner

CMD	Function Code	Data
NFCCMD	0018	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0018	NULL	2 byte	N byte

Card Data: It is the reset response information returned by the card (ATS).

5.6.2 Send command

Pass-through application protocol instruction.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0019	N byte

Data:
NAD (default 00)
CID (default 00)
PCB (default 00)
LEN: (COS instruction length)
Data [4] -DATA [LEN + 4] COS instruction

For example, ISO7816-4 gets the random number
CLA 00
INS 84
P1 00
P2 00
Le 08
Send command: NFCCMD0019000000050084000008

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0019	N byte	2 byte	N byte

Card Data: data returned by pass through instruction.

5.6.3 Stop Card

Host->Scanner

CMD	Function Code	Data
NFCCMD	001A	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	001A	NULL	2 byte	NULL

5.7 Mifare Desfire Light

Operation process of Mifare Desfire light card:

- Command mode: request-> anti-collision-> select card-> secondary anti-collision-> secondary select card-> reset-> send command to verify key, read and write-> stop
- Service mode: receive UID-> reset-> send command to verify key, read and write-> stop-> find card

5.7.1 Authenticate EV2

Performs an authentication based on standard AES. This will be using the AES128 keys and will generate and verify the contents based on generic AES algorithm.

Host->Scanner

CMD	Function Code	Data
NFCCMD	002D	N byte

Data:

The following length and contents are before conversion to hexadecimal strings

Packets data are bFirstAuth, wOption, wKeyNo, wKeyVer, bKeyNoCard, bDivLen, pDivInput, bLenPcdCapsIn, bPcdCapsIn.

bFirstAuth: 1 byte,One of the below options

0x00 --Non First Auth in regular EV2 auth Mode.

0x01 --First Auth in regular EV2 auth Mode.

0x02 --Non First Auth in LRP mode.

0x03 --First Auth in LRP mode.

wOption: Diversification option, 2 bytes. The values are as follows

0xFFFF --No diversification.

0x0000 --Encryption based method of diversification.

0x0001 --CMAC based method of diversification.

wKeyNo: The key number in keystore to authenticate, 2 bytes and LSB first.

wKeyVer: Key version in the key store, 2 bytes and LSB first.

bKeyNoCard: Key number on card, 1 byte.

bDivLen: Length of diversification input, 1 byte.

pDivInput: Diversification input, up to 16 bytes. Can be NULL.

bLenPcdCapsIn: Length of PcdCapsIn, 1 byte.Always 0 for following authentication.

pPcdCapsIn: PCD Capabilities. Up to 6 bytes.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	002D	N byte	2 byte	24 byte

Card Data: The following length and contents are before conversion to hexadecimal strings

Result data data is bPcdCapsOut, bPdCapsOut.

bPcdCapsOut: PCD Capabilities, 6 bytes.

bPdCapsOut: PD Capabilities, 6 bytes.

5.7.2 Get Key Version

Read out the current key version of any key stored on the PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	002F	4 byte

Data:

The following length and contents are before conversion to hexadecimal strings

The data packet data is bKeyNo and bKeySetNo in turn

bKeyNo: Key number of the targeted key, 1 byte.

bKeySetNo: Key set number, 1 byte. Optional as it is passed only when bit6 of **bKeyNo** is set.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	002F	4 byte	2 byte	N byte

Card Data: data is the version of the specified key.

5.7.3 Get Version

Returns manufacturing related data of the PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0030	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0030	NULL	2 byte	56

5.7.4 Set Configuration

Configures the card and pre personalizes the card with a key,defines if the UID or the random ID is sent back during communication setup and configures the ATS string.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0031	N byte

Data:
The following length and contents are before conversion to hexadecimal strings
The data packets data are bOption, bDataLen, pData.
bOption:Define length and content of the **pData**, 1 byte.

0x00 --Update the PICC Configuration.
0x02 --Update the ATS.
0x03 --Update the SAK.
0x04 --Update the Secure Messaging.
0x05 --Update Capibility Data.
0x06 --Application Renaming.
0x08 --File Renaming.
0x09 --Value file type configuration.
0x0A --Failed Authentication Counter Configuration.
0x0B --Hardware Configuration.

bDataLen:The size of **pData**, 1 byte.
pData:Configuration data for the option specified.The length of bytes is specified by **bDataLen.**

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0031	N byte	2 byte	NULL

5.7.5 Get Card UID

Return the Unique ID of the PICC. The key must be authenticated before this command.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0032	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0032	NULL	2 byte	14 byte

The following length and contents are before conversion to hexadecimal strings
Card data: data is the 7-byte UID returned.

5.7.6 Get File IDs

Returns the file identifiers of all active files within the currently selected application. Each File ID is coded in one byte.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0033	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0033	NULL	2 byte	N byte

The following length and contents are before conversion to hexadecimal strings.
Card Data :
Results data is the fields, if the Transaction MAC file is present 6 file IDs will be returned, otherwise 5 file IDs returned.

5.7.7 Get ISO File IDs

Returns 2 bytes ISO/IEC 7816-4 File IDentifiers of all active files within the currently selected application Each ISO File ID is coded in 2 bytes.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0034	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0034	NULL	2 byte	16 byte

The following length and contents are before conversion to hexadecimal strings.
Card Data: Data is the fields of the 3 Standard data files and the Cyclic Record file.

5.7.8 Get File Settings

Get information on the properties of a specific file.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0035	2 byte

Data:
data is the file number of the targeted file.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0035	2 byte	2 byte	N byte

Card Data: N bytes of the file settings.

5.7.9 Read Data

Read data from standard data files or backup data files

Host->Scanner

CMD	Function Code	Data
NFCCMD	0039	18 byte

Data:
The following length and contents are before conversion to hexadecimal strings
The data packet data is bOption, bIns, bFileNo, pOffset, pLength.
bCommOption:Communication settings for the file, 1 byte.

0x00 --Plain mode of communication.
0x10 --MAC mode of communication.
0x30 --Enciphered mode of communication.

bIns: The ISO14443-4 chaining format, 1 byte. This should always be set to 1.
bFileNo: File number to be read data, 1 byte.
pOffset: Starting position for the read operation, 3 bytes and LSB first.
pLength:The number of bytes to be read, 3 bytes and LSB first.
Ifitis 0x000000, Read the entire data file, starting from the position specified in the **pOffset** value.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0039	18 byte	2 byte	N byte

Card Data :
If more data is to be read, the status **0x71** is returned, then should call This command again with **bCommOption=| 0x02**.

5.7.10 Write Data

Write data to standard data files or backup data files.

Host->Scanner

CMD	Function Code	Data
NFCCMD	003A	N byte

Data:
The following length and contents are before conversion to hexadecimal strings
Packet data is bOption, bIns, bFileNo, pOffset, pTxDataLen, pTxData.
bOption:Communication settings for the file, 1 byte.

0x00 --Plain mode of communication.
0x10 --MAC mode of communication.
0x30 --Enciphered mode of communication.

bIns:The ISO14443-4 chaining format, 1 byte. This should always be set to 1.
bFileNo: File number to be writedata, 1 byte.
pOffset: Starting position for the writeoperation, 3 bytes and LSB first.
pTxDataLen:The length of data to be written, 3 bytes and LSB first.
pTxData: Data to be written, the length of bytes is specified by **pTxDataLen**.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	003A	N byte	2 byte	NULL

5.7.11 ISO Select File

This command is a standard ISO/IEC 7816-4 command. It selects either thePICC level, an application or a file within the application.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0046	N byte

Data:
The following length and contents are before conversion to hexadecimal strings

Packets data are bOption, bSelector, pFid, pDFName, bDFNameLen, bExtendedLenApdu.
bOption: Indicates whether to return File Control Information (FCI), 1 byte.
0x00 --Return FCI;
0x0C --No return FCI.
bSelector:Selection Control, 1 byte.
0x00 --Select MF, DF or EF, by file identifier.
0x01 --Select child DF.
0x02 --Select EF under the current DF, by file identifier.
0x03 --Select parent DF of the current DF.
0x04 --Select by DF name.
pFid: File Identifier, 2 bytes. Valid only when **bSelector**= 0x00 or 0x02.
pDFName: DF Name, up to 16 bytes. Valid only when **bSelector**= 0x04.
bDFNameLen: The length of DFName, 1 byte. Valid only when **bOption**= 0x04.
bExtendedLenApdu: Default 0x00.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0046	N byte	2 byte	N byte

Card Data :
The following length and contents are before conversion to hexadecimal strings
data is the returned FCI stored in file ID 1Fh of the DF.
Valid only when **bOption**= 0x00.

5.7.12 Get Config

Perform a Get Config command.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0049	4 byte

Data: The following length and contents are before conversion to hexadecimal strings
Packet data is wConfig.
wConfig: Configuration to read, 2 bytes. Will be one of the below values

0xA100 --Get additional info of a generic error.
0xA200 --Get current status of command wrapping in ISO 7816-4 APDUs.
0xA300 --Get Short Length APDU wrapping in ISO 7816-4 APDUs.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0049	4 byte	2 byte	N byte

Card Data: data is the value for the mentioned configuration.

5.7.13 Set Config

Perform a SetConfig command.

Host->Scanner

CMD	Function Code	Data
NFCCMD	004A	4 byte

Data: The following length and contents are before conversion to hexadecimal strings
Packets data are wConfig, wValue.
wConfig: Configuration to set, 2 bytes. Will be one of the below values

0xA100 --Set additional info of a generic error.
0xA200 --Set current status of command wrapping in ISO 7816-4 APDUs.
0xA300 --Set Short Length APDU wrapping in ISO 7816-4 APDUs.

wValue: The value for the mentioned configuration, 2 bytes and LSB first.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	004A	4 byte	2 byte	NULL

Card Data: data is the value for the mentioned configuration.

5.7.14 Reset Authentication

Reset Authentication status.

Host->Scanner

CMD	Function Code	Data
NFCCMD	004B	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	004B	NULL	2 byte	NULL

5.7.15 Read Sign

Performs the originality check to verify the genuineness of PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	004E	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	004E	NULL	2 byte	112 byte

Card Data:
The following length and contents are before conversion to hexadecimal strings
data is the plain 56 bytes originality signature of the PICC.

5.8 Mifare Desfire EV1/EV2/EV3

Operation process of Mifare Desfire card:

- Command mode: request-> anti-collision-> select card-> secondary anti-collision-> secondary select card-> reset-> send command to verify key, read and write-> stop
- Service mode: receive UID-> reset-> send command to verify key, read and write-> stop-> find card

5.8.1 Authencation

Performs an Authentication with the specified key number， This command supports the backward compatible D40 authentication. The command can be used with DES and 2K3DES keys and performs DESFire native authentication。

Host->Scanner

CMD	Function Code	Data
NFCCMD	004F	N byte

Data:

The following length and contents are before conversion to hexadecimal strings

The data packet data is wOption, wKeyNo, wKeyVer, bKeyNoCard, pDivInput, bDivLen.

wOption: Diversification option, 2 bytes. The values are as follows

0xFFFF --No diversification.

0x8000 --Encryption based method, 2K3DES half key diversification.

0x0000 --Encryption based method, 2K3DES full key diversification.

0x0001 --CMAC based method of diversification.

wKeyNo: The key number in keystore to authenticate,2 bytes and LSB first.

wKeyVer: Key version in the key store, 2 bytes and LSB first.

bKeyNoCard:Key number on card, 1 byte.

ORed with **0x80**to indicate Shared application identifier (SAI).

pDivInput: Diversification input, up to 16bytes. Can be NULL.

bDivLen: Length of diversification input, 1 byte.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	004F	N byte	2 byte	NULL

5.8.2 AES Authentication

Perform an AES Authentication。 The command should be used with AES128 keys.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0051	N byte

Data:

The following length and contents are before conversion to hexadecimal strings.

The data packet data is wOption, wKeyNo, wKeyVer, bKeyNoCard, pDivInput, bDivLen.

wOption: Diversification option, 2 bytes. The values are as follows

0xFFFF --No diversification.

0x0000 --Encryption based method of diversification.

0x0001 --CMAC based method of diversification.

wKeyNo: The key number in keystore to authenticate, 2 bytes and LSB first.

wKeyVer: Key version in the key store, 2 bytes and LSB first.

ORed with **0x80** to indicate Shared application identifier (SAI).

pDivInput: Diversification input, up to 16 bytes. Can be NULL.

bDivLen: Length of diversification input, 1 byte

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0051	N byte	2 byte	NULL

5.8.3 Authenticate EV2

Performs an Ev2 First or Non First Authentication.This will be using the AES128 keys and will generate and verify the contents based on generic AES algorithm.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0052	N byte

Data:

The following length and contents are before conversion to hexadecimal strings

Packets data are bFirstAuth, wOption, wKeyNo, wKeyVer, bKeyNoCard, bDivLen, pDivInput, bLenPcdCapsIn, bPcdCapsIn.

bFirstAuth: 1 byte,One of the below options

0x00 --Following (NonFirst) Authentication;

0x01 --First Authentication.

wOption: Diversification option, 2 bytes. The values are as follows

0xFFFF --No diversification.

0x0000 --Encryption based method ofdiversification.

0x0001 --CMAC based method of diversification.

wKeyNo: The key number in keystore to authenticate, 2 bytes and LSB first.

wKeyVer: Key version in the key store, 2 bytes and LSB first.

bDivLen: Length of diversification input, 1 byte.

pDivInput: Diversification input, up to 16 bytes. Can be NULL.

bLenPcdCapsIn: Length of PcdCapsIn, 1 byte.Always 0 for following authentication.

pPcdCapsIn: PCD Capabilities. Up to 6 bytes.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0052	N byte	2 byte	24 byte

Card Data:

The following length and contents are before conversion to hexadecimal strings

Result data data is bPcdCapsOut, bPdCapsOut.

bPcdCapsOut: PCD Capabilities, 6 bytes.

bPdCapsOut: PD Capabilities, 6 bytes.

5.8.4 Change Key Settings

Changes the master key settings on PICC and application level.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0053	2 byte

Data:data is New key settings to be updated**.**

If AID = 0 selected, its PICCKeySettiongs, else its AppKeySettings.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0053	N byte	2 byte	NULL

5.8.5 Get Key Settings

Gets information on the PICC and application master key settings.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0054	NULL

If AID = 0 selected, its PICCKeySettiongs, else its AppKeySettings.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0054	NULL	2 byte	4/6 byte

Card Data:
data is the KeySettings, Can be 4 or 6 bytes.
If AID = 0 selected, its PICCKeySettiongs, else its AppKeySettings.

5.8.6 Change Key

Changes any key on the PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0055	N byte

Data:
The following length and contents are before conversion to hexadecimal strings
The packet data is wOption, wOldKeyNo, wOldKeyVer, wNewKeyNo, wNewKeyVer, bKeyNoCard, pDivInput, bDivLen.
wOption:Diversification option, 2 bytes. The values are one of the below options.

- 0xFFFF
- 0x0002 | 0x0020
- 0x0004 | 0x0020
- 0x0002 | 0x0008
- 0x0004 | 0x0010
- 0x0002 | 0x0004
- 0x0002 | 0x0004 | 0x0020
- 0x0002 | 0x0004 | 0x0008 | 0x0010

The value description:

0xFFFF--No diversification.
0x0002--Diversification of new key required (bit 1).
0x0004--Old key was diversified (bit 2).
0x0008--New key diversification using one rnd (bit 3).
Default is two rounds.
0x0010--Old key diversification using one rnd (bit 4).
Default is two rounds.
0x0020--Key diversification method based on CMAC (bit 5).
Default is Encryption method

wOldKeyNo: Old key number in keystore, 2 bytes and LSB first.

wOldKeyVer: Old key version in keystore, 2 bytes and LSB first.
wNewKeyNo: New key number in keystore,2 bytes and LSB first.
wNewKeyVer: New key version in keystore,2 bytes and LSB first.
bKeyNoCard:Key number of the key to be changed, 1 byte.
ORed with **0x80**to indicate Shared application identifier (SAI).
pDivInput: Diversification input.Up to 16bytes,can be NULL.
bDivLen: Length of **pDivInput**, 1 byte.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0055	N byte	2 byte	NULL

5.8.7 Change Key EV2

Changes any key present in keyset on the PICC. The key on the PICC is changed to the new key. The key type of the application keys cannot be changed. The key type of only the PICC master key can be changed. The keys changeable are PICCDAMAuthKey, PICCDAMMACKey, PICCDAMEncKey, VCConfigurationKey, SelectVCKey, VCProximityKey, VCPollingEncKey, VCPollingMACKey.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0056	N byte

Data:
The following length and contents are before conversion to hexadecimal strings
The packet data is wOption, wOldKeyNo, wOldKeyVer, wNewKeyNo, wNewKeyVer, bKeySetNo, bKeyNoCard, pDivInput, bDivLen.
wOption:Diversification option, 2 bytes. Same as **5.8.6 Change Key.**
wOldKeyNo: Old key number in keystore, 2 bytes and LSB first.
wOldKeyVer: Old key version in keystore, 2 bytes and LSB first.
wNewKeyNo: New key number in keystore,2 bytes and LSB first.
wNewKeyVer: New key version in keystore, 2 bytes and LSB first.
bKeySetNo: Key set number within targeted application, 1 byte.
bKeyNoCard:Key number of the key to be changed, 1 byte.
ORed with **0x80** to indicate Shared application identifier (SAI).
pDivInput: Diversification input.Up to 16bytes,can be NULL.
pDivLen: Length of **pDivInput**, 1 byte.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0056	N byte	2 byte	NULL

5.8.8 Get Key Version

Read out the current key version of any key stored on the PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0057	4 byte

Data:
The following length and contents are before conversion to hexadecimal strings
The data packet data is bKeyNo, bKeySetNo.
bKeyNo: Key number of the targeted key, 1 byte.ORed with one of the below options
0x80 --Secondaryary application indicator (SAI).
0x00 --**KeySetNo**not available in the command.
0x70 --**KeySetNo**available in the command.
bKeySetNo: Key set number, 1 byte. Optional as it is passed only when bit6 of **bKeyNo** is set. ORed with one of the below options
0x00 --Key version retrieval from specific key set.
0x80 --Key set versions retrieva.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0057	4 byte	2 byte	N byte

Card Data:
Data is Key set versions of the selected application ordered by ascending key set number, i.e. starting with the AKS.

5.8.9 Create Application

Create new applications on the PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	005B	N byte

Data:
The following length and contents are before conversion to hexadecimal strings
The data package data is bOption, pAid, bKeySettings 1, bKeySettings 2, bKey Settings 3, pKeySetValues,**pISOFileId**, **pISODFName**, **bISODFNameLen**.
bOption:Option to represent the present of ISO information, 1 byte.
0x01 meaning **pISOFileId**is supplied.

0x02 meaning **pISODFName**is present.

0x03 meaning both **pISOFileId**and **pISODFName**are present.

0x00 meaning both not present.

pAid: Application Identifier of the application to be created, 3 bytes. The value ranges from 0x000001 to 0xFFFFFF, 0x000000 is reserved for the PICC level.

bKeySettings1: Application Key settings, 1 byte.

bKeySettings2: Several other key settings, 1 byte.

bKeySettings3: Additional optional key settings,present if **bKeySettings2**[b4] is set, 1 byte.

pKeySetValues: The Key set values for the application, present if **bKeySettings3**[b4] is set. Should consists of the following data

Byte0 = AKS ver

Byte1 = Number of Keysets

Byte2 = MaxKeysize

Byte3 = Application KeySet Settings

pISOFileId: ISO File ID if present, 2 bytes and LSB first.

pISODFName: ISO DF Name if present, upto 16 bytes. Can be NULL.

bISODFNameLen: Size of **pISODFName**if that is present, 1 byte.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	005B	N byte	2 byte	NULL

5.8.10 Delete Application

Permanently deactivates applications on the PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	005D	N byte

Data:

The following length and contents are before conversion to hexadecimal strings

The data packet data is pAid, pDAMMAC, and bDAMMAC _ Len. In turn.

pAid: Application Identifier of the application to be deleted, 3 bytes. The value ranges from 0x000001 to 0xFFFFFF, 0x000000 is reserved for the PICC level and cannot be deleted.

pDAMMAC: The MAC calculated by the card issuer to allow delegatedapplication deletion, present if KeyID.PICCDAMAuthKey or KeyID.NXPDAMAuthKey authentication, 8 bytes.

bDAMMAC_Len: Size of **pDAMMAC** if that is present, 1 byte.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	005D	N byte	2 byte	NULL

5.8.11 Get Application IDs

Returns application identifiers of all applications on the PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	005E	2 byte

Data:
The following length and contents are before conversion to hexadecimal strings
The value of data packet data is 0 x00 or 0 x02.
0x00 --Default exchange mode.
0x02 --Starts transmission with and R(ACK) block and performs Rx chaining with the Card/Target

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	005E	2 byte	2 byte	NULL

5.8.1 Select Application

Select 1 or 2 applications or the PICC level specified by their application identifier

Host->Scanner

CMD	Function Code	Data
NFCCMD	0061	N byte

Data:
The following length and contents are before conversion to hexadecimal strings
The data package data is bOption, pAid, pAid2.
bOption: indicates the presence of secondary Aid, 1 byte. One of the below options
0x00 --Primary application selection.
0x01 --Secondaryary application selection.
pAid: The primary application identifier to be selected, 3 bytes and LSB first.
pAid2:The secondaryary application identifier to be selected when **bOption**= 0x01, 3bytes and LSB first.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0061	N byte	2 byte	NULL

5.8.13 Format PICC

Release the PICC user memory.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0062	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0062	NULL	2 byte	NULL

5.8.14 Get Version

Returns manufacturing related data of the PICC

Host->Scanner

CMD	Function Code	Data
NFCCMD	0063	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0063	NULL	2 byte	56

5.8.15 Free Memory

Returns free memory available on the PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0064	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0064	NULL	2 byte	6

Card Data: Data is the size of free memory and LSB first.

5.8.16 Set Configuration

Configures the card and pre personalizes the card with a key, defines if the UID or the random ID is sent back during communication setup and configures the ATS string.

Commands need to be supplemented.

5.8.17 Get Card UID

Return the Unique ID of the PICC.The key must be authenticated before this command.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0066	4 byte

The following length and contents are before conversion to hexadecimal strings
bExchangeOption: Flag to indicate whether the parameter information bOption to be exchanged to PICC or not.

0x00 -- Not exchanging the Option information to PICC.(EV1/EV2)

0x01 -- Exchanging the Option information to PICC.(EV3)

bOption: Indicates whether a 4-byte NUID is returned from PICC.

0x00 -- No returned 4 bytes NUID.

0x01 -- Returned 4 bytes NUID.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0066	4 byte	2 byte	14 byte

The following length and contents are before conversion to hexadecimal strings
Card data: data is the 7-byte UID returned.

5.8.18 Get File IDs

Returns the File IDentifiers of all active files within the currently selected application.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0067	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0067	NULL	2 byte	N byte

The following length and contents are before conversion to hexadecimal strings
Card Data : data is the fields, Each File ID is coded in one byte and is in the range from 0x00 to 0x1F.

5.8.19 Get File Settings

Get information on the properties of a specific file.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0069	2 byte

Data:
data is File number of the targeted file.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0069	2 byte	2 byte	N byte

Card Data: N bytes of the file settings.

5.8.20 Change File Settings

Changes the access parameters of an existing file.

Host->Scanner

CMD	Function Code	Data
NFCCMD	006B	N byte

Data:
The following length and contents are before conversion to hexadecimal strings

The data package data is bOption, bFileNo, bFileOption, pAccessRights, bAddInfoLen, pAddInfo.
bOption: Indicates the mode of communication to be used while exchanging the data to PICC, 1 byte.
One of the below mentioned options

0x00 --Plain mode of communication
0x10 --MAC mode of communication.
0x30 --Enciphered mode of communication.

Ored with **0x80**to exchange the information available in **pAddInfo**buffer as is.

bFileNo: File number for which the setting needs to be updated, 1 byte.

ORed with **0x80**to indicate secondaryary application indicator.

bFileOption: New communication settings for the file, 1 byte.

0x00 --Plain mode of communication.
0x01--MAC mode of communication.
0x03--Enciphered mode of communication.
Ored with one of the below flags if required:
0x80 --Additional Access Rigths enabled.
0x40 –-Secure Dynamic Messaging and Mirroring support enabled
0x20 --5th Bit indicating TMC limit config.

pAccessRights: The new access right to be applied for the file, 2 bytes.

bAddInfoLen: The length of **pAddInfo**, 1 byte.

pAddInfo: Contains the following optional information.

[Additional access rights] || [SDMOption || SDM AccessRights ||
VCUIDOffset || SDMReadCtrOffset || PICCDataOffset ||
SDMACInputOffset || SDMENCOffset || SDMENCLength ||
SDMMACOffset] || [TMCLimit]

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	006B	N byte	2 byte	NULL

5.8.21 Create StdData File

Creates files for storage of plain unformatted user data within an existing application on the PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	006C	N byte

Data:

The following length and contents are before conversion to hexadecimal strings

The data package data is bOption, bFileNo, pISOFileId, bCommSett, pAccessRights, pFileSize.

bOption: Indicates ISO file ID is present or not, 1 byte.

0x00 --ISOFileId is not provided.
0x01 --ISOFileId is provided and is valid.

bFileNo:File number of the file to be created, 1 byte.
pISOFileId: When **bOption**= 0x00, No this parameter. When **bOption**= 0x01, Indicates ISO/IEC 7816-4 File ID for the file to be created, 2 byts. Excluding the following values reserved by ISO: 0x0000, 0x3F00, 0x3FFF, 0xFFFF.
bCommSett: Communication settings for the file, 1 byte.

0x00 --Plain mode of communication.
0x01 --MAC mode of communication.
0x03 --Enciphered mode of communication.
Ored with one of the below options
0x20 --MIFARE Classic contactless IC mapping support enabled.
0x40 --Secure Dynamic Messaging and Mirroring support enabled.
0x80 --Additional Access Rigths enabled.

pAccessRights: Access Rights, 2 bytes.
pFileSize: File size in bytes for the file to be created, 3 bytes and LSB first.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	006C	N byte	2 byte	NULL

5.8.22 Create Backup Data File

Creates files for the storage of plain unformatted user data within an existing application on the PICC, additionally supporting the feature of an integrated backup mechanism.

Host->Scanner

CMD	Function Code	Data
NFCCMD	006D	N byte

Data:
The following length and contents are before conversion to hexadecimal strings
The data package data is bOption, bFileNo, pISOFileId, bCommSett, pAccessRights, pFileSize.
bOption: Indicates ISO file ID is present or not, 1 byte.

0x01 --ISOFileId is provided and is valid.

bFileNo:File number of the file to be created, 1 byte.
pISOFileId: When **bOption**= 0x00, No this parameter. When **bOption**= 0x01, Indicates ISO/IEC 7816-4 File ID for the file to be created, 2 byts. Excluding the following values reserved by ISO: 0x0000, 0x3F00, 0x3FFF, 0xFFFF.

bCommSett: Communication settings for the file, 1 byte.

0x00 --Plain mode of communication.
0x01 --MAC mode of communication.
0x03 --Enciphered mode of communication.

Ored with one of the below options

0x20 --MIFARE Classic contactless IC mapping support enabled.

0x40 --Secure Dynamic Messaging and Mirroring support enabled.

0x80 --Additional Access Rigths enabled.

pAccessRights: Access Rights, 2 bytes.

pFileSize: File size in bytes for the file to be created, 3 bytes and LSB first.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	006D	N byte	2 byte	NULL

5.8.23 Create Value File

Creates files for the storage and manipulation of 32bit signed integer values within an existing application on the PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	006E	34 byte

Data:

The following length and contents are before conversion to hexadecimal strings

Packet data is bFileNo, bCommSett, pAccessRights, pLowerLmit, pUpperLmit, pValue, bLimitedCredit.

bFileNo:File number of the file to be created, 1 byte.

ORed with **0x80** to indicate secondaryary application indicator.

bCommSett:Communication settings for the file, 1 byte.

0x00 --Plain mode of communication.

0x01 --MAC mode of communication.

0x03 --Enciphered mode of communication.

Ored with one of the below options

0x20 --MIFARE Classic contactless IC mapping support enabled.

0x80 --Additional Access Rigths enabled.

pAccessRights: Access Rights, 2 bytes.

PValue: Initial value, 4 bytes and LSB first.

pUpperLimit: Upper limit value, 4 bytes and LSB first.

pLowerLmit: Lower limit value, 4 bytes and LSB first.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	006E	N byte	2 byte	NULL

5.8.24 Create Linear Record File

Creates files for multiple storage of structural similar data, for example for layalty programs within an existing application.Once the file is filled, further writing is not possible unless it is cleared.

Host->Scanner

CMD	Function Code	Data
NFCCMD	006F	N byte

Data:

The following length and contents are before conversion to hexadecimal strings

The packet data is bOption, bFileNo, pISOFileId, bCommSett, pAccessRights, pRecordSize, pMaxNoOfRec.

bOption: Indicates ISO file ID is present or not, 1 byte.

0x01 --ISOFileId is provided and is valid.

bFileNo:File number of the file to be created, 1 byte. ORed with **0x80**to indicate secondaryary application indicator.

pISOFileId: When **bOption**= 0x00, No this parameter. When **bOption**= 0x01, Indicates ISO/IEC 7816-4 File ID for the file to be created, 2 byts. Excluding the following values reserved by ISO: 0x0000, 0x3F00, 0x3FFF, 0xFFFF.

bCommSett: Communication settings for the file, 1 byte.

0x00 --Plain modeof communication.

0x01 --MAC mode of communication.

0x03 --Enciphered mode of communication.

Ored with the below options

0x80 --Additional Access Rigths enabled.

pAccessRights: Access Rights, 2 bytes.

pRecordSize: Size of onesingle record in bytes, 3 bytes and LSB first.Empty record not allowed.

pMaxNoOfRec:Max Number of Records, 3 bytes and LSB first.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	006F	N byte	2 byte	NULL

5.8.25 Create Cyclic Record File

Creating files for multiple storage of structural similar data, for example for logging transactions, within an existing application. Once the file is filled, the PICC automatically overwrites the oldest record with the latest written one.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0070	N byte

Data:
The following length and contents are before conversion to hexadecimal strings
The packet data is bOption, bFileNo, pISOFileId, bCommSett, pAccessRights, pRecordSize, pMaxNoOfRec.
bOption: Indicates ISO file ID is present or not, 1 byte. 0x01 --ISOFileIdis provided and is valid.
bFileNo:File number of the file to be created, 1 byte. ORed with **0x80**to indicate secondary application indicator.
pISOFileId: When **bOption**= 0x00, No this parameter. When **bOption**= 0x01, Indicates ISO/IEC 7816-4 File IDfor the file to be created, 2 byts. Excluding the following values reserved by ISO: 0x0000, 0x3F00, 0x3FFF, 0xFFFF.
bCommSett: Communication settings for the file, 1 byte.
0x00 --Plain mode of communication.
0x01 --MAC mode of communication.
0x03 --Enciphered mode of communication.
Ored with the below options
0x80 --Additional Access Rigths enabled.
pAccessRights: Access Rights, 2 bytes.
pRecordSize: Size of onesingle record in bytes, 3 bytes and LSB first. Empty record is not allowed.
pMaxNoOfRec: Max Number of Records, 3 bytes and LSB first.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0070	N byte	2 byte	NULL

5.8.26 Delete File

Permanently deactivates a file within the file directory of the currently selected application.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0072	2 byte

Data:
The following length and contents are before conversion to hexadecimal strings
data is the file number of the file to be deleted. ORed with 0x80 to indicate secondaryary application indicator.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0072	2 byte	2 byte	NULL

5.8.27 Read Data

Read data from standard data files or backup data files

Host->Scanner

CMD	Function Code	Data
NFCCMD	0073	18 byte

Data:
The following length and contents are before conversion to hexadecimal strings
The data packet data is bOption, bIns, bFileNo, pOffset, pLength.
bCommOption:Communication settings for the file, 1 byte.

0x00 --Plain mode of communication.
0x10 --MAC mode of communication.
0x30 --Enciphered mode of communication.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0073	18 byte	2 byte	N byte

Card Data:
If more data is to be read, the status **0x71** is returned, then should call This command again with **bCommOption=| 0x02**.

5.8.28 Write Data

Write data to standard data files or backup data files.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0074	N byte

Data:
The following length and contents are before conversion to hexadecimal strings
Packet data is bOption, bIns, bFileNo, pOffset, pTxDataLen, pTxData.
bOption:Communication settings for the file, 1 byte.

0x00 --Plain mode of communication.

0x10 --MAC mode of communication.

0x30 --Enciphered mode of communication.

bIns: One of the below mentioned options, 1 byte.

0x00 --Represent the application chaining format.

0x01 --Represent the ISO14443-4 chaining format.

bFileNo: File number to be writedata, 1 byte.

ORed with **0x80** to indicate secondaryary application indicator.

pOffset: Starting position for the writeoperation, 3 bytes and LSB first.

pTxDataLen:The length of data to be written, 3 bytes and LSB first.

pTxData: Data to be written, the length of bytes is specified by **pTxDataLen**.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0074	N byte	2 byte	NULL

5.8.29 Get Value

Read the currently stored value from value files.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0075	4 byte

Data:

The following length and contents are before conversion to hexadecimal strings

The data package data is bOption, bFileNo.

bOption: Communication settings for the file, 1 byte.

0x00 --Plain mode of communication.

0x10 --MAC mode of communication.

0x30 --Enciphered mode of communication.

bFileNo: File number to be read value, 1 byte.

ORed with **0x80** to indicate secondaryary application indicator

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0075	4 byte	2 byte	8 byte

Card Data:

The following length and contents are before conversion to hexadecimal strings.

Data is the value read out, 4 bytes and LSB first.

5.8.30 Credit

Increase a value stored in a Value File.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0076	12 byte

Data:
The following length and contents are before conversion to hexadecimal strings.
The data package data is bOption, bFileNo, pValue.
bOption:Communication settings for the file, 1 byte.
0x00 --Plain mode of communication.
0x10 --MAC mode of communication.
0x30 --Enciphered mode of communication.
bFileNo: The file number to which the value should be credited, 1 byte.
ORed with **0x80**to indicate secondaryary application indicator.
pValue: Value to be credited, 4 bytes and LSB first.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0076	12 byte	2 byte	NULL

5.8.31 Debit

Decrease a value stored in a Value File.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0077	12 byte

Data:
The following length and contents are before conversion to hexadecimal strings
The data package data is bOption, bFileNo, pValue.
bOption:Communication settings for the file, 1 byte.
0x00 --Plain mode of communication.
0x10 --MAC mode of communication.
0x30 --Enciphered mode of communication.
bFileNo: The file number to which the value should be debited, 1 byte.
ORed with 0x80 to indicate secondaryary application indicator.
pValue: Value to be debited, 4 bytes and LSB first.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0077	12 byte	2 byte	NULL

5.8.32 Limited Credit

Allows a limited increase of a value stored in a Value File without having full credit permissions to the file.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0078	12 byte

Data:
The following length and contents are before conversion to hexadecimal strings
The data package data is bOption, bFileNo, pValue.
bOption:Communication settings for the file, 1 byte.
0x00 --Plain mode of communication.
0x10 --MAC mode of communication.
0x30 --Enciphered mode of communication.
bFileNo: The file number to which the value should be credited, 1 byte.ORed with **0x80** to indicate secondaryary application indicator.
pValue: Value to be credited, 4 bytes and LSB first.

canner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0078	12 byte	2 byte	NULL

5.8.33 Commit Transaction

Validate all previous write access on Backup Data files, value files and record files within one application.

Host->Scanner

CMD	Function Code	Data
NFCCMD	007E	2 byte

Data:
The following length and contents are before conversion to hexadecimal strings
Packet data is bOption.
bOption: One of the below options, 1 byte.

0x00 --Not exchanged to the PICC(EV2).
0x80 --Exchanged to PICC and represent no return of TMC and TMV(EV3).
0x81 --Exchanged to PICC and represent return of TMC and TMV(EV3).

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	007E	2 byte	2 byte	N byte

The following length and contents are before conversion to hexadecimal strings
Card Data:
Results Data were pTMC, pTMV.
pTMC: TransactionMAC Counter (TMC), when **bOption** = 0x80, 4 bytes.
pTMV: Transaction MAC Value (TMV), when **bOption** = 0x81, 8 bytes.

5.8.34 Abort Transaction

Aborts all previous write accesses on Backup Data files, value files and record files within the selected application(s).If applicable, the Transaction MAC calculation is aborted.

Host->Scanner

CMD	Function Code	Data
NFCCMD	007F	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	007F	NULL	2 byte	NULL

5.8.35 Get Config

Perform a Get Config command.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0089	4 byte

Data:
The following length and contents are before conversion to hexadecimal strings
Packet data is wConfig.
wConfig: Configuration to read, 2 bytes. Will be one of the below values
0xA100 --Get additional info of a generic error or some length exposed by interfaces.
0xA200 --Get current status of command wrapping in ISO 7816-4 APDUs.
0xA300 --Get current status of Short Length APDU wrapping in ISO 7816-4 APDUs.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0089	4 byte	2 byte	N byte

Card Data:
data is the value for the mentioned configuration.

5.8.36 Set Config

Perform a SetConfig command.

Host->Scanner

CMD	Function Code	Data
NFCCMD	008A	4 byte

Data:
The following length and contents are before conversion to hexadecimal strings
Packets data are wConfig, wValue.
wConfig: Configuration to set, 2 bytes. Will be one of the below values exposed by interfaces.

0xA200 --Get current status of command wrapping in ISO 7816-4 APDUs.
0xA300 --Get current status of Short Length APDU wrapping in ISO 7816-4 APDUs.

wValue: The value for the mentioned configuration, 2 bytes and LSB first.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	008A	4 byte	2 byte	NULL

Card Data:
data is the value for the mentioned configuration.

5.8.37 Reset Authentication

Reset Authentication status.

Host->Scanner

CMD	Function Code	Data
NFCCMD	008B	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	008B	NULL	2 byte	NULL

5.8.38 Read Sign

Performs the originality check to verify the genuineness of PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0091	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0091	NULL	2 byte	112 byte

Card Data:
The following length and contents are before conversion to hexadecimal strings
data is the plain 56 bytes originality signature of the PICC.

5.9 Mifare classic series card

Operation process of Mifare class card:

- Command mode: request-> anti-collision-> select card-> send command to verify key, read and write-> stop
- Business mode: receive UID-> send command to verify key, read and write-> stop-> find card

5.9.1 Authentication Password

Host->Scanner

CMD	Function Code	Data
NFCCMD	000A	16 byte

Data: Block+ Key type+key
Block: The block number to verify. 2 bytes
Key type: 00: keyA ; 01 keyB 。 2 bytes
Key: 12 bytes

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	000A	16 byte	2 byte	NULL

5.9.2 Read Data

Host –> Scanner

CMD	Function Code	Data
NFCCMD	0003	2 byte

Data: 2 bytes of block number to be read

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0003	2 byte	2 byte	32 bytes

Card Data: read block data

5.9.3 Write Data

Host -> Scanner

CMD	Function Code	Data
NFCCMD	0008	34 byte

Data: Block+ Wrtre Data
Block: 2 bytes
Wrtre Data: 32 bytes

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0008	34 byte	2 byte	NULL

5.10 Ultralight/C/EV1/NTAG21x

Ultralight/C/EV1/NTAG21x card operation process:

- Command mode: request-> anti-collision-> select card-> secondary anti-collision-> secondary select card-> send command-> stop
- Service mode: UID received-> Send command-> Stop-> Find card

5.10.1 Read Data

Read data of specified block/page of Ultralight, UltralightC, UltralightEV1, NTAG213/215/216.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0092	2 byte

Data contains: block address

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0092	2byte	2 byte	32byte

Card Data: is the read block data. Since the data of each block/page is 4 bytes, the card will return data of 4 consecutive blocks/pages. For example, when block = 0, the card will return data of 4 blocks/pages: 0, 1, 2 and 3, totaling 16 * 2 bytes.

5.10.2 Write Data

Write data to designated block/page of Ultralight, UltralightC, UltralightEV1, NTAG213/215/216.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0093	10 byte

Data contains:

block	data
2 byte	8 byte

Block: The block number to be written

Data: data to be written

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0093	10byte	2 byte	NULL

5.10.3 Ultralight C Card Password Verification

Verify Ultralight C Card Password

Host->Scanner

CMD	Function Code	Data
NFCCMD	0094	32 byte

Data: Key

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0094	32byte	2 byte	NULL

5.10.4 Password verification

Verify Ultralight EV1, NTAG213/215/216 card password。

Host->Scanner

CMD	Function Code	Data
NFCCMD	0095	8 byte

Data: Key

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0095	8byte	2 byte	NULL

5.10.5 Obtaining Version Information

Get NTAG21x version information to retrieve information about the NTAG family, product version, storage size, and other product data needed to identify a specific NTAG21x.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0096	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0096	NULL	2 byte	N byte

5.10.6 Reading the counter values

Used to read the current value of the NFC one-way counter for NTAG213, NTAG215, and NTAG216. This command has an argument that specifies the counter number and returns the 24-bit counter value of the corresponding counter.
If the NFC _ CNT _ PWD _ PROT bit is set to 1b, the counter is password protected and cannot be read until a valid password is authenticated.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0097	2 byte

Data: counter assignment number

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0097	2 byte	2 byte	6 byte

5.10.7 Read signature information

Read the signature information of NTAG21x. Returns an IC-specific 32-byte ECC signature to verify that NXP Semiconductor is the silicon supplier. The signature is programmed in the chip production.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0098	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0098	NULL	2 byte	64 byte

Card Data: Signature Information

5.11 ICODE2(ISO15693)

Operation process of ICODE2 card:

- Command mode: Count Label- > Select Label- > Send Command
- Service mode: UID- > Send command- > Find card

5.11.1 Inventory Labels

Count and query the tags within the antenna range and perform anti-collision operation.

Host->Scanner

CMD	Function Code	Data
NFCCMD	0020	6 byte

Data: Afi_flag + AFI+ Slot_flag:
Afi _ flag: whether to match the value of AFI, 00 -- no match, 01 -- match. 2 bytes
AFI: The value of the AFI. If it does not match the AFI, it defaults to 00. 2 bytes
Slot _ flag: Channel Type 0 indicates 16 channels, which can operate multiple cards within the antenna range;
1 means 1 channel, only one card can be operated. 2 bytes.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0020	6 byte	2 byte	N byte

Card Data： len+DSFID0+UID0…DSFIDN+UIDN
Len: Returns the total length of the card information. 2 bytes
DSFID0: The DSFID of card 0. 2 bytes
UID0: UID of card 0. 16 bytes
…
…
DSFIDN: DSFID 2 bytes of card N
UIDN: UID of card N

5.11.2 Select Label

Host->Scanner

CMD	Function Code	Data
NFCCMD	0021	16 byte

Data: UID of the card, obtained by the counting label process

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0021	16 byte	2 byte	NULL

5.11.3 Read block information

Host->Scanner

CMD	Function Code	Data
NFCCMD	0022	N byte

Data includes the following:

StartBlock	select_flag	address_flag	option_flag	UID	Block No
2 byte	2 byte	2 byte	2 byte	16byte	2byte

StartBlock: that star address of the data block to be read

Select _ flag:

00 -- Select the tab to execute the command based on the setting of Address _ flag.

01--Only the card in the selected state can execute this command

Address _ flag:

00 -- The request is not in address mode, the UID is invalid, and any card executes;

01 -- The request is in address mode, and the UID matches. Only the card with a matching UID will execute the command.

Option _ flag:

00 – does not return the security status of the block;

01 -- Returns the security status of the block

UID: The tag UID to read the data from

Block No: The number of blocks to be read

If select _ flag is set to 1, then address _ flag should be 0 and the UID of the card is invalid

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0022	N byte	2 byte	N byte

Card Data：

Len: Total length of returned data: 2 bytes

When option _ flag = 1

Block 1 security status: 2 bytes

Block 1 content: 8 bytes

…

…

BlockN Security Status 2 bytes

BlockN content: 8 bytes

When option _ flag = 0
Block 1 content: 8 bytes
…
…
BlockN content: 8 bytes

5.11.4 Writing block data

Host->Scanner

CMD	Function Code	Data
NFCCMD	0023	N byte

Data includes the following:

StartBlock	select_flag	address_flag	UID	Block No	Data
2 byte	2 byte	2 byte	16 byte	2byte（M）	M*4*2byte

StartBlock: that star address of the data block to be written
Select _ flag:
00 -- Select the tab to execute the command based on the setting of Address _ flag.
01--Only the card in the selected state can execute this command
Address _ flag:
00 -- The request is not in address mode, the UID is invalid, and any card executes;
01 -- The request is in address mode, and the UID matches. Only the card with a matching UID will execute the command.
UID: Tag UID to write data to
Block No: Number of blocks to be written and fetched M
Data: Data to be written M * 4 * 2

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0023	N byte	2 byte	NULL

5.11.5 Permanent Locking Block

Host->Scanner

CMD	Function Code	Data
NFCCMD	0024	22 byte

Data contains:

block	select_flag	address_flag	UID
2 byte	2 byte	2 byte	16 bytes

Block: is the block address to be locked

Select _ flag:

00 -- Select the tab to execute the command based on the setting of Address _ flag.

01--Only the card in the selected state can execute this command

Address _ flag:

00 -- The request is not in address mode, the UID is invalid, and any card executes;

01 -- The request is in address mode, and the UID has a match. Only the card with a matching UID will execute the command.

UID: The label UID of the block to lock

If select _ flag is set to 1, then address _ flag should be 0 and the UID of the card is invalid.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0024	22 byte	2 byte	NULL

5.11.6 Write AFI

Host->Scanner

CMD	Function Code	Data
NFCCMD	0025	22 byte

Data contains:

select_flag	address_flag	UID	AFI
2 byte	2 byte	16 byte	2 byte

Select _ flag:

00 -- Select the tab to execute the command based on the setting of Address _ flag.

01--Only the card in the selected state can execute this command

Address _ flag:

00 -- The request is not in address mode, the UID is invalid, and any card executes;

01 -- The request is in address mode, and the UID has a match. Only the card with a matching UID will execute the command.

UID: The label UID of the block to lock

AFI: The value of the AFI to be written

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0025	22 byte	2 byte	NULL

5.11.7 Locking AFI

Host->Scanner

CMD	Function Code	Data
NFCCMD	0026	20 byte

Data contains:

select_flag	address_flag	UID
2 byte	2 byte	16 byte

Select _ flag:

00 -- Select the tab to execute the command based on the setting of Address _ flag.

01--Only the card in the selected state can execute this command

Address _ flag:

00 -- The request is not in address mode, the UID is invalid, and any card executes;

01 -- The request is in address mode, and the UID has a match. Only the card with 1 matching UID will execute the command.

UID: The label UID of the block to lock

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0026	20byte	2 byte	NULL

5.11.8 Write DSFID

Host->Scanner

CMD	Function Code	Data
NFCCMD	0027	22 byte

Data is structured as follows:

select_flag	address_flag	UID	DSFID
2 byte	2 byte	16 bytes	2 byte

Select _ flag:

00 -- Select the tab to execute the command based on the setting of Address _ flag.

01--Only the card in the selected state can execute this command

Address _ flag:

00 -- The request is not in address mode, the UID is invalid, and any card executes;

01 -- The request is in address mode, and the UID has a match. Only the card with a matching UID will execute the command.

UID: The label UID of the block to lock

DSFID: The value of the DSFID to be written

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0027	22byte	2 byte	NULL

5.11.9 Lock DSFID

Host->Scanner

CMD	Function Code	Data
NFCCMD	0028	20 byte

Data is structured as follows:

select_flag	address_flag	UID
2 byte	2 byte	16 bytes

Select _ flag:

00 -- Select the tab to execute the command based on the setting of Address _ flag.

01--Only the card in the selected state can execute this command

Address _ flag:

00 -- The request is not in address mode, the UID is invalid, and any card executes;

01 -- The request is in address mode, and the UID has a match. Only the card with a matching UID will execute the command.

UID: Tag UID to lock ADSFID

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0028	20byte	2 byte	NULL

5.11.10 Setup EAS

Host->Scanner

CMD	Function Code	Data
NFCCMD	0029	4 byte

Data is structured as follows:

EAS	Request_flag
2 byte	2 byte

EAS:

00 EAS bit set to 0

01 EAS bit set to 1

Request _ flag: is the request flag, and the value is

00 -- the request can be executed by any card;

01--Only the card in the selected state is executed.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	0029	4byte	2 byte	NULL

5.11.11 Locking EAS

Host->Scanner

CMD	Function Code	Data
NFCCMD	002A	2 byte

Data: Request _ flag: It is the request flag, and the value is

00 -- the request can be executed by any card;

01--Only the card in the selected state is executed.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	002A	2byte	2 byte	NULL

5.12 NTAG 42x DNA / TT

NTAG 42x DNA/TT Card Operation Procedure:

- Command mode: request-> anti-collision-> select card-> secondary anti-collision-> secondary select card-> reset-> send command to verify key, read and write-> stop
- Service mode: receive UID-> reset-> send command to verify key, read and write-> stop-> find card

5.12.1 Authentication EV2

Performs a First or Non First Authentication depending upon bFirstAuth Parameter. This will be using the AES128 keys and will generate and verify the contents based on generic AES algorithm.

Host->Scanner

CMD	Function Code	Data
NFCCMD	009B	N byte

Data:

The following length and contents are before conversion to hexadecimal strings

bFirstAuth, **wOption**, **wKeyNo**, **wKeyVer**, **bKeyNoCard**, **bDivLen**, **pDivInput**, **bLenPcdCapsIn**, **pPcdCapsIn**.

bFirstAuth:

Authentication mode,1 byte.

0x00 --Non First Auth in regular EV2 auth Mode.

0x01 --First Auth in regular EV2 auth Mode.

0x02 --Non First Auth in LRP mode.

0x03 --First Auth in LRP mode.

wOption: Diversification option, 2 bytes.

0xFFFF --No diversification.

0x0000 --Encryption based method of diversification.

0x0001 --CMAC based method of diversification

wKeyNo: Key number in keystore, 2 bytes and LSB first.

wKeyVer: Key version in keystore, 2 bytes and LSB first.

bKeyNoCard: Key number on card, 1 byte.

bDivLen: Length of diversification input, 1 byte.

pDivInput: Diversification input, up to 16 bytes. Can be NULL.

bLenPcdCapsIn: Length of PcdCapsIn, 1 byte.Always 0 for following authentication.

pPcdCapsIn: PCD Capabilities. Up to 6 bytes.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	009B	N byte	2 byte	24byte

Card Data：

bPcdCapsOut: PCD Capabilities, 12 bytes.

bPdCapsOut: PD Capabilities, 12 bytes.

5.12.2 Set Configuration

Configures the card and pre personalizes the card with a key, defines ifthe UID or the random ID is sent back during communication setup and configures ATS string.

Host->Scanner

CMD	Function Code	Data
NFCCMD	009C	N byte

Data:

The following length and contents are before conversion to hexadecimal strings

The data packets data are bOption, bDataLen, pData.

bOption:Define length and content of the **pData**, 1 byte.

0x00 --Update the PICC Configuration.

0x04 --Update the Secure Messaging.

0x05 --Update Capibility Data.

0x07 --Update Tag Tamper configuration

0x0A --Update Failed Authentication Counter Configuration.

0x0B --Update Hardware Configuration.

pData:Configuration data for the option specified.The length of bytes is specified by **bDataLen.**

bDataLen:The size of **pData**, 1 byte.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	009C	Nbyte	2 byte	NULL

5.12.3 Get Version

Returns manufacturing related data of the PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	009D	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	009D	NULL	2 byte	N byte

Card Data: Get version information for.

5.12.4 Get Card UID

Return the Unique ID of the PICC. The key must be authenticated before this command.

Host->Scanner

CMD	Function Code	Data
NFCCMD	009E	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	009E	NULL	2 byte	14 byte

Card Data: UID returned by the card.

5.12.5 Change Key

Changes any key on the PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	009F	N byte

Data:

The following length and contents are before conversion to hexadecimal string.

The packets are wOption, wOldKeyNo, wOldKeyVer, wNewKeyNo, wNewKeyVer, bKeyNoCard, pDivInput, bDivLen.

wOption:Diversification option, 2 bytes. The values are one of the below options.

- 0xFFFF
- 0x0002 | 0x0020
- 0x0004 | 0x0020
- 0x0002 | 0x0008
- 0x0004 | 0x0010
- 0x0002 | 0x0004
- 0x0002 | 0x0004 | 0x0020
- 0x0002 | 0x0004 | 0x0008 | 0x0010

The value description:

0xFFFF--No diversification.

0x0002--Diversification of new key required (bit 1).

0x0004--Old key was diversified (bit 2).

0x0008--New key diversification using one rnd (bit 3).

Default is two rounds.

0x0010--Old key diversification using one rnd (bit 4).

Default is two rounds.

0x0020--Key diversification method based on CMAC (bit 5).

Default is Encryption method

wOldKeyNo: Old key number in keystore, 2 bytes and LSB first.

wOldKeyVer: Old key version in keystore, 2 bytes and LSB first.

wNewKeyNo: New key number in keystore,2 bytes and LSB first.

wNewKeyVer: New key version in keystore,2 bytes and LSB first.

bKeyNoCard:Key number of the key to be changed, 1 byte.

pDivInput: Diversification input.Up to 16bytes,can be NULL.

bDivLen: Length of **pDivInput**, 1 byte.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	009F	N byte	2 byte	NULL

5.12.6 Get Key Version

Read out the current key version of any key stored on the PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	00A0	4byte

Data:

bKeyNo, **bKeySetNo**.

bKeyNo: Key number on card, 2byte.

bKeySetNo: Key set number, 2 byte. Optional as it is passed only when bit6 of bKeyNo is set.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	00A0	4 byte	2 byte	N byte

5.12.7 Get File Settings

Get informtion on the properties of a specific file.

Host->Scanner

CMD	Function Code	Data
NFCCMD	00A1	2byte

Data:

data is File number of the targeted file.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	00A1	2byte	2 byte	N byte

5.12.8 Get File Counters

Returns manufacturing related data of the PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	00A2	4byte

The data package data is bOption, bFileNo.

bOption: Indicates the mode of communication to be used while exchanging the data to PICC, 2 byte.

00 --Plain mode of communication.

30 --Enciphered mode of communication.

bFileNo: File number for which the Counter information need to be received, 2 byte.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	00A2	4byte	2 byte	N byte

5.12.9 Change File Settings

Changes the access parameters of an existing file.

Host->Scanner

CMD	Function Code	Data
NFCCMD	00A3	70byte

Data:

The following length and contents are before conversion to hexadecimal strings

The data package data is bOption, bFileNo, bFileOption, pAccessRights, bSdmOptions, pSdmAccessRights, **dwVCUIDOffset**,**dwSDMReadCtrOffset**, **dwPICCDataOffset**,**dwTTPermStatusOffset**,**dwSDMMACInputOffset**, **dwSDMENCOffset**,**dwSDMENCLen**,**dwSDMMACOffset**,**dwSDMReadCtrLimit**.

bOption: Indicates the mode of communication to be used while exchangingthe data to PICC.

0x00 --Plain mode of communication.

0x10 --MAC mode of communication.

0x30 --Enciphered mode of communication.

bFileNo: File number for which the setting needsto be updated, 1 byte.

bFileOption: New communication settings for the file, 1 byte.

0x00 --Plain mode of communication.

0x01 --MAC mode of communication.

0x03 --Enciphered mode of communication.

Oredwith below option

0x40 --Secure Dynamic Messaging and Mirroring is enabled.

pAccessRights: The new access right to be applied for the file, 2 bytes.

bSdmOptions: Secure Dynamic Messaging options. One of the below values to be used.

Can be ORed, 1 byte.

0x80 --Only VCUID is considred for SDM MAC calculation.

0x40 --Only RDCTR is considred for SDM MAC calculation.

0x20 --Indicates the presence of SDM Read Counter Limit.

0x10 --Indicates the presence of SDM ENC File data.

0x08 --Indicates the presence of SDM TT Status.
Must be ored with the above values
0x01 --Indicates the encoding as ASCII.

pSdmAccessRights: The SDM access rights to be applied, 2 bytes.
dwVCUIDOffset: VCUID Offset information, 3 bytes and LSB first.
dwSDMReadCtrOffset: SDMReadCtr value, 3 bytes and LSB first.
dwPICCDataOffset: Mirror position for encrypted PICCData, 3 bytes and LSB first.
dwTTPermStatusOffset: Tag Tamper Permanent Status Offset value, 3 bytes and LSB first.
dwSDMMACInputOffset: Offset in the file where the SDM MAC computation starts, 3 bytes and LSB first.
dwSDMENCOffset: SDMENCFileData mirror position, 3 bytes and LSB first.
dwSDMENCLen: Length of the SDMENCFileData, 3 bytes and LSB first.
dwSDMMACOffset: SDMMAC mirror position, 3 bytes and LSB first.
dwSDMReadCtrLimit: SDM Read Counter Limit value, 3 bytes and LSB first.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	00A3	70byte	2 byte	NULL

5.12.10 Read Data

Read data from standard data files or backup data files.

Host->Scanner

CMD	Function Code	Data
NFCCMD	00A4	18 byte

Data:
The following length and contents are before conversion to hexadecimal strings
The data packet data is bOption, bIns, bFileNo, pOffset, pLength.
bCommOption:Communication settings for the file, 1 byte.
0x00 --Plain mode of communication.
0x10 --MAC mode of communication.
0x30 --Enciphered mode of communication.
bIns: One of the below mentioned options, 1 byte.
0x00 --Represent the application chaining format.
0x01 --Represent the ISO14443-4 chaining format.
bFileNo: File number to be read data, 1 byte.
pOffset: Starting position for the read operation, 3 bytes and LSB first.

If it is 0x000000, Read the entire data file, starting from the position specified in the offset value.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	00A4	18 byte	2 byte	N byte

Card Data:

If more data is to be read, the status 0x71 is returned, then should call This command again with bCommOption=| 0x02.

5.12.11 Write Data

Write data to standard data files or backup data files.

Host->Scanner

CMD	Function Code	Data
NFCCMD	00A5	N byte

Data:

The following length and contents are before conversion to hexadecimal strings

Packet data is bOption, bIns, bFileNo, pOffset, pTxData, pTxDataLen.

bOption:Communication settings for the file, 1 byte.

0x00 --Plain mode of communication.

0x10 --MAC mode of communication.

0x30 --Enciphered mode of communication.

bIns: One of the below mentioned options, 1 byte.

0x00 --Represent the application chaining format.

0x01 --Represent the ISO14443-4 chaining format.

bFileNo: File number to be writedata, 1 byte.

pOffset: Starting position for the writeoperation, 3 bytes and LSB first.

pTxData: Data to be written, the length of bytes is specified by **pTxDataLen**.

pTxDataLen:The length of data to be written, 3 bytes and LSB first.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	00A5	N byte	2 byte	NULL

5.12.12 ISO Select File

Host->Scanner

CMD	Function Code	Data
NFCCMD	00A6	N byte

Data:
The following length and contents are before conversion to hexadecimal strings
Packets data are bOption, bSelector, pFid, pDFName, bDFNameLen, bExtendedLenApdu.
bOption: Indicates whether to return File Control Information (FCI), 1 byte.

0x0C --No return FCI.

bSelector:Selection Control, 1 byte.

0x00 --Select MF, DF or EF, by file identifier.

0x01 --Select child DF.

0x02 --Select EF under the current DF, by file identifier.

0x03 --Select parent DF of the current DF.

0x04 --Select by DF name.

pFid: The ISO File number to be selected, 2 bytes.
Valid only when **bSelector**= 0x00 or 0x02.
pDFName: The ISO DFNameto be selected, up to 16 bytes.
Valid only when **bSelector**= 0x04.
bDFNameLen: The length of DFName, 1 byte. Valid only when **bOption**= 0x04.
bExtendedLenApdu: Flag for Extended Length APDU, 1 byte.

0x01 for Extended Length APDUs.

0x00 or any other value for Short APDUs.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	00A6	N byte	2 byte	N byte

Card Data : data is the returned FCI,Valid only when **bOption**= 0x00.

5.12.13 Read Sign

Performs the originality check to verify the genuineness of the chip.

Host->Scanner

CMD	Function Code	Data
NFCCMD	00A9	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	00A9	NULL	2 byte	112

Card Data: Data is the plain 56 bytes originality signature of the PICC.

5.12.14 Get TT Status

Returns Tag Tamper Status data of the PICC.

Host->Scanner

CMD	Function Code	Data
NFCCMD	00AA	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	00AA	NULL	2 byte	4

data is theTag Tamper Protection status

TTPermStatus	TTCurrStatus
2 byte	2 byte

5.13 Felica

Operation process of Felica card:

- Command mode: Activate Card-> Send Command
- Service mode: UID- > Send command received

5.13.1 Exchange Data

Exchange data with the Picc
Commands need to be supplemented.

5.14 Contactless CPU card operation (ISO14443 TypeB)

Before executing a TypeB card, use the**5.2 Select Protocol**The command selects the current protocol of the reader as TypeB.

5.14.1 Activate Card

Host->Scanner

CMD	Function Code	Data
NFCCMD	00AB	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	00AB	NULL	2 byte	N byte

The following length and contents are before conversion to hexadecimal strings
Card Data: Bytes 2-5 are the serial number of the card.

5.14.2 Reset

Host->Scanner

CMD	Function Code	Data
NFCCMD	00AC	14 byte

The following length and contents are before conversion to hexadecimal strings
Packet data is UID, CID, BrTx, BrRx.
UID:5.14.1 Activate CardUID returned, 4 bytes.
CID: The value is 0-14, which can be used only when the card supports CID, 1 byte. Default 0.
BrTx: default 0 x00, 1 byte.
BrTx: default 0 x00, 1byte.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	00AC	14 byte	2 byte	N byte

5.14.3 Send command

Host->Scanner

CMD	Function Code	Data
NFCCMD	00AD	N byte

NAD (default 00)
CID (default 00)
PCB (default 00)
LEN: (COS instruction length)
Data [4] -DATA [LEN + 4] COS instruction

For example, ISO7816-4 gets the random number
CLA 00
INS 84
P1 00
P2 00

Le 08
Send command: NFCCMD0019000000050084000008

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	00AD	N byte	2 byte	N byte

Card Data: data returned by pass through command.

5.14.4 Stop Card

Host->Scanner

CMD	Function Code	Data
NFCCMD	00AE	8 byte

The following length and contents are before conversion to hexadecimal strings
Data: packet data is**5.14.1 Activate Card** UID returned.

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	00AE	8 byte	2 byte	NULL

5.14.5 Obtain Chinese ID Card UID

Host->Scanner

CMD	Function Code	Data
NFCCMD	00AF	NULL

Scanner ->Host

CMD	Function Code	Data	Status	Card Data
NFCCMD	00AF	NULL	2 byte	N byte

5.15 Mifare plus

Example 1 （Mifare classic）

Read data in block5 area

Step 1: Set the working mode to business mode

Request: NFCMOD2

Response: NFCMOD2<ACK>

Step 2: The Mifare class card is close to the return UID

Device: 1daf2b9a

Step 3: Verify block5 key

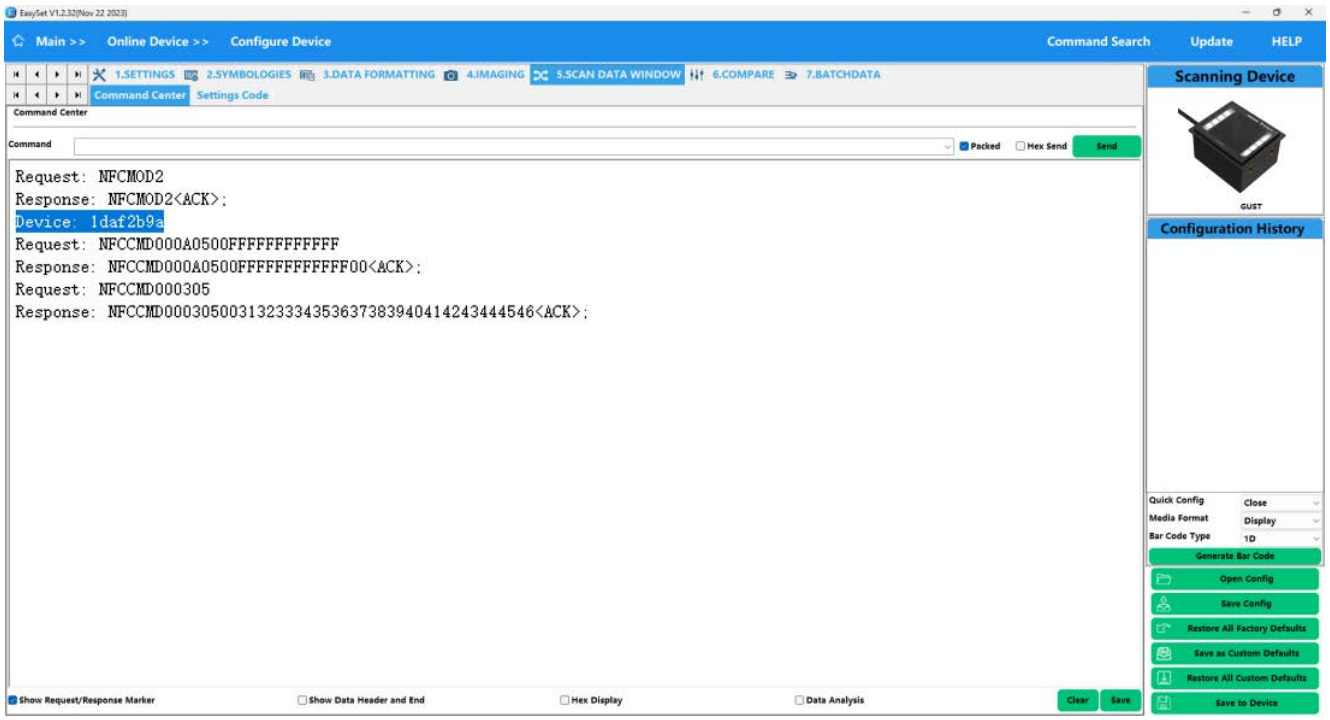
Request: NFCCMD000A0500FFFFFFFFFF key defaults to 0xff, 0xff

Response: NFCCMD000A0500FFFFFFFFFFFF00<ACK>;

Step 4: Read the data of block5

Request: NFCCMD000305

Response: NFCCMD000305003132333435363738394041424344454 6<ACK>;



Example 2 （Mifare desfire EVx)

Read Data :APP ID 0x00001 \File ID 0x00

Step 1: Discovery
Request: NFCCMD0011
Response: NFCCMD001100<ACK>;

Step 2： Card near output UID
Device: 04b44803030180

Step 3： Reset
Request: NFCCMD0018
Response: NFCCMD0018000675b3b00280<ACK>;

Step 4： FormatKeyEntry&SetKey AES128 key hex : 00000000000000000000000000000000
Request: NFCCMD002B00000000
Response: NFCCMD002B0000000000<ACK>;
Request: NFCCMD002C00000000000000000000000000000000000000000000000000
Response:
NFCCMD002C00000000000000000000000000000000000000000000000000<ACK>;

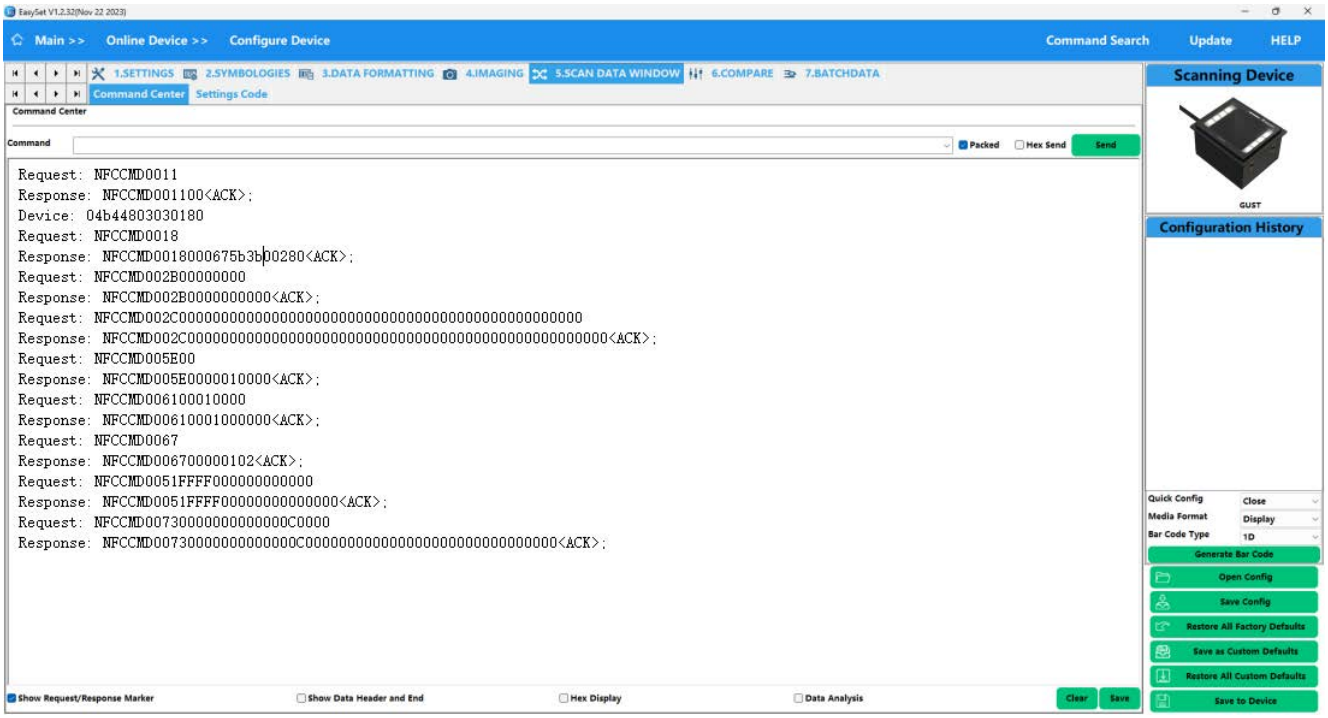
Step 5： Get Application IDs
Request: NFCCMD005E00
Response: NFCCMD005E0000010000<ACK>;

Step 6： Select Application 0x000001
Request: NFCCMD006100010000
Response: NFCCMD00610001000000<ACK>;

Step 7： Get File IDs
Request: NFCCMD0067
Response: NFCCMD006700000102<ACK>;

Step 8： AES Authentication (key store 00 card key 00)
Request: NFCCMD0051FFFF000000000000
Response: NFCCMD0051FFFF00000000000000<ACK>;

Step 9： Read Data File 0x00
Request: NFCCMD00730000000000000C0000
Response: NFCCMD00730000000000000C00000000000000000000000000000000<ACK>;



Example 3 （Mifare desfire Light）

Read Data :FILE ID 0xDF01

Step1: Discovery

Request: NFCCMD0011

Response: NFCCMD001100<ACK>;

Step 2： Card near output UID

Device: 0447587af16780

Step 3： Reset

Request: NFCCMD0018

Response: NFCCMD001800067777710280<ACK>;

Step 4： FormatKeyEntry&SetKey AES128 key hex : 00000000000000000000000000000000

Request: NFCCMD002B00000000

Response: NFCCMD002B0000000000<ACK>;

Request: NFCCMD002C00000000000000000000000000000000000000000000000000000000

Response: NFCCMD002C00000000000000000000000000000000000000000000000000000000<ACK>;

Step 5： ISO Select File 0xDF01

Request: NFCCMD0046000001DF0000

Response:

NFCCMD0046000001DF00000000000000000000000000000000000000000000000000000000000000000000000000000000<ACK>;

Step 6： Authentication EV2

Request: NFCCMD002D01FFFF00000000010000

Response: NFCCMD002D01FFFF000000000100000000000000000000000000000000<ACK>;

Request: NFCCMD002D00FFFF00000000010000

Response: NFCCMD002D00FFFF000000000100000000000000000000000000000000<ACK>;

Step 7： Get File IDs

Request: NFCCMD0033

Response: NFCCMD0033000f1f03000104<ACK>;

Step 8： Get File Settings

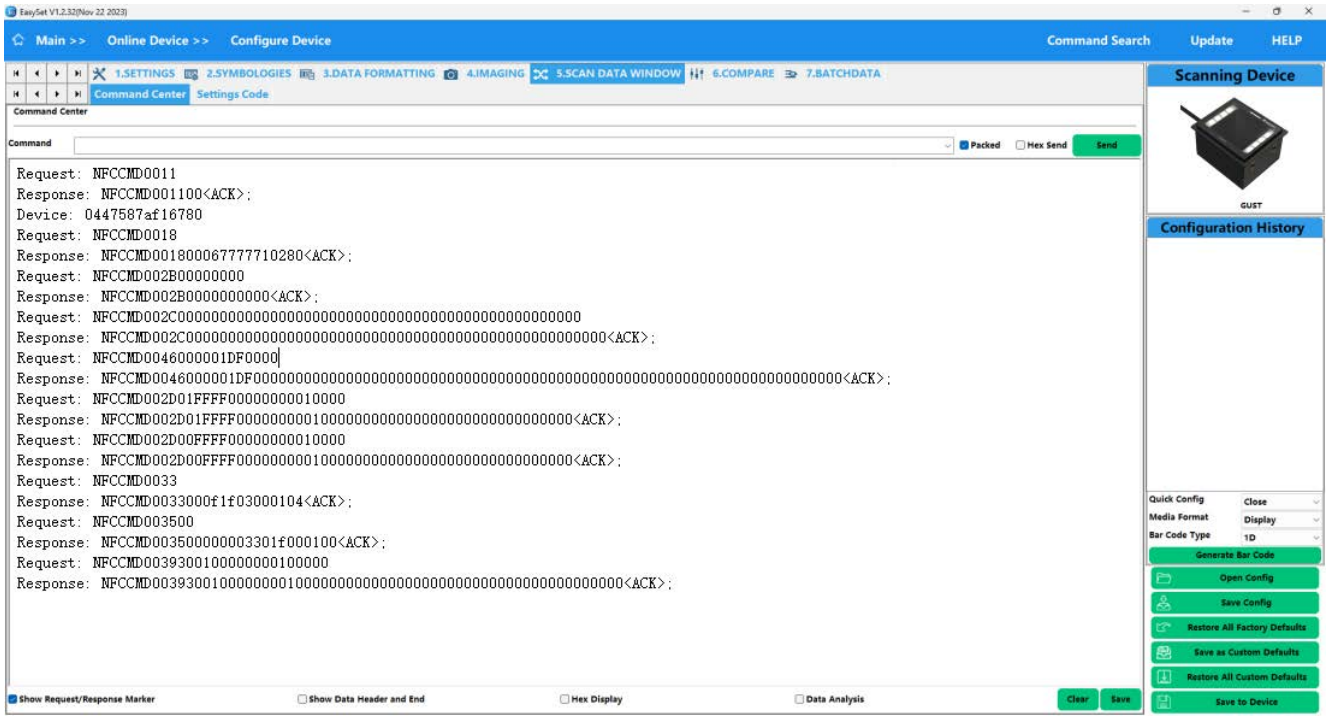
Request: NFCCMD003500

Response: NFCCMD003500000003301f000100<ACK>;

Step 9: Read Data

Request: NFCCMD0039300100000000100000

Response: NFCCMD003930010000000010000000000000000000000000000000000000000000<ACK>;



Example 4（Ultralight/C/EV1/NTAG21x）

Read data(NTAG21x)

Step 1: Discovery
Request: NFCCMD0011
Response: NFCCMD001100<ACK>;

Step 2： Card near output UID
Device: 045de7aa064f80
Request: NFCCMD0095FFFFFFFF
Response: NFCCMD0095FFFFFFFF00<ACK>;

Step 3： Read Data
Request: NFCCMD009200
Response: NFCCMD00920000045de736aa064f806348337cf1323f37<ACK>;

Example 5（NTAG 42x DNA / TT）

Read Data

Step 1: Discovery
Request: NFCCMD0011
Response: NFCCMD001100<ACK>;

Step 2： Card near output UID
Device: 046c492aaa6180

Step 3： Reset
Request: NFCCMD0018
Response: NFCCMD001800067777710280<ACK>;

Step 4: ISO Select File 0xE110
Request: NFCCMD00A60C0010E10000
Response: NFCCMD00A60C0010E1000000<ACK>;

Step 5: FormatKeyEntry&SetKey AES128 key hex : 00000000000000000000000000000000
Request: NFCCMD002B00000000
Response: NFCCMD002B0000000000<ACK>;
Request: NFCCMD002C00000000000000000000000000000000000000000000000000
Response:
NFCCMD002C00000000000000000000000000000000000000000000000000<ACK>;

Step 6： Authentication EV2

Request: NFCCMD009B01FFFF00000000000000

Response: NFCCMD009B01FFFF00000000000000000000000000000000000000000000<ACK>;

Request: NFCCMD009B00FFFF00000000000000

Response: NFCCMD009B00FFFF00000000000000000000000000000000000000000000<ACK>;

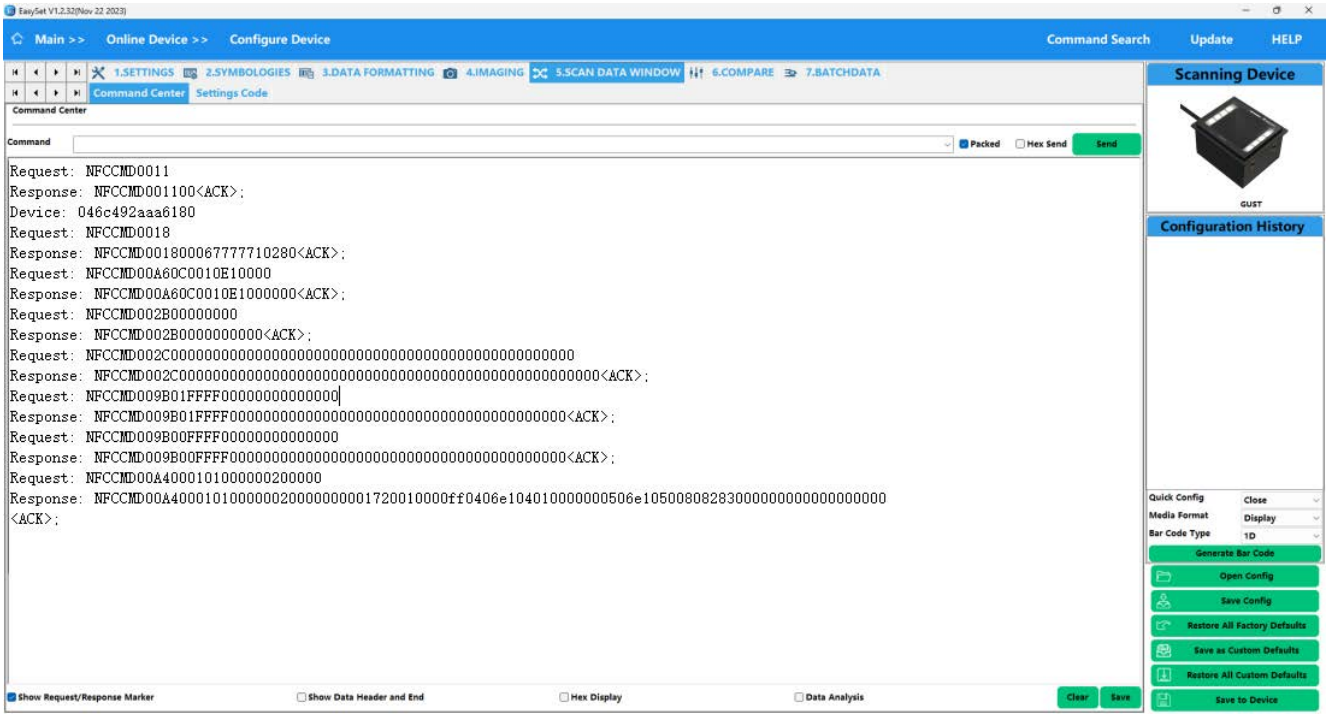
Step 7： Read Data

Request: NFCCMD00A4000101000000200000

Response:

NFCCMD00A400010100000020000000001720010000ff0406e104010000000506e10500808

2830000000000000000000<ACK>;



Example 6 (ICODE2)

Read Data

Step1: Discovery

Request: NFCCMD0011

Response: NFCCMD001100<ACK>;

Step2: Card near output UID

Device: e004015097bda21a

Step3: Select Card

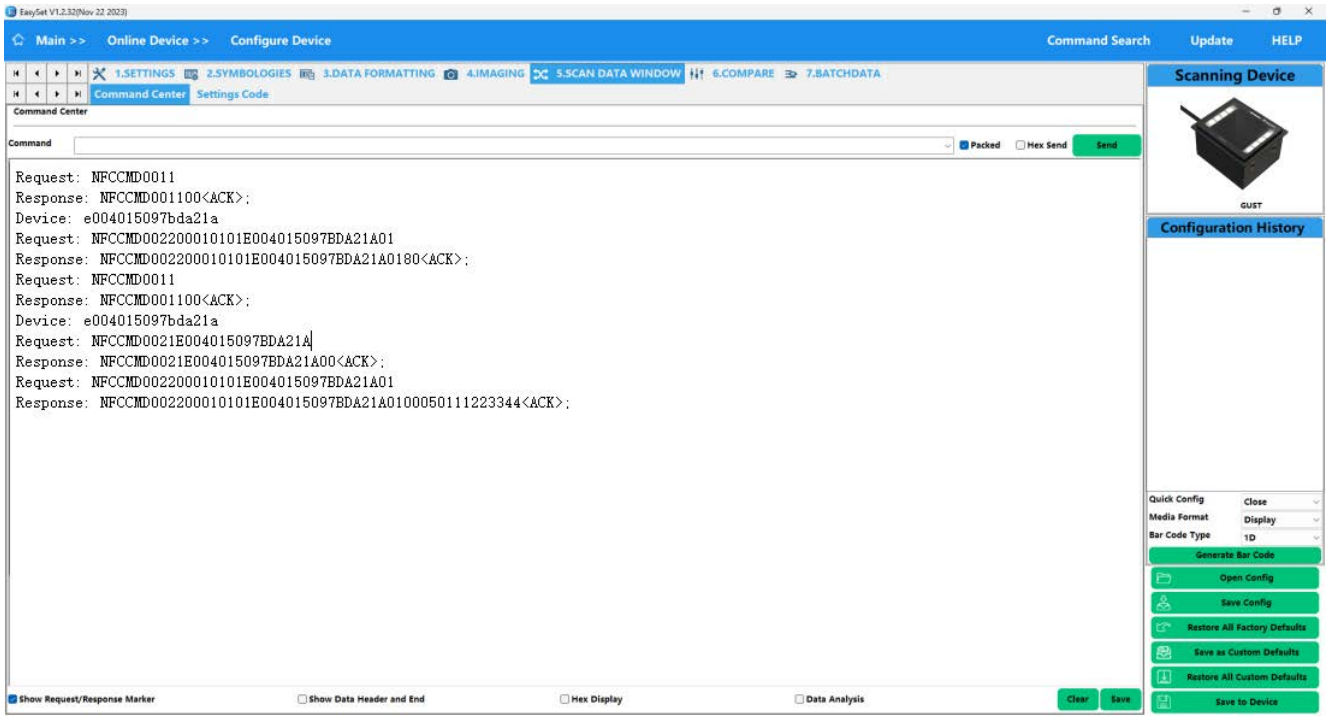
Request: NFCCMD0021E004015097BDA21A

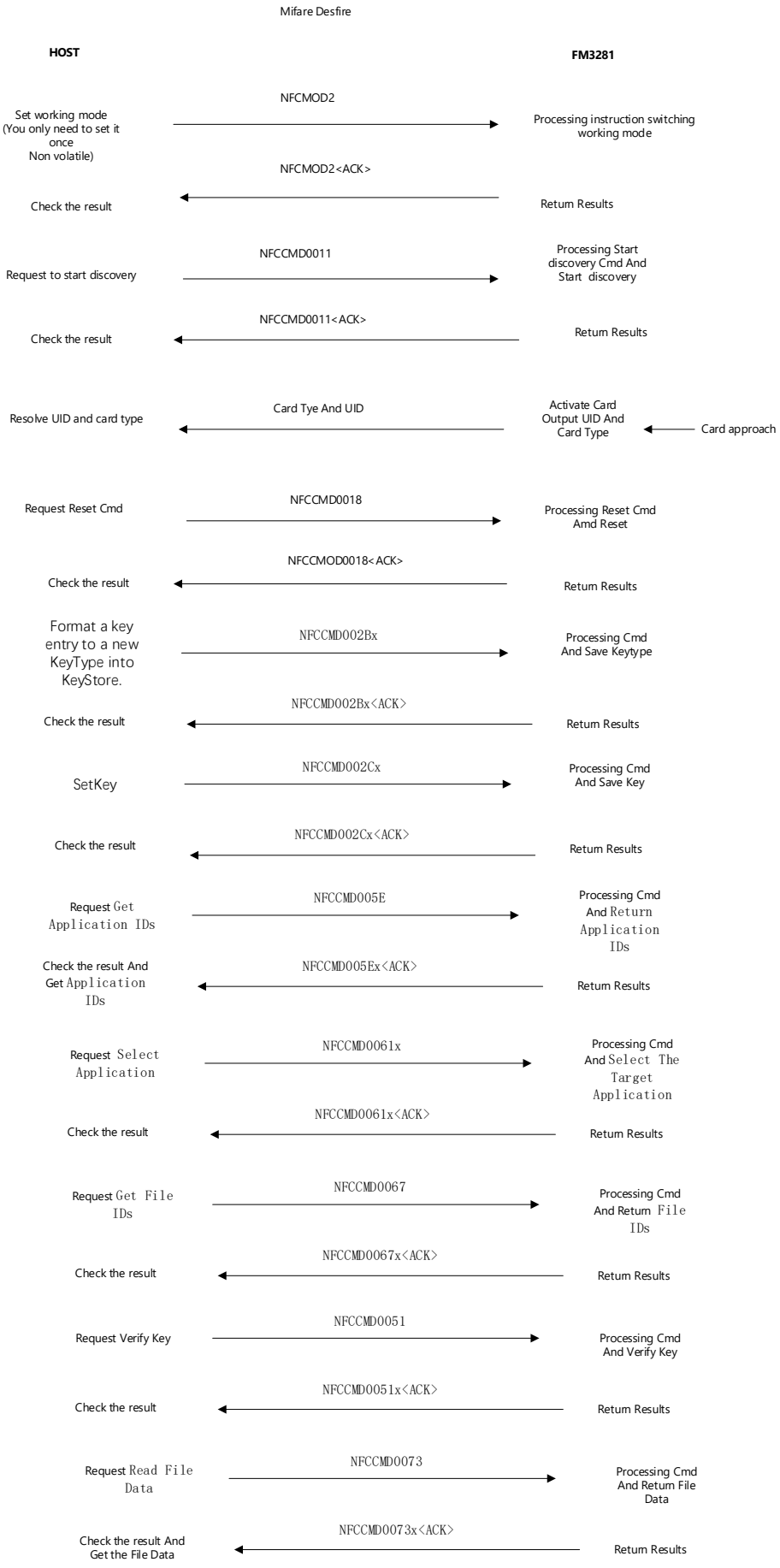
Response: NFCCMD0021E004015097BDA21A00<ACK>;

Step4: Read block

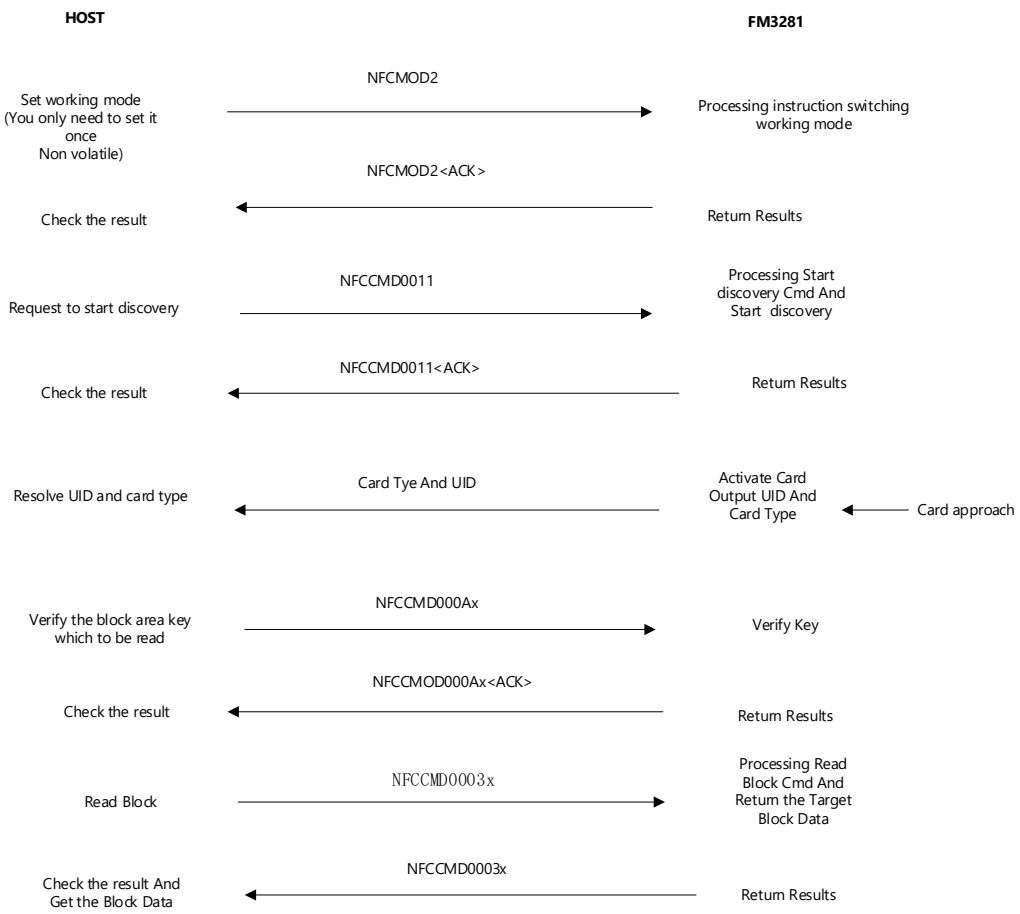
Request: NFCCMD002200010101E004015097BDA21A01

Response: NFCCMD002200010101E004015097BDA21A0100050111223344<ACK>;




Mifare Desfire
HOST
FM3281
NFCMOD2
Set working mode
(You only need to set it
once
Non volatile)
Processing instruction switching
working mode
NFCMOD2<ACK>
Check the result
Return Results
NFCCMD0011
Request to start discovery
Processing Start
discovery Cmd And
Start discovery
NFCCMD0011<ACK>
Check the result
Return Results
Card Tye And UID
Resolve UID and card type
Activate Card
Output UID And
Card Type
Card approach
NFCCMD0018
Request Reset Cmd
Processing Reset Cmd
Amd Reset
NFCCMOD0018<ACK>
Check the result
Return Results
Format a key
entry to a new
KeyType into
KeyStore.
NFCCMD002Bx
Processing Cmd
And Save Keytype
NFCCMD002Bx<ACK>
Check the result
Return Results
SetKey
NFCCMD002Cx
Processing Cmd
And Save Key
NFCCMD002Cx<ACK>
Check the result
Return Results
Request Get
Application IDs
NFCCMD005E
Processing Cmd
And Return
Application
IDs
Check the result And
Get Application
IDs
NFCCMD005Ex<ACK>
Return Results
Request Select
Application
NFCCMD0061x
Processing Cmd
And Select The
Target
Application
Check the result
NFCCMD0061x<ACK>
Return Results
Request Get File
IDs
NFCCMD0067
Processing Cmd
And Return File
IDs
Check the result
NFCCMD0067x<ACK>
Return Results
Request Verify Key
NFCCMD0051
Processing Cmd
And Verify Key
Check the result
NFCCMD0051x<ACK>
Return Results
Request Read File
Data
NFCCMD0073
Processing Cmd
And Return File
Data
Check the result And
Get the File Data
NFCCMD0073x<ACK>
Return Results

Mifare classic



Newland EMEA
+ 31 (0) 345 87 00 33
info@newland-id.com

Rolweg 25
4104 AV Culemborg
The Netherlands

Need more info?
Contact us or one of our partners at
newland-id.com/contact