NFC Forum Type 4
Dual Interface Contact - Contactless
8kB / 32kB / 64kB Flash Based Memory IC

General Description

The NF4 chip is an NFC Forum Type 4 dual interface tag IC. It is intended for use in applications requiring a dual communication interface, contact and contactless, read/write memory with optionally security features to protect data privacy and integrity.

The contactless interface is based on ISO/IEC14443 standard Type A. The chip supports all protocol layers. The robust and sensitive contactless interface permits complete transactions at minimum operating field strength of 0.7A/m. The NFC reader device can select any communication data rates from 106k bps up to 848k bps.

The 2-wires serial contact interface is composed by a bi-directional IO data line and a slave clock (1 clock per data bit).

The chip security features are based on AES-128 cryptography. In order to enforce the confidentiality level of the data exchanged between the reader and the NF4 chip, the contactless communication can be optionally encrypted including also a Message Authentication Code (MAC). The chip maximizes flexibility in terms of access conditions to memory data.

The IC supports all the ISO/IEC14443 -3/-4 commands and a transport layer based on an optimized ISO/IEC 7816-4 command set.

Each NF4 chip has a 7 byte unique serial number, programmed at wafer level, which guaranties the uniqueness of each device.

An optional Random ID feature according to ISO/IEC14443-3 can be enabled in the application or during the product personalization.

In case of a passive application, without battery assistance, the NF4 chip offers energy harvesting capabilities in order to power supply an external device like a microcontroller or a digital sensor.

Features

- NFC Forum Type 4 compliant tag IC
- Supports ISO/IEC14443 Type A
- RF data rates from 106kbps up to 848kbps
- 8kB, 32kB or 64kB user's data NVM memory
- 7 Byte Unique Identifier number (UID)
- Optional Random ID to enhance security and product privacy
- Chip security based on AES-128 crypto algorithm
- Optional Secure Messaging (SM). Encryption of the RF communication channel
- Optional Message Authentication Code (MAC)
- 2-wires serial interface (clock and data).
- Serial interface data rates up to 1MHz
- RF Busy line indicates presence of an on-going contactless communication
- RF field detector available on VPOS
- Energy harvesting capability – extracting energy from the electromagnetic field (up to 5mA – 3.6V)
- Possibility to create proprietary application files
- ISO/IEC7816-4 optimized command set
- On-chip resonant capacitor: 14pF
- Optimized resonant frequency of 14.5MHz
- Minimum operating field strength of 0.7A/m
- -25°C to +85˚C temperature range
- Available in SO8 package or standard or bumped wafers form

Applications

- Healthcare market segment
- NFC pairing
- Smart and Secure metering
- Product personalization and diagnostic tool
- Smart Sensors – Data logger
- NFC tags with large memory capabilities
- Consumer
- Home appliance
- Automation, Industrial

TABLE OF CONTENTS

1 Technical data

1.1 Absolute Maximum Ratings

Parameter	Symbol	Conditions	Min.	Max.	Unit
Power supply VCC-VSS	Vcc		-0.3	+5.7	V
Voltage at remaining pin except C1, C2 inputs	Vpin		Vss-0.3	Vcc+0.3	V
Storage temperature	Tstore	R.H. < 20 %	-55	+ 125	°C
ESD on IO, RST, CLK pins	V_{ESDISO}	HBM reference to substrate VSS, Ileak[1]= +/-100nA at 5.5V, 25°C	-3000	3000	V
ESD on Coil inputs	$V_{ESDCOIL}$	HBM reference to substrate VSS	-3000	3000	V
Maximum AC current induced on C1, C2	I_{coil_RMS}			50	mA

Table 1

Stresses above these listed maximum ratings may cause permanent damage to the device. Exposure beyond specified electrical characteristics may affect device reliability or cause malfunction.

1.2 Handling Procedures

This device has built-in protection against high static voltages or electric fields; however, anti-static precautions must be taken as for any other CMOS component. Unless otherwise specified, proper operation can only occur when all terminal voltages are kept within the voltage range. Unused inputs must always be tied to a defined logic voltage level.

1.3 Latch-up Protections

RST, CLK and IO pads are qualified according to JEDEC 78 class I level A specification.

1.4 Standard operating conditions

Parameter	Symbol	Conditions	Min.	Typical	Max.	Unit
Supply Operating Voltage through VCC pin – Class A	V_{CC}		2.7	5	5.5	V
Supply Operating Voltage through VCC pin – Class B	V_{CC_MV}		2.7	3	3.3	V
Operating temperature	T_{OP}		-25	+25	+85	°C
Default temperature	D_Top			25		°C
Op ext clock frequency at CLK input	F_{ISOCLK}	25 °C			1	MHz
Op ext clock duty cycle at CLK input	Dclk	25 °C	40		60	%
AC peak current induced on C1, C2 in operating conditions	I_{coil_OP}				30	mA

Table 2

[1] Procedure MILSTD-883 D Method3015.7, pin to pin, pin to VSS, pin to VCC.

1.5 2-wires serial contact interface

1.5.1 Power consumption

All tests below were made at D_Top.

Parameter	Symbol	Conditions	Limit values			Unit
			Min	Typ	Max	
NVM Memory Update	$I_{CC\_ERLP\_HV}$	NVM Low power Erase mode VCC = V_{VCC_MAX}		2.4	3.0	mA

Table 3

1.5.2 Contact mode power supply AC Characteristics

Parameter	Symbol	Conditions	GSM spec	Limit Values			Unit
		GPNVM write	Max	Min	Typ	Max	
Maximum charge (q=I*t)	Q_max	3.3V, 330nF, 20 Ohms, 25°C	12		2.5		nA.s
Spike duration	QT_max	3.3V, 330nF, 20 Ohms, 25°C	400		300		ns
Spike magnitude	QA_max	3.3V, 330nF, 20 Ohms, 25°C	60		5		mA

Table 4

1.5.3 Serial Contact mode pins characteristics

1.5.3.1 I/O pin reception mode

When in reception mode, and with the supply voltage (VCC) in the range specified in Table 3, the device shall correctly interpret signals from the host terminal having the characteristics shown in the table below:

Parameter	Symbol	Conditions	Min.	Max.	Unit
H Input Voltage	V_{IH}		0.7 x V_{CC}	V_{CC}+0.3V	V
L Input Voltage	V_{IL}	hv mv	-0.3 -0.3	0.8 0.2 x Vcc	V V
Rise Fall Time	t_R and t_F	Cin =Cout =30pF, Vcc=Vccmax		1.0	µs
Input leakage	IIL	0V <Vin< Vcc, Vcc=Vccmax	-20	20	µA

Table 5

Note: The device shall not be damaged by overshoot or undershoot on the I/O line in the range –0.3 V to V_{CC} + 0.3 V.

1.5.3.2 I/O pin transmission mode

When in transmission mode, the device shall send data to the terminal with the characteristics shown in the table below:

Parameter	Symbol	Conditions	Min	Max	Unit
H Output Voltage	V_{OH}[2]	0 < I_{OH} < 1 mA, V_{CC} = min.	0.7xV_{CC}	V_{CC}	V
L Output Voltage	V_{OL}	0 < I_{OL} < 1 mA, V_{CC} = min. hv, mv range	0	0.4	V
Rise Fall Time	t_R and t_F	C_{IN} (terminal) = 30 pF max. Vcc = Vcc min, 10% to 90%	-	1.0	µs

Table 6

Unless transmitting, the device sets its I/O line driver to reception mode.

There is no requirement for the device to have any current source capability from I/O.

1.5.3.3 I/O pull-up resistor (to VCC pad)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Pull-up Resistance on IO pad	RpupIO	Vcc = max.	20k	40k	70k	Ohms

Table 7

1.5.3.4 Clock (CLK)

With VCC in the specified product range, the device shall operate correctly with a CLK signal having the characteristics shown in the table below:

Parameter	Symbol	Conditions	Min	Max	Unit
H Input Voltage	V_{IH}		0.7 x V_{cc}	V_{cc}+0.3V	V
L Input Voltage	V_{IL}	hv mv	-0.3 -0.3	0.5 0.2 x Vcc	V V
Rise Fall Time	t_R and t_F	V_{CC} = min. to max. Tiso_clk=1µs max	-	9% of clock period	µs
Input leakage	IIL	0V <Vin< Vcc, Vcc=Vccmax	-20	20	µA

Table 8

1.6 NFC & ISO/IEC14443A Contactless Interface

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resonant capacitor	C_{RES}	f = 100kHz U = 100 mVpp	12.3	14	15.7	pF
Coil limiting voltage	V_{LIM}	I_{COILDC} = 10mA	4.05	5	5.75	V
Receive coil limiting voltage	$V_{LIM\_REC\_A}$	I_{COILDC} = 10mA	3.7	4.8	5.7	V
Modulator voltage drop, high current	V_{MOD}	I_{COILDC} = 10mA	1.2	1.65	2.1	V
Non-regulated voltage on VPOS	V_{POS}	I_{COILDC} = 10mA	3.0		3.85	V

Table 9

Timing characteristics for all contactless interfaces are described in appropriate ISO/IEC14443 standards.

1.7 Voltage Monitoring

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
VCC Under Voltage detection threshold	V_{UVD}	T_{OP}	2.4		2.65	V
VCC Over Voltage detection threshold	V_{OVD}	T_{OP}	5.5	5.75	6.3	V

Table 10

1.8 NVM Flash Based Memory parameters

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Endurance Page Erase Byte Program	END_{GPNVM}	25C with page erase retry	100			K cycles
Byte Write Time	T_{GPNVM_WR}	Hardware controlled		20		us

Table 11

2 Hardware Description

2.1 Power supply management

The NF4 is able to operate from two power sources – contact and contactless interfaces. The contact interface is powered from VCC and contactless interface extracts energy from the electromagnetic field received on the antenna coil inputs (C1 and C2).

To achieve stable voltage and to maximize reliability of the non-volatile memory and remove voltage transient effects, the device contains on-chip voltage regulators. Regulators are isolating the device from power supply effects. There are several voltage monitors on-chip to assure that device works in its specified voltage range where full and secure functionality is guaranteed.

2.2 Power Switch

The power from the serial contact interface on VCC has priority over RF. The serial contact interface is considered as master power supply. If VCC drops below VCC min, the NF4 goes to power on reset mode. If there is RF field present at this moment, the NF4 switches to RF field and restarts the device. The boot sequence is restarted and has duration of 650us.

If there isn't VCC power supply at all and the NF4 enters inside RF field, the device is powered from the RF field.

Every time the internal power switch changes power line between VCC and RF the chip is reset for security reasons. The NF4 IC boot time (initialization time) is around 650us for both modes, contact and contactless.

2.3 Serial Contact Interface

The NF4 Flash based non-volatile memory is also accessible through a 2-wires serial contact interface. The serial interface is composed by:

- IO input/output used to send and receive data from a host terminal
- CLK clock input (slave mode). The clock is always provided by the host terminal (microcontroller for ie)

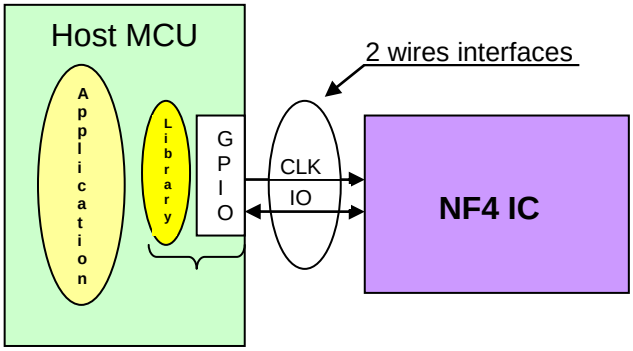
The RST input has to always be connected to the VCC power supply.

The NF4 supports T=0 asynchronous half-duplex character transmission protocol. The bit duration used on the I/O line is defined as an elementary time unit (ETU) and it corresponds to 1 clock on CLK per exchanged bit.

The guard time between two consecutive characters/bytes is of 12 etu.

The NF4 contact interface brings the following benefits:

- Fast data transfer rate with 1CLK/bit
- Data integrity verification based on parity bit
- Tolerant protocol with resend request mechanism
- Multiplexed I /O using bidirectional pad to minimize wires
- No need for external component (IO pull-up already integrated into NF4 IC)



Product range	I/F type	Signals to generate	Host MCU HW resources
NF4-XX	2 wires	CLK, IO	2 GPIO or 2xO + 1xI

2.3.1 Physical layer

Name	IO type [States]	Signal Type	Initial configuration
CLK	Output [High/Low]	Clock	Output Low
IO	Bidir [High/Low/High-Z] + Pull-up[2]	Data	Input High Z

Note 2: NF4 integrates pull-up on IO so that external pull up on IO is optional.

Host MCU GPIO pin assignment

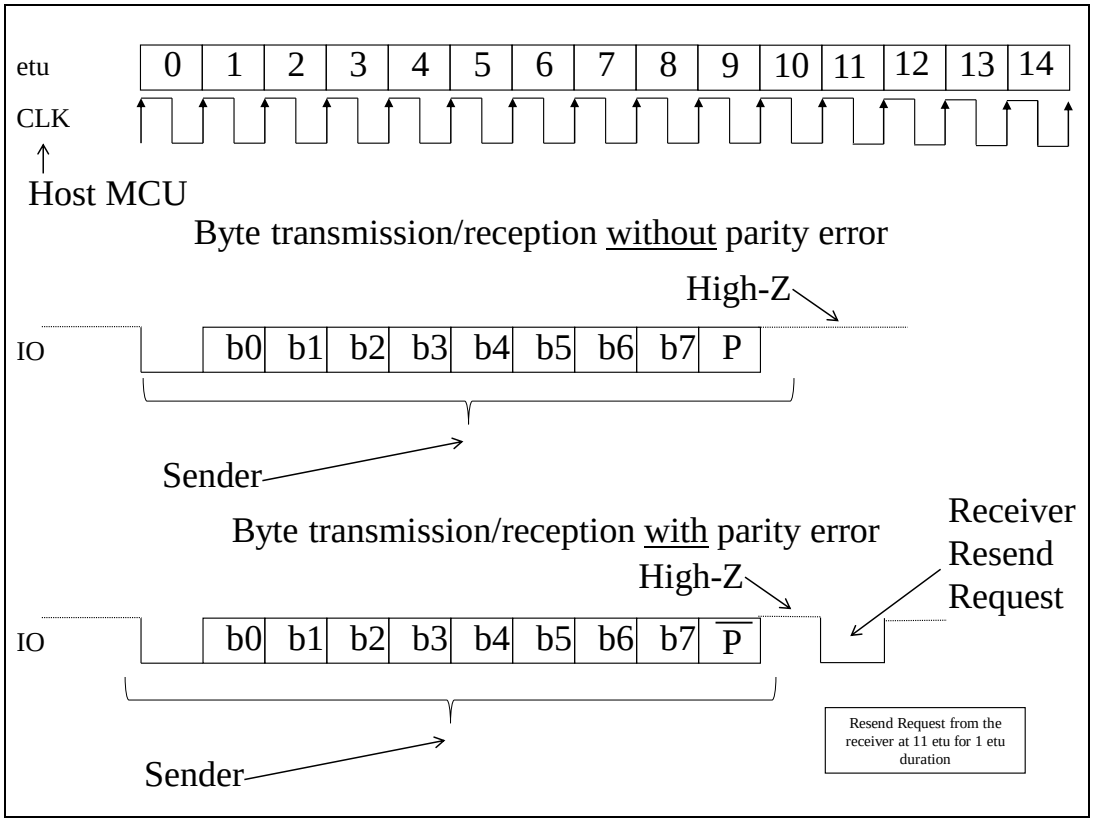
```
#define RST_3w BIT4             // Mapped on SDO : BIT4
#define CLK_2w BIT2             // mapped on SCK : BIT2
#define IO_2w  BIT3             // mapped on SDI : BIT3
```

Elementary operations on pins defined by macro definitions

IO pin example

```
#define mConfigureIO    RegPAOE&=~IO_2w; RegPAInpE|=IO_2w;  // OE=0 pull-up ON
#define mSetIO        RegPADOut|=IO_2w; RegPAOE|=IO_2w;       // set IO=1 and OE=1
#define mClrIO        RegPADOut&=~IO_2w; RegPAOE|=IO_2w;    // set IO=0 and OE=1
#define mReleaseIO    RegPAOE&=~IO_2w; // Disable OE =0  (pull up permanent)
```

2.3.2 Bit duration = 1 CLK



Data in have to be setup at a positive edge of the CLK signal so that they may be correctly read.

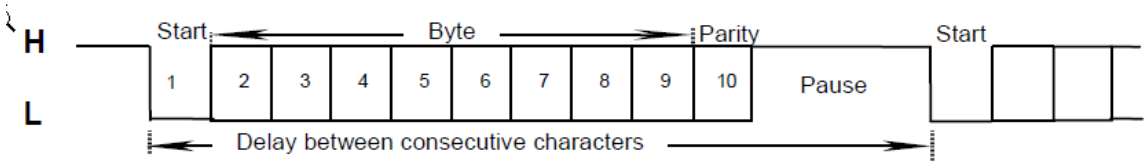
2.3.3 Character format

Character structure is composed of 10 bits (ETU):

- Bit 1: Start bit
- Bit 2-9: 8 bits data frame
- Bit 10: Parity bit even parity
- Pause = Guard time = 2ETU

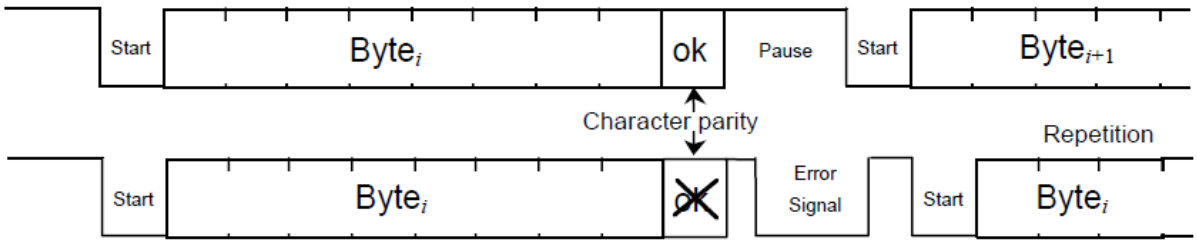
2.3.4 Direct convention LSB first

- Logic '0' = L
- Logic '1' = H



2.3.5 Parity Error Management / Resend request

This mechanism is enabled; an error signal is asserted by receiver allows requesting character repetition from transmitter.



To signal an error, the receiver shall put I/O to state L at (10,5±0,2) etu in receiver time for one etu minimum, two etu maximum.
To detect an error signal, the transmitter shall read I/O at (11±0,2) etu after the character leading edge. The correct reception is assumed if the state is H. The incorrect reception is assumed if the state is L. After a delay of at least two etu from the detection of the error signal, the transmitter shall repeat the character. If either the card or the interface device provides no character repetition, it ignores and shall not suffer damage from the incoming error signal.

Note: The minimum time between two commands on the 2 wires contact interface is about 200us.

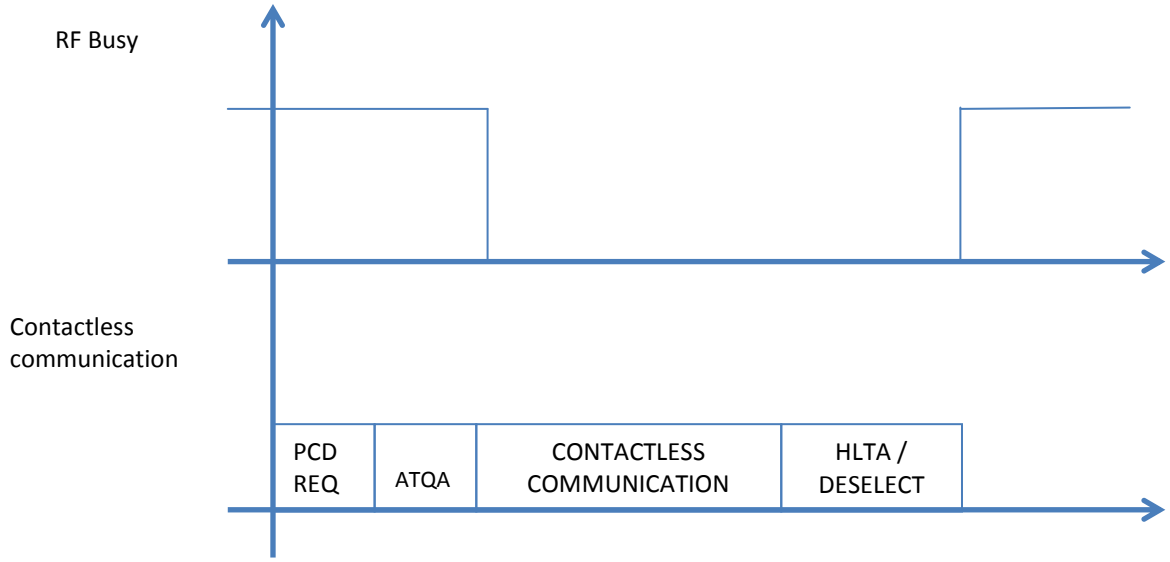
RF Busy line

RF busy line is present on the I/O pad of the serial contact interface. The RF busy signal corresponds to an output indicating to the external terminal that a contactless communication is on-going.

When there is no contactless communication between the PCD and the NF4, the RF busy line is set to a high level.

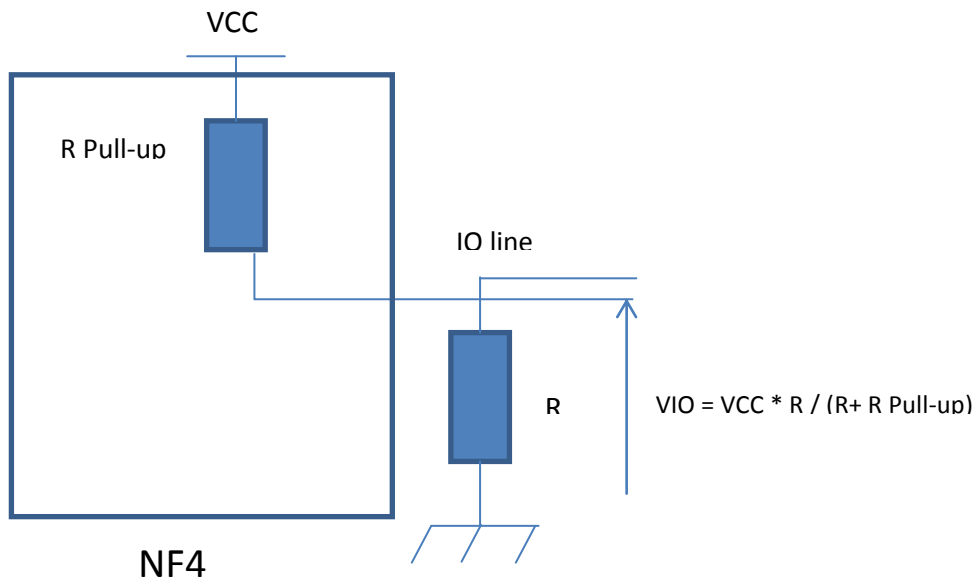
In case that a contactless communication starts between the NF4 and the PCD, the RF Busy line is set to a low level after the NF4 answers by an ATQA.

The RF busy line remains to a low level until the NF4 goes outside the RF field or after receives an HLTA or Deselect A ISO/IEC14443 command.



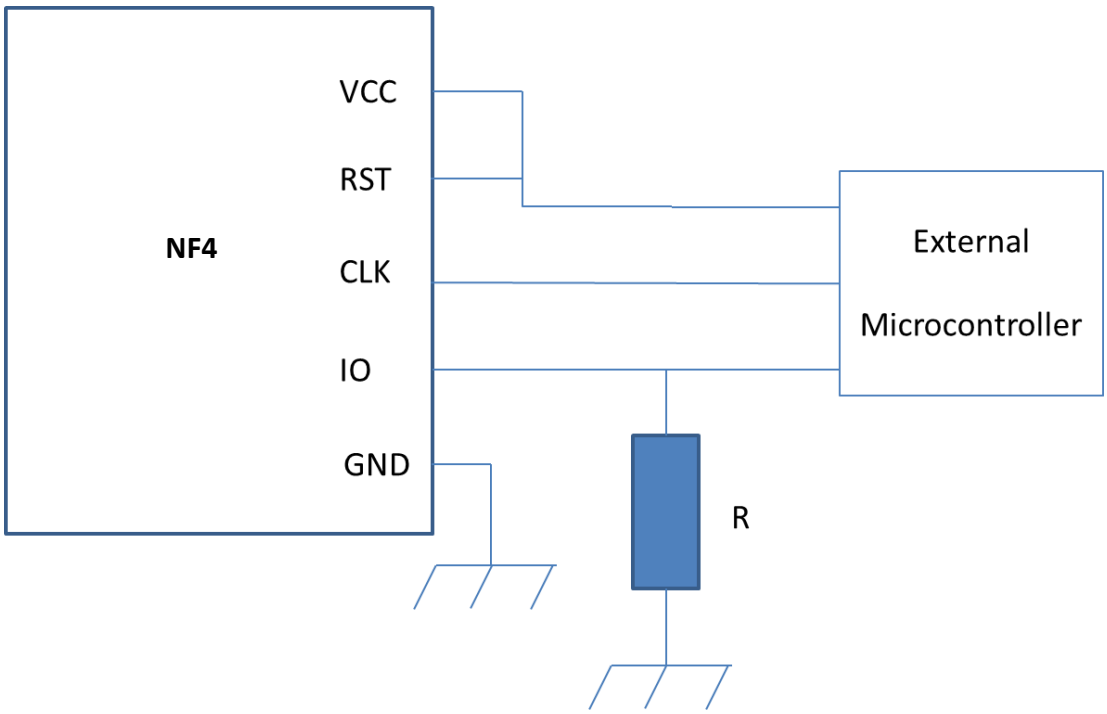
2.3.6 Hardware considerations

It is recommended to add an external pull-down resistor R on the IO line in order to guaranty a low level voltage on IO pad when the internal pull-up of the NF4 chip is disconnected (as indicated below).



Note: R Pull-up resistor typical value is 40kΩ

2.3.7 Typical RF Busy application example



Note: In the above example, the energy is provided by the external microcontroller on VCC pad. The RST input is connected to VCC pad to avoid any mal functionality.

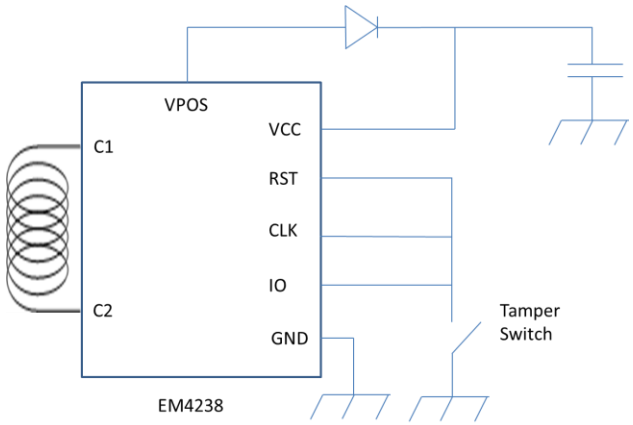
2.4 *Tamper Feature – NF4 chip in RF mode ONLY*

In a RF configuration mode only, the device supports a tamper detection feature that checks the continuity of a loop connected between IO/CLK/RST and GND. Pads RST, CLK and IO has to be all connected together.

Tamper detection can be implemented using a simple continuity loop, with heat sensitive fuse wire, with sensors having both high and low impedance states, or with external devices controlling an electronic switch such as a MOSFET.

The state of Tamper switch can be read via the ISO/IEC7816-4 GET DATA command. It is delivered with a saved initial state set to 1.

When there is tampering, the state value is 1 otherwise 0 (see System Configuration).



Note: The diode between VPOS and VCC pad can be removed in case of pure contactless inlays

3 Unique Identification Number

Each NF4 contains a unique serial ID number programmed at the wafer test level. The UID number guaranties the uniqueness of each device.

The serial ID number is used during an anti-collision procedure between the NF4 chip and the RFID reader (PCD).

It is based on 7-bytes and compliant with ISO/IEC14443 Type A as described below.

3.1 UID format ISO/IEC 14443 Type A

Defined as a 7-bytes serial number cascade level 2 according to ISO/IEC14443-3:

UID0	UID1 UID6
55 48	47 0
IC Mfg Code	**IC manufacturer serial number**

3.2 IC manufacturer serial number

The 48 bit that composed the serial number field contains the following information:

- bit IC Id 0x10 corresponds to the NF4 chip
- 10 bit Customer ID (standard version customer ID corresponds to 1)
- 32 bit incremental Unique serial number

47 42	41 32	31 0
IC Id	**Customer ID**	**32 bit Unique Serial Number**

3.3 Random ID

The support of Random ID feature is implemented according to ISO/IEC14443-3 standard.

4 Memory organization – File system

The file system is implemented on a non-volatile and writable flash based memory.

The logical structure conforming to ISO/IEC7816-4 is a hierarchical file system based on three file types:

- MF (Master File)
- NFC Application Dedicated File (DF) compliant with NFC tag type 4 format
- Optional Elementary File (EF) to store application or proprietary data
- Password File containing a 4-8 bytes password for user authentication
- Key File hosting 2 master keys used for mutual authentication and optional secure messaging

The number of optional elementary files is only limited by the available memory. At delivery, the NF4 chip memory is composed by the MF, the NFC Dedicated File, the Password File and the Key File.

The memory files are allocated by pages of 256 bytes in a contiguous way during the file creation. Each data file is composed by a data area and a system area that contains the following parameters of the file

- Size of the data file
- FID – File Identifier
- SFI – Short File Identifier
- Access conditions

The system area of a data file has a length of 32 bytes. As an example, a file of 224 bytes of user's data will be placed in a memory page of 256 bytes (224 bytes + 32 bytes). In case of 225 bytes of user's data, the corresponding file will be placed in 2 memory pages of 256 bytes. Two files will not share the same memory page which means that the 255 bytes of the second pages are lost.

4.1 MF (Master File)

The MF is the root of the file system and is the one that is selected at start-up or after a reset[3]. NFC DF and EFs reside under the MF.

MF also contains a KEY file, a Password File and default system configuration settings. The KEY file has two records.

MSE: ERASE and MSE: STORE commands allow modifying the key values. CREATE FILE command allows creating EFs under MF. These commands are only possible when either a Mutual Authentication with KEY 1 or a User Authentication with password has been successfully done.

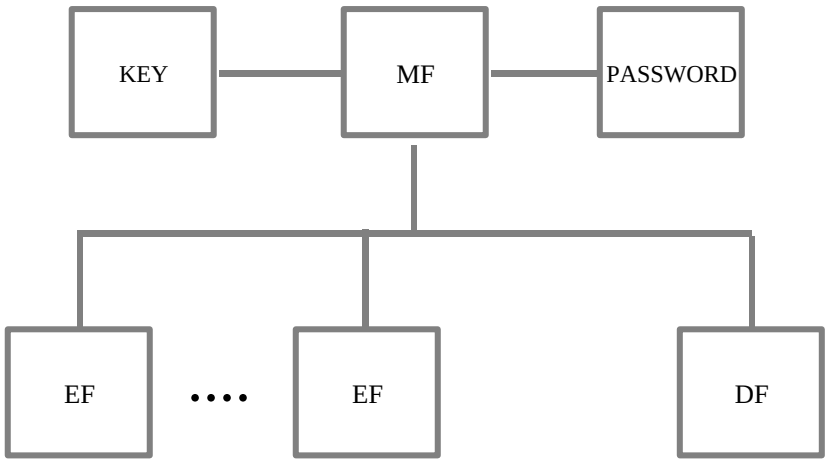


Figure 1

[3] Unless an alternate DF/EF is defined in the system configuration.

4.2 NFC Dedicated File (DF)

4.2.1 NFC Type 4 Tag Structure

The structure defined by the NFC forum for Type 4 tag is shown in Figure 2 and is described below:

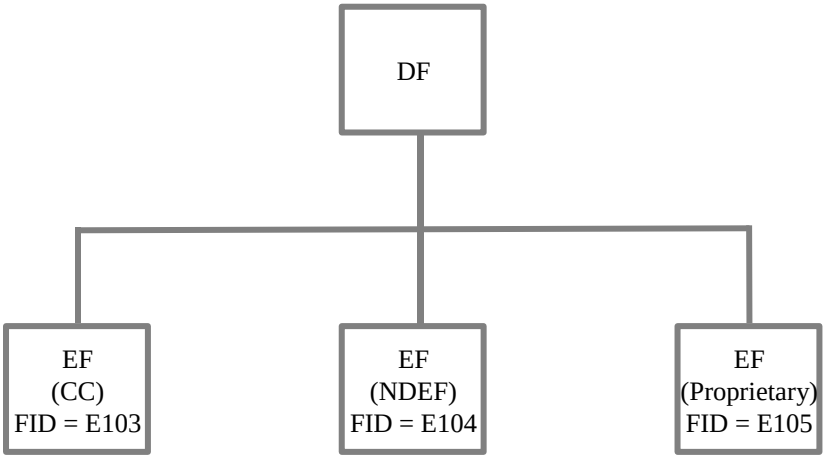


Figure 2

4.2.2 NFC Application DF

The NFC application is contained in a DF with name (AID): D2760000850100 (NFC Tag Type 4 version 1.0).

4.2.3 Capability Container (CC) file

The Capability Container is a transparent EF under the NFC DF with FID equal to E103h.

It can always be read but can never be written. This file can be only modified via System Configuration command with Mutual Authentication using KEY 1.See ref. **[13]** for a complete description of the CC content structure.

CC content (see Table 4 of ref [13]):

Offset	Size	Field	Value (hex)
0000h	2	CLEN	0017
0002h	1	Mapping Version	Defined in System Configuration
0003h	2	MLe	Defined in System Configuration
0005h	2	MLc	Defined in System Configuration
0007h	8	NDEF File Control	see below
000Fh	8	Proprietary File Control	see below

Table 12

The NDEF Control Block TLV has the following structure and default values:

Offset	Size	Field	Value (hex)
0000h	1	T	04
0001h	1	L	06
0002h	2	FID	E104
0004h	2	Msize	PREDEFINED NDEF FILE SIZE[4]
0006h	1	NDEF Read Access	Defined in System Configuration
0007h	1	NDEF Write Access	Defined in System Configuration

Table 13

[4] Size depends on the configuration.

The Proprietary Control Block TLV has the following structure and default values:

Offset	Size	Field	Value (hex)
0000h	1	T	05
0001h	1	L	06
0002h	2	FID	E105
0004h	2	Msize	PREDEFINED PROPRIETARY FILE SIZE
0006h	1	NDEF Read Access	Defined in System Configuration
0007h	1	NDEF Write Access	Defined in System Configuration

Table 14

4.2.4 NDEF message file

The NDEF message file contains the NFC data itself. It contains en empty message. Tag is in INITIALIZED state.

It can always be read. If WRITE access condition byte in NDEF Control Block of CC File equals 00h it can always be written; otherwise either a Mutual Authentication with KEY 1 or a User Authentication is required.

Offset	Size	Field	Value (hex)
0000h	2	NLEN	0003
0002h	3	NDEF message	D00000

Table 15

4.2.5 Proprietary file

The Proprietary message file contains any proprietary data. It is initialized to 00h.

It can always be read. If WRITE access condition byte in Proprietary Control Block of CC File equals 00h it can always be written; otherwise either a Mutual Authentication with KEY 1 or User Authentication is required.

4.2.6 NFC Command set

Following commands from the NF4 chip command set are used for NFC Type 4 Tag commands:

Command		***Description***
NFC	***NF4***	
Select	SELECT	Select of Application (DF) or File (EF)
Read Binary	READ BINARY	Read data from file
Write Binary	UPDATE BINARY	Update (Erase + Write) data to file

Table 16

4.3 EF (Elementary Files)

EFs are used to store application or general purpose data. EFs are selected via the SELECT FILE command.

After a successful update via an update binary command, the new data are permanently written to the file.

The transparent EF has no internal structure and is a sequence of bytes. Information is accessed by an offset from the start of the file and a length (in bytes). The largest possible file size is 32768 bytes.

4.4 File Selection

Before a file can be accessed it must first be selected.

Files can be selected on the basis of a 2-byte file identifier (FID). The FID is searched amongst the current DF. The FID values 3F00h, 3FFFh and FFFFh are reserved, as well as the FID of MF, which has value 3F00h.

An EF can also be selected by short file identifier (SFI). The SFI is optional and has a value in the range 1-30. The SFI must be unique amongst direct neighbours'.

The NF4 chip maintains a Current DF and a Current EF. After reset, by default the Current DF is the MF and the Current EF is undefined. The system configuration allows specifying alternative DF and EF. The Current DF and current EF settings dependent on the communication interface. The Current EF and Current DF established on contact interface will not be changed by selection on the contactless interface.

When a DF (or MF) is selected, it becomes the new Current DF and the Current EF is undefined. When an EF is selected, it becomes the Current EF. The Current DF is never undefined; the Current EF may be undefined.

The FID must be unique amongst the immediate children of the Current DF.

4.5 Special Files

4.5.1 Password File

A Password file is a file holding the required information for user authentication. It resides under MF. Password length can be between 4-8 bytes.

At chip delivery, the PASSWORD value is: 00000000 (Hexadecimal)

4.5.2 KEY File

The KEY file holds key information. It resides under MF and has space for two AES-128 keys. These keys can be used for security related actions such as authentication and encryption. The meaning and structure of the keys is described in KEY File Management.

At chip delivery, the master key values are:

MF KEY 1: 9B475F50C612B0A7E5C44629DCDE6AEE (Hexadecimal)

MF KEY 2: 00000000000000000000000000000000 (Hexadecimal)

5 Security System

The NF4 chip presents optionally a secure environment where information can be stored and accessed in a secure way.

There are two aspects to the security. On the one hand there is the exchange of information (communication) between the contactless interface device and the NF4 chip; on the other hand there is the storage of the information within the NF4 chip itself and the rules for accessing and modifying it.

Communication is optionally protected using Secure Messaging (SM) offering the encryption of the RF communication channel, while data access is protected by Access Conditions (ACs).

The Access Conditions conceptually follow ISO/IEC7816-9 (ref **[5]**).

5.1 Access Conditions

Each file has defined a set of Access Conditions (ACs), which is a set of conditions that must be satisfied before access is granted. ACs are coded as an array of three bytes indexed from 0-2. Each AC byte defines the security requirements for a specific command or type of command. The ACs are defined when the file is created and cannot be modified afterwards.

Each AC byte is coded as indicated in Table 17 and Table 18.

Special values:

value	Meaning
00	ALWAYS (=no condition applies)
FFh	NEVER (=never grant access)

Table 17

For other values the following bit definitions apply:

bit	Meaning
7	0=either one of the conditions must be satisfied, 1=all conditions must be satisfied
6	1=mutual authentication is required (PCD must have knowledge of the secret AES-128 Key)
5	1=user authentication is required (i.e. PCD must have entered a correct Password)
4-1	RFU
0	key index[5] used for Mutual Authentication

Table 18

5.1.1 Access Conditions for MF

The Access Condition bytes for MF apply to the following commands:

AC byte	command for which the AC applies
0	CREATE FILE (EF)
1	MSE: ERASE, MSE: STORE
2	RFU

Table 19

[5] index = record - 1

5.1.2 Access Conditions for DF

The Access Condition bytes for the NFC DF apply to the following commands:

AC byte	command for which the AC applies
0	RFU
1	RFU
2	RFU

Table 20

5.1.3 Access Conditions for EF

The Access Conditions for an EF apply to the following commands:

AC byte	command for which the AC applies
0	READ
1	UPDATE
2	RFU

Table 21

5.2 Security Environment

A security environment specifies what authentications have been done and thus encompasses:

- User authentication using the 4-8 bytes password
- Mutual authentication (Secure Messaging)

5.2.1 User authentication

The NF4 user authentication is by means of a 4-8 bytes password. The password functionality corresponds to chip secure mode level 0.

Use the VERIFY PASSWORD command to authenticate and enter in the chip secure mode level 0.

If the Password value only contains hexadecimal zeros (irrespective of the length), the Password is considered to be non-active. Any Access Condition requiring User Authentication will then be automatically fulfilled.

There is no limit on the number of Password verification attempts.

CHANGE PASSWORD command permits to update the password with a new value defined in the command parameters.

5.2.2 Mutual authentication (Secure Messaging)

Secure Messaging (SM) is a mechanism to protect the exchanges between the contactless device (PCD) and NF4 chip. It uses a dynamically established session key.

SM is established by means of the MUTUAL AUTHENTICATE command. It requires shared knowledge of a secret key value between PCD and the NF4 chip. This condition applies for a well-defined key of the KEY file. The Access Conditions specify which key in the KEY file must be used.

The MUTUAL AUTHENTICATE command also defines what security (data encryption in command and/or response) if any must be used when using SM.

5.2.3 Validity of Security Context

A security context is the set of authentications that were done and that are still valid. A security context thus encompasses user authentication and mutual authentication.

The security environment only applies to Elementary Files (EF) created under MF and that are accessed via the contactless interface[6]. EF access via the contact serial interface is always unrestricted.

5.3 Access Conditions error code

When the NF4 recognizes an Access Condition error while interpreting a command, the status bytes 6982h are returned.

[6] For the Access Conditions in the NFC files, see NFC Dedicated File (DF).

6 Initial Settings

6.1 Default Keys and Password

The first key in the KEY file is an AES-128 predefined key. The first Key is called Transport Key and at delivery it has the preset value: 9Bh 47h 5Fh 50h C6h 12h B0h A7h E5h C4h 46h 29h DCh DEh 6Ah EEh.

The second key is an AES-128 key initialized to all hexadecimal zeros.

The default Password value is 4 hexadecimal zeros.

6.2 Default ATS

The structure of the ATS is defined in ISO/IEC7816-3. It consists of two parts:

- The first part indicates hardware and protocol options.
- The second part depends on the protocol used. Its length can be at most 15 bytes and is referred to as 'historical bytes'.

The total length of the ATS is limited to 32 bytes.

6.2.1 ISO/IEC14443 Type A

In case of contactless protocol ISO14443-4 type A the default historical bytes are as follows:

Byte	Meaning
1	'N'
2	'F'
3	'4'
4	Version (in ASCII)
5	Release (in ASCII)

Figure 3

The actual historical bytes used in the ATS can be configured (see System Configuration).

7 Exchanges between Card and Interface Device

Exchanges between an Interface Device (PCD) and the NF4 chip are called Application Protocol Data Units (APDUs).

The NF4 chip never takes the initiative to send an APDU; it only replies to a previous APDU command sent by the PCD or the external microcontroller connected on the serial contact interface. The PCD or external microcontroller cannot send another command until it received a response from the NF4 chip. Exchanges between PCD and NF4 chip are thus half-duplex.

7.1 APDU structure

APDU commands have the structure:

Byte	Name	Meaning
1	CLA	Class
2	INS	Instruction – indicates which NF4 chip function must be performed
3	P1	Parameter 1
4	P2	Parameter 2
5	Lc	Number of data bytes
6	Data	Data bytes in command
N	Le	Number of data bytes expected in response

Table 22

APDU responses without data have the structure:

Byte	Name	Meaning
1	SW1	Status byte 1
2	SW2	Status byte 2

Table 23

APDU responses with data have the structure:

Byte	Name	Meaning
1..Lr	Data	Data returned by card
Lr+1	SW1	Status byte 1
Lr+2	SW2	Status byte 2

Table 24

7.2 Command Class Byte (CLA)

The class byte (CLA) is used to indicate how the command otherwise identified by the instruction byte (INS) should be used to select the handling in the card.

The exact encoding of CLA is defined in ISO/IEC7816-4. The upper nibble specifies whether the command is defined by ISO or not and whether it follows or not the above structure rules. NF4 chip uses two possible values for CLA:

0Xh indicates that the command is defined by ISO

8Xh indicates that the command is not specified by ISO but follows the above structure rules

The meaning of the lower nibble (Xh) is as follows:

Bits 0-1 indicate the logical channel. NF4 chip does not support logical channels; hence these bits should always be zero.

Bits 2-3 indicate whether Secure Messaging (SM) is used and in what form. NF4 chip uses command header authentication, hence both bits should equal 1.

From this we have that the lower nibble of CLA takes the value 0h when no SM is used and Ch when SM is used.

7.3 Command instruction byte (INS)

The instruction byte (INS) selects the function to be executed in the card. Note that the selection of the function may depend on the class byte (CLA) as well. Refer to the section describing the individual commands.

7.4 Parameters P1 and P2

The parameters P1 and P2 are function specific. Refer to the section describing the individual commands.

7.5 Data

The data sent with commands or received in responses are function specific. Refer to the section describing the individual commands.

7.6 Lc, Le and Lr

Lc, Le and Lr indicate lengths. The byte Lc indicates the actual length of the data to send in a command. If there are no data, Lc is missing. The byte Le indicates the (maximum) number of data bytes expected in a response. If Le is missing, no data are expected. If Le is zero, it means that up to 256 bytes are expected.

Lr indicates the actual number of data bytes received from the card. Lr is always less than or equal to Le, unless Le is zero when Lr must be less than or equal to 256.

7.7 SW1 SW2

The status bytes indicate the result (return code) of the command execution in the card.

A successful response has a return code 9000h. All defined values of SW1 SW2 are listed.

7.8 APDU message structure

As can be inferred from the command structure, the following four cases are identified:

Case	Lc	Le
1	missing (no data)	missing (no data)
2	missing (no data)	yes (data)
3	yes (data)	missing (no data)
4	yes (data)	yes (data)

ISO/IEC14443-4 and T=1 support all four cases.

The T=0 protocol supports data in one direction only (i.e. cases 1, 2 and 3), hence case 4 commands are not allowed. For case 4, ISO/IEC7816-3 has defined the GET RESPONSE command, which is used to retrieve the response data. When T=0, commands that have both incoming and outgoing data will not return data in the response. Any pending output data will be signalized by a 61XXh return code, where XX is the number of data bytes waiting to be retrieved via a GET RESPONSE command. The Le value may not be specified in the command.

For case 4 and T=0, Le may not be specified, whereas in the case of ISO/IEC14443-4 (T=CL) Le must be specified.

The following commands are case 4: SELECT FILE and MUTUAL AUTHENTICATE.

When Secure Messaging is used, the command structure is always case 4.

For T=0 and case 2 commands (e.g. READ BINARY) and when Le is larger than the data available, the return code is 6Cxx, where XX is the maximum number of data bytes that can be returned.

8 Transport Protocol

8.1 Contact mode

The NF4 chip supports T=0 (ref. [1]) with a baud rate of 1 clock per etu.

After reset, the NF4 chip will not send an ATR on the contact interface. Neither does it support PPS.

8.2 Contactless mode

The NF4 chip supports ISO/IEC14443-3/4 Type A commands.

8.2.1 ISO/IEC14443-3 Type A

Command	Description
REQA	REQA and the returned ATQA are implemented fully in accordance with ISO/IEC14443-3. Supported UID sizes are 7 or 0 for a random UID of 4 bytes. The proprietary coding bits in ATQA are zero.
WUPA	WAKE UP is implemented fully in accordance with ISO/IEC14443-3.
ANTICOLLISION / SELECT Cascade Levels 1, 2 and 3	The ANTICOLLISION and SELECT commands and the responses by PICC are implemented fully in accordance with ISO/IEC14443-3.
HLTA	The HLTA command is used to set the IC into a different wait state (halt instead of idle), enabling devices whose UIDs are already known because they have passed the anti-collision procedure, to be separated from devices yet to be identified by their UIDs. This mechanism is a very efficient way of finding all contactless devices in the PCD field.

Table 25

8.2.2 ISO/IEC14443-4 Type A

Command	Description
RATS	The response to the RATS command identifies the PICC type to the PCD. The PICC frame size is 128 bytes. The historical bytes in ATS response can be configured. CID is supported. NAD is not supported.
PPS	The PPS command allows an individual selection of the communication baud rate between PCD and PICC. Communication baud rates can be set independently for both directions. Supported rates are (kbit/s): 106, 212, 424 and 847.
DESELECT	DESELECT is implemented fully in accordance with ISO/IEC14443-4.
I-block	I-block is implemented fully in accordance with ISO/IEC14443-4.
R-block	R-block is implemented fully in accordance with ISO/IEC14443-4.
S-block	S-block is implemented fully in accordance with ISO/IEC14443-4. In particular if the PICC needs more time than the defined FWT to respond to a PCD command it will send a request for a waiting time extension using S(WTX).

Table 26

9 KEY File Management

9.1 KEY file entries

In a KEY file each record entry consists of a 3-byte header followed by the key value.

The header has the following structure:

Mnemonic	Length	Meaning
KEY_VERSION	1	Key version
KEY_TYPE	1	Type of key
KEY_SIZE	1	Size of key value

Table 27

9.1.1 Key version

The version field can be freely defined.

9.1.2 Key type

The type field specifies the kind of key. The following key types are can be used:

Mnemonic	Value	Meaning
KEY_T_AES16	C0h	AES with 16-byte key (AES-128)

Table 28

9.2 KEY file update

Keys in a KEY file are managed by the Manage Security Environment (MSE) commands (ref **[4]**)

- To insert a key, use the MSE: STORE command.
- To erase a key, use the MSE: ERASE command.
- To obtain information about a key use MSE: GET INFO.

The use of MSE commands is subjected to the relevant MF Access Condition.

10 Security Mechanisms

The NF4 chip supports AES (ref **[8]**) crypto algorithm. This section details how these keys are used to encrypt/decrypt and compute a MAC.

10.1 Key Verification Value

To be able to verify a key without revealing its value one uses the notion of Key Verification Value, which is defined as the encryption of a zero value[7] with the key:

KVV = ENC(0)

10.2 Padding

Padding is sometimes required to make the length a multiple of the block length of the security algorithm (16 for AES-128). To pad, first add a byte with value 80h optionally followed by bytes of value 00h such that the total length becomes a multiple of the block length.

The use of padding can be selected via system configuration (see System Configuration). By default there is no padding when the original length is a multiple of the block length.

10.3 Cryptographic Checksum

A Message Authentication Code (MAC) can be thought of as a checksum of the APDU, encrypted with a key that is common to PCD and NF4 chip. It protects against modification of the APDU.

In most cases only part of the MAC is transmitted, called Cryptographic Checksum (CC), thus limiting the knowledge about the session key.

The MAC computation uses the CMAC algorithm of NIST 800-38B (ref **[9]**).

The session key K_{MAC} are AES-128 based. The following is a succinct description. For a detailed description, please refer to ref **[9]**.

Input is a data field from which one constructs the message:

IV || data || padding = M1 || M2 || …. || Mn

Mi are blocks of 16-bytes (AES-128).

IV is an Initial Vector as defined in Secure Messaging.

Output from one encryption is XORed with the next data block. The last loop adds an extra sub-key in the XOR using either K2 or K1, depending on whether there was or was no padding respectively.

The sub-keys K1 and K2 (using the terminology of NIST 800-38B) are derived from K_{MAC} via the following algorithm:

- Define R=87h if K_{MAC} is of type AES-128
- Let L = ENC(0), i.e. L equals the key verification value of key K_{MAC}
- If the most significant bit of L is zero then K1 = L<<1, else K1 = (L<<1) $\oplus$ R
- If the most significant bit of K1 is zero then K2 = K1<<1, else K2 = (K1<<1) $\oplus$ R

The Cryptographic Checksum (CC) consists of N bytes of the resulting MAC; here N is a system parameter in the range 4 to 8. Default value is 4. Via the system configuration one defines whether the upper or lower N bytes of the MAC shall be used for CC. Default is upper N bytes.

[7] 16 bytes for AES-128

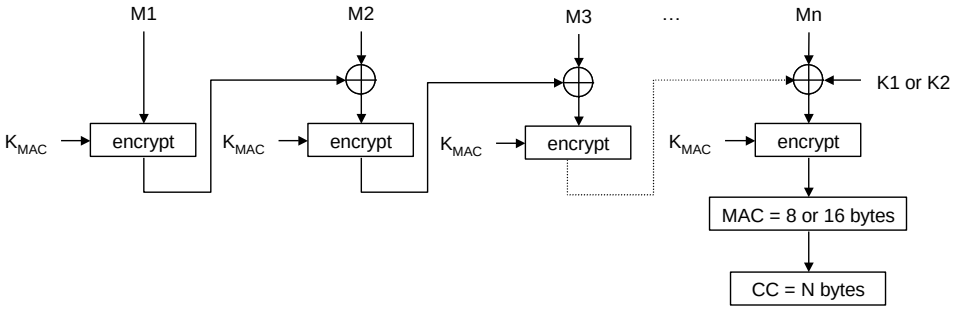


Figure 4

10.4 Encryption

Encryption uses CBC with an initial null vector (see figure).

The size of the data blocks Di depends on the algorithm (16 bytes for AES-128).

The action done by KEY depends on the key type.

Key type	Key value	Action
AES-128	K	AES encryption with K

Table 29

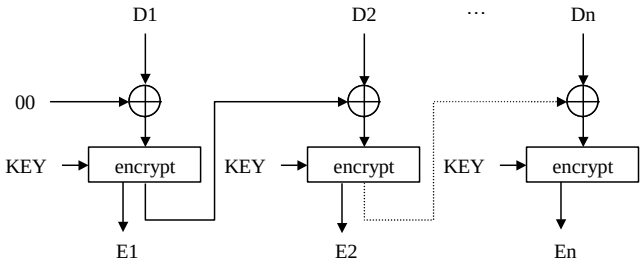


Figure 5

10.5 Decryption

Decryption is the reverse of encryption (see figure).

The action done by KEY depends on the key type.

Key type	Key value	Action
AES-128	K	AES decryption with K

Table 30

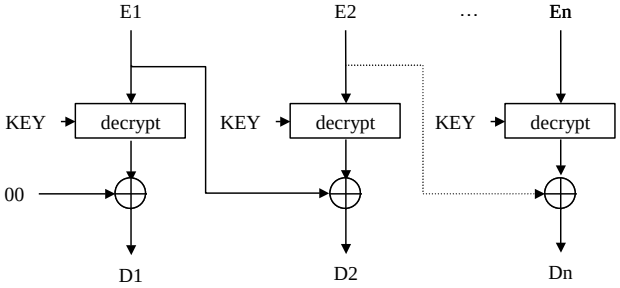


Figure 6

11 Secure Messaging

11.1 General

When the NF4 chip is used in an insecure environment, the exchange of APDUs with the PCD can be exposed to threats. These must be countered by means of counter-measures. The following threats and common counter-measures are identified:

Threat	Counter-measures
Substitution, removal or insertion of APDUs	Use an incremental Send Sequence Counter (SSC) or MAC chaining
Modification of APDU	Use Message Authentication Code (MAC)
Exposure of data content	Encrypt the data

Table 31

These counter measures are implemented in Secure Messaging (SM) using cryptographic mechanisms to protect the exchanges between PCD and NF4 chip. In particular, SM uses session keys and a Send Sequence Counter (SSC) or MAC chaining to protect APDUs with a MAC and possibly data encryption.

When SM is used, there is always MAC protection. Whether there is data encryption depends on the encryption settings (bits 4 and 5) defined in parameter P1 of the Mutual Authenticate command.

11.2 Initial Vector

The CMAC algorithm used in the computation of cryptographic checksum (see Cryptographic Checksum) makes use of an Initial Vector (IV). The IV has a size of 16 bytes (AES-128 crypto).

The initial value of IV is established when the secure session is established.

Two strategies can be used for subsequent IV: MAC chaining and Send Sequence Counter.

The actual mechanism used is defined via system configuration (see System Configuration).

11.2.1 MAC chaining

The first IV is its initial value. Subsequent values of IV consist of the CMAC computed to obtain the Cryptographic Checksum of the last sent or received APDU using SM.

Thus the cryptographic checksum of the first SM command is computed using the initial value. The response from the NF4 chip is computed using the newly computed CMAC, etc.

11.2.2 Send Sequence Counter

IV is a sequence counter, also referred to as Send Sequence Counter (SSC). SSC is incremented before each MAC computation[8]. Thus the first command uses SSC+1, the first response SSC+2, etc.

[8] SCC is only incremented when secure messaging is used.

11.3 Establishing Secure Messaging

The SM environment is established via the MUTUAL AUTHENTICATE command and consists of the session keys K_{ENC} and K_{MAC} and IV.

The SM environment is set up after the NF4 chip has sent the MUTUAL AUTHENTICATE response. If the MUTUAL AUTHENTICATE command itself makes use of SM, the new SM environment will become valid after the response is sent from NF4 chip, i.e. the response still uses the old SM environment.

11.4 BER-TLV Data Objects

SM modifies command and response APDU as described below. It thereby uses the concept of BER-TLV Data Object (DO).

As defined in reference **[12]** a BER-TLV DO consists of 3 consecutive fields:

- The tag field (T) consists of one or more consecutive bytes indicating a class type and a number.
- The length field (L) consists of one or more consecutive bytes indicating the length of the following field.
- The value field (V) indicates the value of the DO. If L=0 the value is not present.

11.4.1 Coding of Length Field

When bit 7 (most significant bit) of the most significant byte of the length field is set to 0, the length consists of only one byte and has the value given by bits 6 to 0. The field length is within the range 1 to 127.

When bit 7 (most significant bit) of the most significant byte of the length field is set to 1, bits 6 to 0 encode the number of subsequent bytes. The subsequent bytes code an integer representing the number of bytes in the value field. Thus two bytes are necessary to express up to 255 bytes in the value field[9].

11.4.2 Data Objects used in SM

The following TLV data objects are used:

Tag T	Length L	Value V	Description
81h	L	data	Plain data field L = length of data field (coded as one or two bytes)
87h	L	PI CG	Encrypted data field L = length of PI CG (coded as one or two bytes) Padding Indicator (PI) equals 01h when padding was used in the construction of CG, 02h otherwise Cryptogram (CG) = ENC(data || PB), where ENC is a CBC encryption using the SM key K_{ENC}
97h	1	Le	Le of unsecured command. When missing it implies a Le value of 256.
99h	2	SW1 SW2	Status code of unsecured response
8Eh	4 - 8	CC	Cryptographic Checksum (see p.26). NF4 chip uses a length that is configurable in the range 4 to 8. See System Configuration on p.34.

Table 32

11.5 Secure Messaging errors

When a data object is either missing or incorrect, the status codes 6987h or 6988h are respectively returned in clear and the Secure Messaging condition is invalidated. The Secure Messaging condition is not invalidated for any other status return.

[9] Thus a length of 255 shall be encoded as 81h FFh.

If Le is larger than possible maximum data size as described in the next section, a return code 6Cxx will be send in secure messaging. The command will then have to be sent with Le at most equals to its possible maximum data size as described.

11.6 Command format

By way of illustration it is assumed that default padding is used and that the Cryptographic Checksum has 4 bytes. The notation PB means 'padding bytes' (see Padding chapter). The maximum data lengths can vary depending on the padding strategy and the length of the Cryptographic Checksum.

Given that the unsecured command has the following form:

command header (CH)	command body
CLA INS P1 P2	[Lc] [PV] [Le]

Table 33

PV = plain value.

The corresponding SM command is as follows.

11.6.1 Case 1 and case 2 (no command data)

command header (CH)	command body							
CLA INS P1 P2	new Lc							new Le
		T_{LE}	L_{LE}	Le	T_{CC}	L_{CC}	CC	
CLA⊕0Ch INS P1 P2	length of new data	97h	1	Le	8Eh	4	CC	00

Table 34

The LE data object is only present when Le is present in the plain command (case 2).

The data to include in the MAC computation are:

- Le absent: CH
- Le present: CH || PB || T_{LE} || L_{LE} || Le

11.6.2 Case 3 and case 4 (command data)

11.6.2.1 MAC only

command header (CH)	command body										
CLA INS P1 P2	new Lc	new data									new Le
CLA⊕0Ch INS P1 P2	length of new data	T_{PV}	L_{PV}	PV	T_{LE}	L_{LE}	Le	T_{CC}	L_{CC}	CC	00
		81h	Lc	command data	97h	1	Le	8Eh	4	CC	

Table 35

The LE data object is only present when Le is present in the plain command (case 4).

The data to include in the MAC computation are:

- Le absent: CH || PB || T_{PV} || L_{PV} || PV
- Le present: CH || PB || T_{PV} || L_{PV} || PV || T_{LE} || L_{LE} || Le

The total command data length is (assuming L_{LE} takes one byte):

- Lc < 128: Lc+8 (respectively Lc+11 when Le is present)
- Lc ≥ 128: Lc+9 (respectively Lc+12 when Le is present)

The maximum value of Lc is thus 255-9=246 (respectively 243).

11.6.2.2 Encryption + MAC

command header (CH)	command body										
CLA INS P1 P2	new Lc	new data									new Le
CLA⊕0Ch INS P1 P2	length of new data	T_{PICG}	L_{PICG}	PI	CG	T_{LE}	L_{LE}	Le	T_{CC}	L_{CC}	CC	00
		87h	L_{PICG}	padding indicator	cryptogram	97h	1	Le	8Eh	4	CC	

Table 36

The LE data object is only present when Le is present in the plain command (case 4).

L_{PICG} = 1 + size of cryptogram

The data to include in the MAC computation are:

- Le absent: CH || PB || T_{PICG} || L_{PICG} || PI || CG
- Le present: CH || PB || T_{PICG} || L_{PICG} || PI || CG || T_{LE} || L_{LE} || Le

Given a cryptogram size N, the total command data length is:

- L_{PICG} < 128: N+9 (respectively N+12 when Le is present)
- L_{PICG} ≥ 128: N+10 (respectively N+13 when Le is present)

Hence the maximum size of N is 240, being the largest multiple of 8 or 16 smaller than 255-10=245 (respectively 255-13=242). The plain data (PV) are thus restricted to 240 bytes since there is no padding when the data are a multiple of 16 (AES-128).

11.7 Response format

As illustration it is assumed that default padding is used and that the Cryptographic Checksum has 4 bytes. The notation PB means 'padding bytes' (see Padding chapter).

Given that the unsecured response frame has the following form:

response body	status bytes
[Lr bytes]	SW1 SW2

The corresponding SM response is as follows.

11.7.1 Case 1 and case 3 (no response data)

response body						status bytes
new data						new SW1 SW2
T_{SW}	L_{SW}	SW	T_{CC}	L_{CC}	CC	
99h	2	SW1 SW2	8Eh	4	CC	

Table 37

The data to include in the MAC computation are:

T_{SW} || L_{SW} || SW

11.7.2 Case 2 and case 4 (response data)

11.7.2.1 MAC only

response body									status bytes
new data									new SW1 SW2
T_{PV}	L_{PV}	PV	T_{SW}	L_{SW}	SW	T_{CC}	L_{CC}	CC	
81h	Lr	response data	99h	2	SW1 SW2	8Eh	4	CC	

Table 38

The data to include in the MAC computation are:

T_{PV} || L_{PV} || PV || T_{SW} || L_{SW} || SW

The maximum value of L_{PV} is 256-13=243.

11.7.2.2 Encryption + MAC

response body										status bytes
new data										new SW1 SW2
T_{PICG}	L_{PICG}	PI	CG	T_{SW}	L_{SW}	SW	T_{CC}	L_{CC}	CC	
87h	L_{PICG}	padding indicator	cryptogram	99h	2	SW1 SW2	8Eh	4	CC	

Table 39

L_{PICG} = 1 + size of cryptogram

The data to include in the MAC computation are:

T_{PICG} || L_{PICG} || PI || CG || T_{SW} || L_{SW} || SW

Given a cryptogram size N, the total response data length (i.e. excluding new SW1 SW2) is:

- N < 127: N+13
- N ≥ 127: N+14

Hence the maximum size of N is 240, being the largest multiple of 8 or 16 smaller than 256-14=242. The plain data (PV) are thus restricted to 240 bytes since there is no padding when the data are a multiple of 16 (AES-128).

12 Anti-tearing and Transaction Protection

12.1 Concept

Updating information should be all or nothing. Yet there are many situations whereby an action is not always completed, such as when the user prematurely removes a tag from the NFC field, power failure, a faulty terminal, etc. In such cases the NF4 chip memory must be restored to the state before the action was started. To be able to do so, the NF4 chip memory status or Before Image (BIM) must be saved before the action is done. The BIM not only involves file data but in some cases control information.

The mechanism that implements the all or nothing action for a single command is referred to as anti-tearing support. In certain situations it may be required to update two or more locations in memory and these updates should be all or nothing. Hence anti-tearing support for each individual update is not sufficient. The mechanism that implements the all or nothing action across multiple commands is referred to as Transaction-Protected Session (TPS).

When the operation is not completed, an automatic rollback will be done, either during the current command execution or after reset (e.g. when chip was removed).

12.2 Transaction-Protected Session

In certain situations it may be required to update two or more files or separate locations within a same file and these updates should be all or nothing. Hence TP support for each individual update is not sufficient.

For such situations there is the notion of Transaction-Protected Session (TPS).

A TPS is initiated via the TPS START command. In a TPS any updates (write operations) will be transaction protected, irrespective of the location where the updates are done. Thus a file can be selected and updated, followed by another file selection and update.

A TPS is terminated via the TPS COMMIT command. If a TPS is interrupted before the TPS COMMIT command was completed, any updates will be reversed at the next usage and the card will return to the state before the TPS START command. Alternatively, a TPS can be reversed with a TPS ROLLBACK command.

TPS cannot be used with all commands. In particular, the following commands may not be used within a TPS and will terminate with status return 6985h:

- CHANGE PASSWORD
- CREATE FILE
- TPS START

Note that VERIFY PASSWORD is allowed within a TPS. However this command does not take part in the session, i.e. on rollback the try count will not be restored to its earlier value.

During a TPS, NF4 chip is set in reception mode on the interface that has received the TPS START command. If for example a contactless reader sends a command while a TPS session is active on the contact interface, it will be ignored. After TPS COMMIT or TPS ROLLBACK command, the NF4 chip will again be in reception mode on both interfaces.

12.3 Error handling in a TPS

In a normal situation when an error occurs during execution of a command, any intermediate data updates will have been saved and will be restored automatically, either during execution of the command or at the next card usage. Hence the card image will not have changed.

Inside a TPS the situation is different. If for some reason insufficient resources are available, an error will be returned. However any intermediate data updates that were saved will not be automatically restored. This could leave the card in a logically inconsistent state.

The only action that should be done is a TPS ROLLBACK command.

The memory image will also be restored after a reset.

Note: On reception of error 6F06h inside a TPS always execute a TPS ROLLBACK or do a chip reset.

13 System Configuration

Several system parameters relate to:

- Hardware configuration: logistic data, …
- NF4 chip configuration: version, release, …
- Communication: selection of communication modes and associated parameters

System parameters can be get and/or put via the GET DATA and PUT DATA commands respectively. Each parameter is given a 2-byte identifier corresponding to a class and subclass.

System parameters can be read without restriction.

Updating requires that the mutual authentication condition be established with the first key of the KEY file[10].

The table below indicates the parameters. Not all are necessarily supported in a given release of NF4 chip.

Class	Sub class	size of value	read only	Description	default value
41h				**Hardware related parameters**	
	00h	1	Y	hardware platform	03h
	01h	1	Y	chip model	10h
	02h	L	Y	logistic data L depends on platform	-
42h				**NF4 IC related parameters**	
	00h	1	Y	Version	1
	01h	1	Y	Release	0
	02h	1	Y	block size/8	32
	03h	2	Y	available memory in units of block size (MSB - LSB)	
	10h	2		FID of DF selected by default after reset specify FFFFh if none[11]	FFFFh
	11h	2		FID of EF selected by default after reset specify FFFFh if none[12]	FFFFh
	20h	1	[13]	Cryptographic Checksum: bit 7: rule for subsequent IV 0: use MAC chaining 1: use Send Sequence Counter bit 6: rule for extracting CC from CMAC 0: CC is left-most bytes of CMAC 1: CC is right-most bytes of CMAC bit 5: use of padding in computing CMAC 0: no padding when multiple of block length 1: always padding bits 3-4: RFU bits 0-2: CC size – 4 must be ≤ 4[14]	00h
	21h	1	[13]	SM encryption bit 7: use of padding in computing cryptogram 0: no padding when multiple of block length 1: always padding	00h
	40h	8		data space freely available	8x 00h

10 For enhanced security one can always use Secure Messaging.
11 MF is selected when none is specified or when the file does not exist.
12 No EF shall be selected when none is specified or when the file does not exist.
13 Any changes only apply as of the next Mutual Authenticate command.
14 Hence the CC size can be in the range 4 to 8.

Class	Sub class	size of value	read only	Description	default value
	50h	2		read or save the value of the tamper switch (RF version mode only) GET DATA command returns: byte 0: Current value of tamper switch (0 or 1) byte 1: Last saved value (0 or 1, initial value=1)) PUT DATA command saves the current value of the tamper switch. The input data are ignored.	0001h
43h				**Communication related parameters**	
	00h	1	Y	supported protocols bit mask bit value 0=not supported, 1= supported bit 5 RFU bit 4 RFU bit 1: RFU bit 0: PICC Type A	13h
	01h	1		configuration bit mask bit value 0=deactivated, 1=activated Only the bits as in supported protocols bit mask can be used. There must always be at least one protocol activated.	01h
	02h		Y	historical bytes in default ATS	'NF410'
	03h	L ≤ 15		historical bytes in user configured ATS L=0 activates the default ATS[15]	-
	A0h	1		UID configuration random UID when 0, else manufacturer UID	01h
	A1h	7	Y	manufacturer UID	UID defined on the card
	B3h	L ≤ 16		AID L=0 corresponds to a one byte zero value	-
44h				**NFC related parameters**	
	00h	1		Mapping version field in Capability Container[16] allowed values are 10h and 20h	10h
	01h	1		MLe field of Capability Container (MSB - LSB) 14 < MLe < 256	FFh
	02h	1		MLc field of Capability Container (MSB - LSB) 0 < MLc < 256	FFh
	10h	2	Y	NDEF file File Identifier (MSB - LSB)	E104h
	11h	2	Y	NDEF file maximum size (MSB - LSB)	4K/16K/32K
	12h	1		NDEF file write access condition 00h: write access granted without security FFh: no write access (read-only)	00h
	20h	2	Y	Proprietary file File Identifier (MSB - LSB)	E105h
	21h	2	Y	Proprietary file maximum size (MSB - LSB)	0400h
	22h	1		Proprietary file write access condition 00h: write access granted without security FFh: no write access (read-only)	00h

Table 40

[15] Hence user historical bytes must be at least 1 byte long.
[16] This changes also the AID of DF NFC to respect the specification

14 Commands

14.1 Command set

The following table gives an overview of all commands supported by the NF4 chip[17].

Command	CLA	INS
Change Password	00h	24h
Create File	00h	E0h
Directory Info	80h	DCh
Get Challenge	00h	84h
Get Data	00h	CAh
Get Random	80h	84h
Get Response	00h	C0h
MSE: Erase	00h	22h
MSE: Get Info	80h	22h
MSE: Store	00h	22h
Mutual Authenticate	00h	82h
Put Data	00h	DAh
Read Binary	00h	B0h
Select File	00h	A4h
TPS Commit	80h	8Eh
TPS Rollback	80h	8Ch
TPS Start	80h	8Ah
Update Binary	00h	D6h
Verify Password	00h	20h
Write Binary	00h	D0h

[17] Commands with CLA ≠ 00h are proprietary.

14.2 Change Password

CHANGE PASSWORD command assigns a new value to the Password. The Password must be verified first when it has a non-zero value.

14.2.1 Command APDU

CLA	00h
INS	24h
P1 P2	00h 00h
Lc	Length of data
Data	New PASSWORD value: 4-8 bytes
Le	Empty

14.2.2 Response APDU

Data	Empty
SW1 SW2	Status bytes

14.3 Create File

CREATE FILE command initiates the creation of an Elementary file (EF) under MF.

Upon successful completion of the command, the created file becomes the current file.

The FID of the file to be created must be specified. The FID must be unique as discussed in File Selection.

For an EF, by default the file will be given a Short File Identifier (SFI) corresponding to the lower 5 bits of the FID. If another SFI value is required, include tag 88h specifying the required value. If you do not want the EF to have an SFI, include tag 88h with zero length.

Note: CREATE FILE command is only allowed under MF.

14.3.1 Command APDU

CLA	00h
INS	E0h
P1 P2	00h 00h
Lc	Length of data
Data	FCP template
Le	Empty

The FCP template is a composite Data Object of structure TLV with T=62h, containing the following data objects[18]:

Tag	Mandatory	L	Meaning
80h	Y	2	Size: Number of data bytes in file
83h	Y	2	FID (MSB LSB)
86h	Y	3	3 x Access Condition byte (1 bytes per AC - see Access Conditions)
88h		1	SFI coded in bits b7-b3
C0h		1	EF initial value[19] 00h file body is filled with 00h 03h file body is filled with FFh 1xxxxxxxx file body is filled with (7-bit) ASCII 0xxxxxxx

Table 41

14.3.2 Response APDU

Data	Empty
SW1 SW2	Status bytes

[18] Other tags will be ignored.
[19] If not specified, 00h shall be used.

14.4 Directory Info

DIRECTORY INFO command is a proprietary command that returns information about a file and its children.

It allows constructing the complete file system arborescence.

14.4.1 Command APDU

CLA	80h	
INS	DCh	
P1 P2	Selection	
	00h 00h	currently selected file
	other	n-th child with n = P1 P2 (MSB LSB)
Lc	Empty	
Le	Length of FCI or 0 to indicate 256 bytes	

14.4.2 Response APDU

Data	FCI template
SW1 SW2	Status bytes

The FCI template is a composite Data Object of structure TLV with T=6Fh, containing the following data objects:

Tag	L	Value	Applies to
80h	2	Number of data bytes in file excluding structural information.	EF
81h	2 or 3[20]	Number of data bytes in file including structural information. In case of DF this includes the total size taken up by the DF including all its contained files.	any
82h	1	File descriptor: 38h (MF), 39h (DF)	MF, DF
	2	File descriptor byte followed by data encoding byte (=01h 00h)	transparent EF
83h	2	FID	any
84h	5 to 16	DF name	DF
85h	2	FID of parent (FFh FFh in case of MF)	any
86h	3	3 x Access Condition byte (1 byte per AC)	any
88h	1	SFI coded in bits b7-b3	EF
A5h		Proprietary Information	MF, DF

The Proprietary Information is a composite Data Object of structure TLV with T=A5h, containing the following data objects:

Tag	Mandatory	L	Meaning	Comment
80h	Y	2	number of children	

[20] 2 for EF, 3 for DF

14.5 Get Challenge / Get Random

GET CHALLENGE command initiates the issuing of a challenge by the NF4 chip for use in a security related procedure. The NF4 chip generates and returns an 8-byte random number. The chip retains this number for use in a subsequent MUTUAL AUTHENTICATE command.

GET RANDOM command is similar, except that the generated random number is not retained by the card and can be used by an interface device that does not have good random number generation capabilities.

14.5.1 Command APDU

CLA	00h (GET CHALLENGE), 80h (GET RANDOM)
INS	84h
P1 P2	00h 00h
Lc	Empty
Data	Empty
Le	8

14.5.2 Response APDU

Data	8-byte challenge
SW1 SW2	Status bytes

14.6 Get Data

GET DATA command returns the value of system parameter given by P1 P2.

For details, see System Configuration.

14.6.1 Command APDU

CLA	00h
INS	CAh
P1	Class
P2	Subclass
Lc	Empty
Le	The length depends on parameter.

14.6.2 Response APDU

Data	System parameter (format LV)
SW1 SW2	Status bytes

14.7 Get Response

GET RESPONSE command is used to retrieve response data for case 4 commands (see APDU message structure). It returns all or part of a pending APDU to the interface device.

GET RESPONSE command is not an application level command, but a Transport protocol only relevant to the T=0 protocol.

This command retrieves any pending data in the card's output buffer. By the very nature of the T=0 protocol, parameter P3 must be equal to or less than the number of pending data bytes. If P3 is less, exactly P3 bytes will be returned and the return code (61XXh) indicates that XX data bytes are still pending. If P3 is greater than the number of pending data, the return code (61XXh) indicates the number of data bytes still pending.

NOTE: The command itself cannot use Secure Messaging, since the response would overwrite the pending data. If Secure Messaging is used in the command, a non-Secure Messaging error code is returned.

14.7.1 Command APDU

CLA	00h
INS	C0h
P1 P2	00h 00h
P3	Number of bytes to retrieve

14.7.2 Response APDU

Data	P3 pending response APDU bytes
SW1 SW2	Status bytes

14.8 MSE: Erase

MSE: ERASE command is used to erase the second key in KEY file.

To execute this command the Access Conditions for MSE: ERASE must be fulfilled.

Note: It is not possible to erase the first key of KEY file.

14.8.1 Command APDU

CLA	00h
INS	22h
P1	F4h
P2	01h
Lc	Empty
Data	Empty
Le	Empty

14.8.2 Response APDU

Data	Not present
SW1 SW2	Status bytes

14.9 MSE: Get Info

MSE: GET INFO command is used to obtain information related to a particular key.

No Access Conditions are required to execute this command.

14.9.1 Command APDU

CLA	80h
INS	22h
P1	00h
P2	Reference of key: bits 7-1: 0 bit 0: key in KEY file (0=KEY 1, 1=KEY 2)
Lc	Empty
Data	Empty
Le	Response data length or 0 to indicate 256 bytes

14.9.2 Response APDU

Data	Key Template
SW1 SW2	Status bytes

The Key Template is a composite Data Object of structure TLV with T=B8h, containing the following data objects (see also KEY File Management):

Tag	Mandatory	L	Meaning	Comment
80h	Y	1	Key type	
84h	Y	1	Key version	
83h	Y	L	Key Verification Value (see p.26)	L depends on key type

Table 42

14.10 MSE: Store

MSE: STORE command is used to store a specific key in a KEY file. Input is a key template.

To execute this command the Access Conditions for MSE: STORE must be fulfilled.

14.10.1 Command message

CLA 00h

INS 22h

P1 F2h

P2 Reference of key record:

bits 7-1: 0
bit 0: key in KEY file (0=KEY 1, 1=KEY 2)

Lc Length of data

Data Key Template

Le Empty

The Key Template is a composite Data Object of structure TLV with T=B8h, containing the following data objects (see also KEY File Management):

Tag	Mandatory	L	Meaning	Comment
80h	Y	1	Key type	
84h		1	Key version	default value is zero
82h	Y	L	Key value	length must match algorithm

Table 43

14.10.2 Response message

Data Not present

SW1 SW2 Status bytes

14.11 Mutual Authenticate

MUTUAL AUTHENTICATE command establishes a secure session using a common key to mutually authenticate PCD and NF4 chip. The result is a pair of session keys K_{ENC} and K_{MAC} as well as an initial Send Sequence Counter (SSC). P2 also defines the security requirements of subsequent commands making use of SM.

The mechanism is based on ISO11770-2 (ref [11]). It uses keys as common secret, which for NF4 chip can be any key from the MF KEY file and is selected via P2.

MUTUAL AUTHENTICATE requires that a GET CHALLENGE command be issued by the PCD just before.

The steps to establish a secure session are as follows:

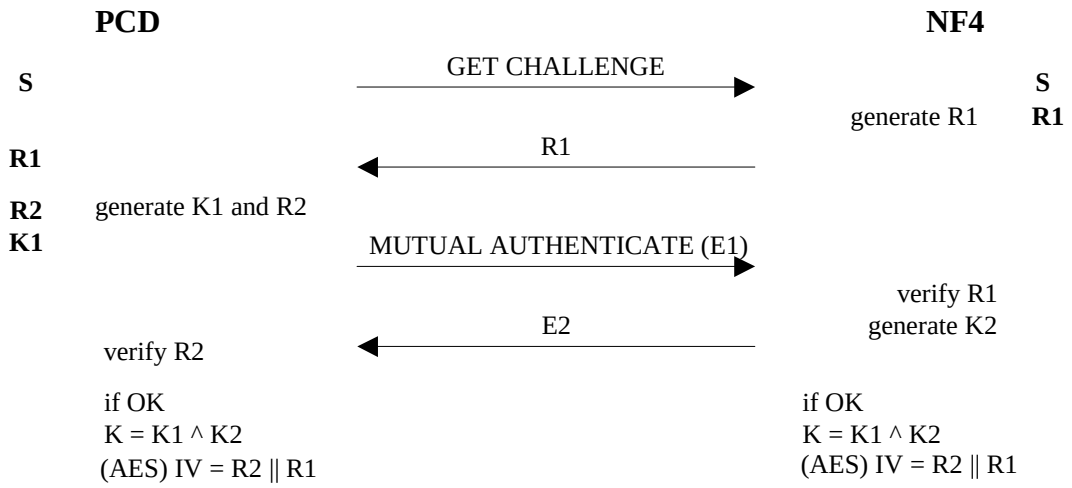


Figure 7

- PCD and NF4 chip share a key S.
- PCD obtains a challenge R1 (8 bytes) via a GET CHALLENGE command. Both NF4 and PCD retain R1.
- PCD generates a random number R2 (8 bytes) and a random number K1 (16 bytes) and constructs the encrypted value E1 = ENC (R2 || R1 || K1) using key S. This value is sent to NF4 in a MUTUAL AUTHENTICATE command, specifying the reference to key S in P2.
- NF4 chip decrypts the command data using the key specified in P2 and verifies R1. On error it returns an error code; otherwise it extracts R2 and K1
- NF4 chip generates a random number K2 (16 bytes) and constructs the encrypted value E2 = ENC(R1 || R2 || K2) using key S. E2 is returned to PCD
- PCD decrypts and verifies R2 and (when OK) extracts K2.

The resulting 16-byte AES-128 session key K shared between PCD and NF4 chip is computed as[21]:

$$K = K2 \oplus K1$$

K is used both for encryption (K_{ENC}) and for MAC computation (K_{MAC}).

The initial value of the Initial Vector is:

- AES: IV (16 bytes) = R2 || R1

[21] The random values K1 and K2 are not the same as the sub-keys K1 and K2 in the MAC computation.

14.11.1 Command message

CLA	00h
INS	82h
P1	Type of session key: 1: AES-128
P2	Reference of common key and SM session requirements: bits 7-6: 0 bit 5: Data encryption in command is required bit 4: Data encryption is enabled in response bits 3-1: 0 bit 0: key in KEY file (0=KEY 1, 1=KEY 2)
Lc	Length of data
Data	E1
Le	Length of response data

14.11.2 Response message

Data	E2
SW1 SW2	Status bytes

14.12 Put Data

PUT DATA command sets the value of system parameter given by P1 P2.

For details, see System Configuration.

14.12.1 Command message

CLA	00h
INS	DAh
P1	Class
P2	Subclass
Lc	Length of data
Data	Parameter value
Le	Empty

14.12.2 Response message

SW1 SW2	Status bytes

14.13 Read Binary

READ BINARY command returns (all or part of) the content of an EF with transparent structure.

The file is read starting from an offset defined by P1-P2.

When the number of bytes to be read goes beyond the file size, the command returns the number of bytes up to the end of file.

When the starting offset is equal to or larger than the file size, the command returns error 6B00h.

When the command contains a Short File Identifier (SFI) it implicitly selects the file as the Current EF.

14.13.1 Command message

CLA	00h	
INS	B0h	
P1 P2	bit7 of P1=0:	(P1,P2) = offset in file to start reading
	bit7 of P1=1:	bits 4-0 of P1 define SFI; P2 = offset in file to start reading
Lc	Empty	
Data	Empty	
Le	Number of bytes to read	

14.13.2 Response message

Data	Date read up to a maximum of Le bytes or 256 if Le=0
SW1 SW2	Status bytes

14.14 Select File

SELECT FILE command sets a current file. Subsequent commands may implicitly apply to this current file.

Selecting a DF sets it as Current DF. After such selection, an EF may be implicitly selected by its SFI.

Selecting an EF sets a pair of current files: the EF and its parent DF.

After the answer to reset, the MF is implicitly selected unless another DF/EF is selected in the system configuration.

14.14.1 Command message

CLA	00h
INS	A4h
P1	Selection control
	supported values are:
	0 select MF, DF or EF by FID (data field = FID) 1 select child DF (data field = FID) 2 select EF under Current DF (data field = FID) 3 select parent DF of Current DF (data field empty) 4 select DF by name
P2	bits 1-0: 00: first or only occurrence 10: next occurrence (P1=4)
	bits 3-2: 00: return FCI 11 no return data
	other bits must be zero
Lc	Length of data
	Possible values are:
	empty select parent DF (if P1=3) 2 FID (if P1=0 ➔2) n name (if P1=4), $5 \leq n \leq 16$
Data	See Lc
Le	value returned depends on bits 3-2 of P2
	Ch Empty 0h Length of FCI or 0 to indicate 256 bytes

14.14.2 Response message

Data	FCI template
SW1 SW2	Status bytes

The FCI template is a composite Data Object of structure TLV with T=6Fh, containing the following data objects:

Tag	L	Value	Applies to
80h	2	Number of data bytes in file excluding structural information[22]	EF
82h	1	File type	MF, DF
	2	File type followed by data encoding byte (=0)	transparent EF
83h	2	FID	all
84h	5 to 16	DF name	DF
88h	1	SFI coded in bits b7-b3	EF

Table 44

14.15 TPS Commit

TPS COMMIT command ends a transaction-protected session. Any updates done since the start of the TPS will become permanent.

14.15.1 Command message

CLA 80h

INS 8Eh

P1 P2 00h 00h

Lc Empty

Le Empty

14.15.2 Response message

Data Empty

SW1 SW2 Status bytes

[22] This equals the number of useful storage bytes. For transparent files it equals the size of the file.

14.16 TPS Rollback

TPS ROLLBACK command terminates a transaction-protected session by undoing any updates done since the start of the TPS.

14.16.1 Command message

CLA	80h
INS	8Ch
P1 P2	00h 00h
Lc	Empty
Le	Empty

14.16.2 Response message

Data	Empty
SW1 SW2	Status bytes

14.17 TPS Start

TPS START command initiates a transaction-protected session.

14.17.1 Command message

CLA	80h
INS	8Ah
P1–P2	00h 00h
Lc	Empty
Le	Empty

14.17.2 Response message

Data	Empty
SW1 SW2	Status bytes

14.18 Update Binary

The UPDATE BINARY command initiates the update of bits already present in a transparent EF with the bits given in the command APDU.

The file is updated starting from an offset defined by P1-P2.

When the command contains a Short File Identifier (SFI) it implicitly selects the file and sets it as Current EF when successful.

14.18.1 Command message

CLA	00h	
INS	D6h	
P1 P2	bit7 of P1=0: bit7 of P1=1:	(P1,P2) = offset in file to start updating bits4-0 of P1 denote SFI; P2=offset in file to start updating
Lc	Length of data field	
Data		
Le	Empty	

14.18.2 Response message

Data	Empty
SW1 SW2	Status bytes

Note:
After sending an Update Binary command to the NF4 IC, the time to receive the 0x9000 answer depends on the number of bytes to be written in the memory of the NF4 chip.

The NF4 IC sends a 0x60 byte according to T=0 protocol to indicate to the external microcontroller (Host terminal) to wait as the

NF4 chip is processing a long command (like the creation of a large file). After each period of 9600 etu max, a 0x60 byte is sent by the NF4 IC to the host terminal.

14.19 Verify Password

VERIFY PASSWORD command initiates the comparison in the NF4 chip of the verification data with the reference data (Password value) stored in the NF4 chip.

See also User authentication.

14.19.1 Command message

CLA	00h
INS	20h
P1 P2	00h 00h
Lc	Length of data
Data	Password value
Le	Empty

14.19.2 Response message

Data	Empty
SW1 SW2	Status bytes

14.20 Write Binary

WRITE BINARY command initiates the logical OR of the data bytes already present in a transparent EF with the data bytes given in the command APDU.

The file is updated starting from an offset defined by P1-P2.

When the command contains a Short File Identifier (SFI) it implicitly selects the file as the Current EF.

14.20.1 Command message

CLA	00h
INS	D0h
P1 P2	bit 7 of P1=0: (P1,P2) = offset in file to start updating bit 7 of P1=1: bits 4-0 of P1 denote SFI; P2=offset in file to start updating
Lc	Length of data field
Data	
Le	Empty

14.20.2 Response message

Data	Empty
SW1 SW2	Status bytes

15 Return Codes

15.1 List of Return Codes

90 00	Success
61 XX	success, XX = number of response bytes (to be retrieved using GET RESPONSE)
63 00	security error
67 00	wrong length
68 00	function in CLA not supported
69 81	command not compatible with file structure
69 82	security status not satisfied
69 85	conditions of use not satisfied (command is not allowed in current context)
69 86	command not allowed
69 88	secure messaging data object incorrect
6A 80	incorrect parameters in data field
6A 81	function not supported
6A 82	file not found
6A 84	insufficient memory space
6A 86	incorrect parameter(s) P1-P2 (applies only to GET RESPONSE command)
6A 88	referenced data not found (e.g. KEY record)
6A 89	file already exists
6B 00	wrong parameter(s) P1-P2
6C XX	wrong length, XX indicates the exact length (Le)
6D 00	instruction code not supported or invalid
6E 00	class not supported
6F 00	non-fatal internal error
6F 01	T0 protocol - no response data available
6F 03	unknown key type
6F 06	insufficient TP memory
6F 07	maximum TPS memory allocated
6F 10	error updating a Transaction Registration Register
6F 11	error updating a Transaction Data Register
6F 12	error updating the Non-Volatile Memory
6F 13	error restoring a Before Image
6F FE	fatal internal error
90 FF	card not factorized

Table 45

15.2 Error Codes per Command

The table indicates the most relevant specific errors per command. For other errors, see List of Return Codes.

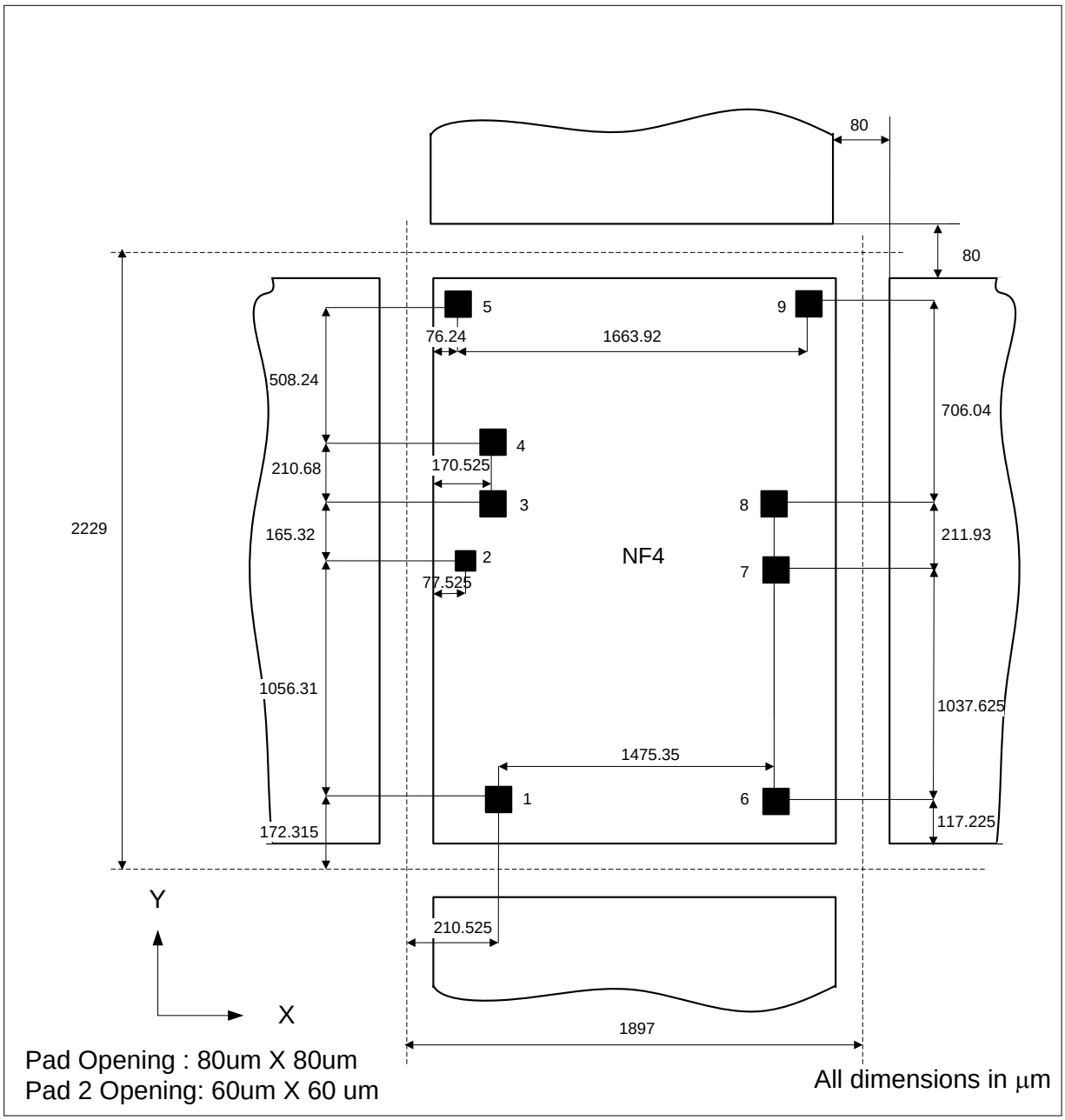
Return Code	Change Password	Create File	Directory Info	Get Challenge	Get Data	Get Random	MSE: Erase	MSE: Get Info	MSE: Store	Mutual Authenticate	Put Data	Read Binary	Select File
63 00										√			
69 81												√	
69 82		√					√		√		√	√	
69 85	√	√								√			
69 86												√	
6A 80		√							√		√		√
6A 81					√						√		
6A 82	√		√				√	√	√	√		√	√
6A 84		√											
6A 88					√			√			√		
6A 89		√											

Return Code	TPS Commit	TPS Rollback	TPS Start	Update Binary	Verify Password	Write Binary
63 00						
69 81				√		√
69 82				√		√
69 85	√	√			√	
69 86				√		√
6A 80						
6A 81						
6A 82				√	√	√
6A 84			√			
6A 88						
6A 89						

Table 46

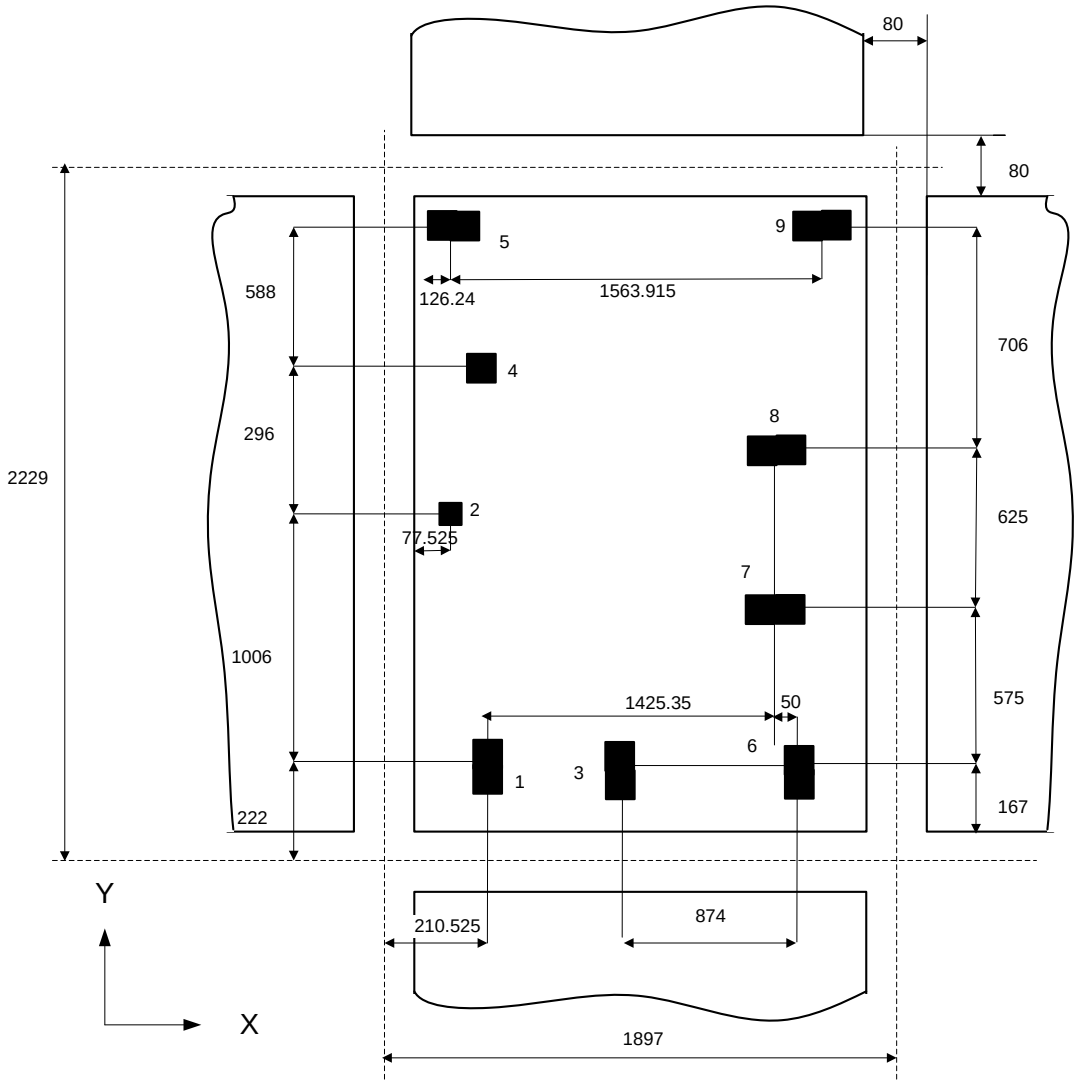
16 NF4 Chip Floor Plan

16.1 NF4 Die Form (Standard aluminum pads)



Pad #	Pad Name	Type	Direction	Description
1	IO	digital	IN/OUT	Serial data input/output – contact interface
2				Not bonded
3	GND	power		Common ground pin
4	VPOS	power	OUT	External power from RF
5	C1	analog/digital	IN	RF coil 1 signal (antenna terminal)
6	CLK	digital	IN	Serial clock signal – contact interface
7	RST/ Tamper	digital	IN	Reset input (must be connected to VCC for NF4 Dual IC version)
8	VCC	power		Serial contact interface power supply
9	C2	analog/digital	IN	RF coil 2 signal (antenna terminal)

16.2 NF4 Die Form (Bumped version)



Bump size: 100 um X 200 um
Bumps 2 & 4 size: 100 um X 100 um

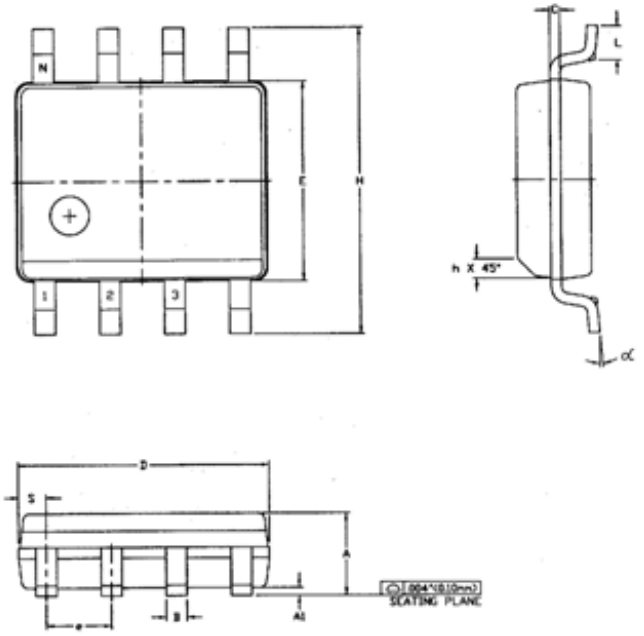
All dimensions in µm

Note: Bumps are Cu/Ni/Au with a height of 23.5+/-5um. Composition of the bumps is: CuNiAu (20.5 µm Cu, 1 µm Ni, 2 µm Au).

Pad #	Pad Name	Type	Direction	Description
1	IO	digital	IN/OUT	Serial data input/output – contact interface
2				Not bonded
3	GND	power		Common ground pin
4	VPOS	power	OUT	External power from RF
5	C1	analog/digital	IN	RF coil 1 signal (antenna terminal)
6	CLK	digital	IN	Serial clock signal – contact interface
7	RST/ Tamper	digital	IN	Reset input (must be connected to VCC for NF4 Dual IC version)
8	VCC	power		Serial contact interface power supply
9	C2	analog/digital	IN	RF coil 2 signal (antenna terminal)

16.3 NF4 SO8 Packaging

Package information



SYMBOL	DIMENSION IN INCH		DIMENSION IN MM	
	MIN.	MAX.	MIN.	MAX.
A	.053	.069	1.35	1.75
A1	.004	.010	0.10	0.25
B	.013	.020	0.33	0.51
C	.008	.010	0.19	0.25
D	.189	.1968	4.80	5.00
E	.150	.157	3.80	4.00
e	.050 BSC		1.27 BSC	
H	.228	.244	5.80	6.20
h	.010	.020	0.25	0.50
L	.016	.050	0.40	1.27
S	.0155	.0255	0.394	0.648
α	0°	8°	0°	8°

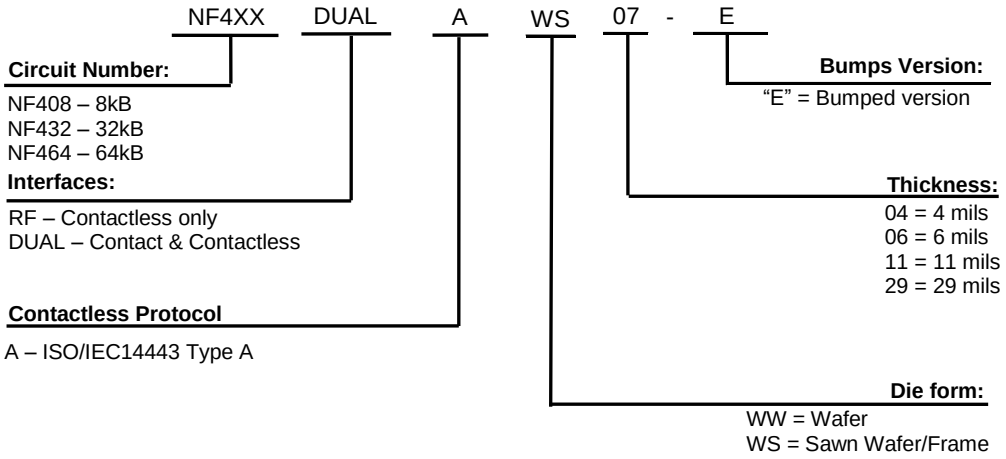
Figure 8

Pin description

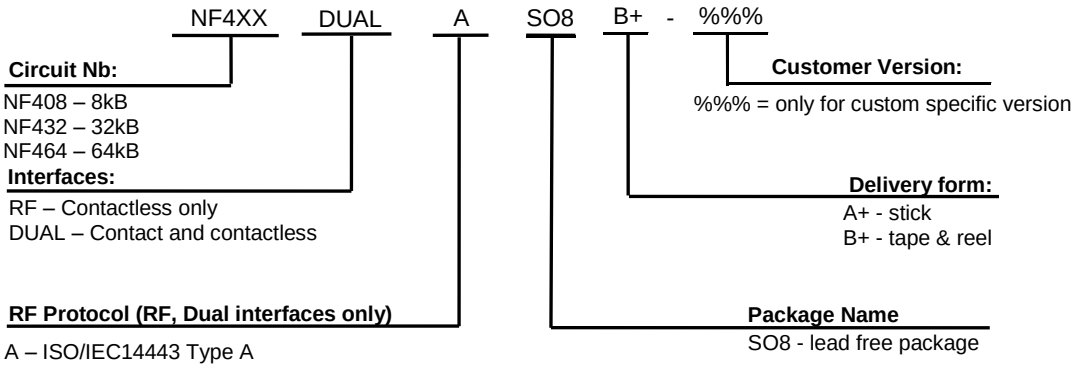
Pad #	Pad Name	Type	Direction	Description
1	C1	analog/digital	IN	RF Coil 1 antenna input
2	VPOS / Field detector	power	OUT	Field detector - DC voltage - external power from RF
3	VSS	power		Common analogue ground
4	IO / RF BUSY	digital	IN/OUT	Serial contact interface data input output / RF Busy line
5	CLK	digital	IN	Serial contact interface clock signal
6	RST / Tamper	digital	IN	Serial contact interface reset signal (should be always connected to VCC)
7	VCC	power		Serial contact interface power supply
8	C2	analog/digital	IN	RF Coil 2 antenna input

17 Ordering Information

17.1 Die form



17.2 Package form



17.3 Standard versions

The versions below are considered standard and should be readily available. For the other delivery form, please contact EM Microelectronic-Marin S.A. Please make sure to give the complete part number when ordering.

Part Number	Memory size	Die Form/Package
NF408RFAWS4E	8KB	8kB, RF only, Type A, Wafer Sawn 4 mils gold bumps
NF432RFAWS4E	32KB	32kB, RF only, Type A, Wafer Sawn 4 mils gold bumps
NF464RFAWS4E	64KB	64kB, RF only, Type A, Wafer Sawn 4 mils gold bumps
NF408RFASO8A+	8KB	8kB, RF only, Type A, SO8, stick
NF432RFASO8A+	32KB	32kB, RF only, Type A, SO8, stick
NF464RFASO8A+	64KB	64kB, RF only, Type A, SO8, stick

Table 47

18 Glossary

AES	Advanced Encryption Standard
APDU	Application Protocol Data Unit
ATR	Answer To Reset
CBC	Cipher Block Chaining
CC	Capability Container
DF	Dedicated File
DO	Data Object
EF	Elementary File
etu	elementary time unit
FID	File Identifier
IV	Initial Vector
KVV	Key Verification Value
MAC	Message Authentication Code
MF	Master File
PCD	Proximity Coupling Device – RFID reader
PICC	Proximity Card – RFID Tag
PIN	Personal Identification Number
RFU	Reserved for Future Use
SFI	Short File Identifier
SM	Secure Messaging
SSC	Send Sequence Counter
T=0	T=0 Protocol
TLV	Type Length Value
TP	Transaction Protected
TPS	Transaction-Protected Session
UID	Unique Identifier

19 NF4 Use cases

19.1 Read and Write inside the NDEF file

Open a Card session

SELECT DF NFC
APDU: 00A4040006D27600008501

SELECT EF E104 (NDEF file)
APDU: 00A4000002E104

READ BINARY
APDU: 00B0000000

UPDATE BINARY (Write NDEF)
APDU:
00D60000350033D1022E537091010F5402656E4E4634204E46432044656D6F5101175501656D6D6963726F656C656374726F6E69632E636F6D2F

Close the Card session

19.2 Recovery of Capability Container 8KB

Open a Card session

SELECT DF NFC
APDU: 00A4040006D27600008501

SELECT CAPACITY CONTAINER E103
APDU: 00A4000002E103

WRITE BINARY
APDU: 00D600001900171000FF00FF0406E104100000000506E10504000000

Close the Card session

19.3 Recovery of Capability Container 32KB

Open a Card session

SELECT DF NFC
APDU_1 00A4040006D27600008501

SELECT CAPACITY CONTAINER E103
APDU_1 00A4000002E103

WRITE BINARY
APDU_1 00D600001900171000FF00FF0406E104400000000506E10504000000

Close the Card session

19.4 Recovery of Capability Container 64KB

Open a Card session

SELECT DF NFC
APDU: 00A4040006D27600008501

SELECT CAPACITY CONTAINER E103
APDU: 00A4000002E103

WRITE BINARY
APDU: 00D600001900171000FF00FF0406E104800000000506E10504000000

Close the Card session

19.5 Create File with SFI

Open a Card session

Create File with FID = 1001, size = 256 bytes and SFI = 1
APDU: 00E000001562138002010083021001860300000880108C00100

Close the Card session

19.6 Read file with Select

Open a Card session

Select EF 1001
APDU: 00A40000021001

Read Binary 256 bytes with SFI=1 at offset 5
APDU: 00B0000500

Close the Card session

19.7 Read file with SFI

Open a Card session

Read Binary 256 bytes with SFI=1 at offset 5
APDU: 00B0810500

Close the Card session

19.8 Update File with Select

Open a Card session

SELECT EF 1001
APDU: 00A40000021001

UPDATE BINARY 5 bytes on a File with FID = 1001
APDU: 00D60000051122334455

READ BINARY
APDU: 00B0000000

Close the Card session

19.9 Update File with SFI

Open a Card session

UPDATE BINARY 5 bytes on a File with SFI = 1
APDU: 00D68100051122334455

READ BINARY
APDU: 00B0000000

Close the Card session

19.10 Write File with Select

Open a Card session

SELECT EF 1001
APDU: 00A40000021001

WRITE BINARY 5 bytes on a File with FID = 1001
APDU: 00D00000051122334455

READ BINARY
APDU: 00B0000000

Close the Card session

19.11 Write File with SFI

Open a Card session

WRITE BINARY 5 bytes on a File with SFI = 1
APDU: 00D08100051122334455

READ BINARY
APDU: 00B0000000

Close the Card session

19.12 Tag Create File with SFI

Open a Card session

Create File with FID = 1001, size = 256 bytes and SFI = 1
APDU: 00E0000015621380020100830210018603000000880108C00100

Close the Card session

19.13 Tag Read file with Select

Open a Card session

Select EF 1001
APDU: 00A40000021001

Read Binary 256 bytes with SFI=1 at offset 5
APDU: 00B0000500

Close the Card session

19.14 Tag Read file with SFI

Open a Card session

Read Binary 256 bytes with SFI=1 at offset 5
APDU: 00B0810500

Close the Card session

References

[1] ISO/IEC 7816-3 — Identification cards - Integrated circuit cards – Part 3: Electronic signals and transmission protocols

[2] ISO/IEC 7816-4 — Identification cards - Integrated circuit cards – Part 4: Organization, security and commands for interchange

[3] ISO/IEC 7816-5 —Identification cards - Integrated circuit cards – Part 5: Numbering system and registration procedure for application identifiers

[4] ISO/IEC 7816-8 —Identification cards - Integrated circuit cards – Part 8: Security related interindustry commands

[5] ISO/IEC 7816-9 — Identification cards - Integrated circuit cards – Part 9: Additional interindustry commands and security attributes

[6] ISO/IEC 14443-3 — Identification cards - Contactless integrated circuit(s) cards - Proximity cards, Part3: Initialization and anticollision

[7] ISO/IEC 14443-4 — Identification cards - Contactless integrated circuit(s) cards - Proximity cards, Part4: Transmission protocol

[8] FIPS PUB 197 — ADVANCED ENCRYPTION STANDARD (AES)

[9] NIST 800-38B — CMAC Mode for Authentication

[10] ISO 9798-2 —Entity authentication – using symmetric encipherment

[11] ISO 11770-2 — Key management – symmetric techniques

[12] ISO/IEC 8825 — Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)

[13] NFC Forum Type 4 Tag Version 1.0 - NFCForum-TS-Type-4-Tag_1.0 - 13 March 2007

[14] NFC Forum Type 4 Tag Version 2.0 - NFCForum-Type-4-Operation-Specification_2.0 - 28 March 2011