CIPURSE™Security Controller V1.2.0

Datasheet

CIPURSE™-based dedicated security controller for cost-optimized tickets, cards and wearables in transport ticketing, physical access, micro-payment, and multi-applications

Key features

- **Open Standard, CIPURSE™T Profile** compliant
  - Up to **8 CIPURSE™ applications** configurable
  - Up to **8 128-bit AES keys** may be assigned to the CIPURSE™ ADF
  - **4 PxSE ADF** configurable
  - **Secured communication** using AES-128 and session key derivation
  - **Mutual authentication** using AES-128
- **8 KB user memory** for application data storage
- Support of **legacy systems**: Optional **1 KB and 4 KB block oriented memory with NRG™ operation**
- **ISO/IEC 14443 Type A contactless interface**
- **Chip capacitance values of 27/56/78 pF** supporting various antenna form factors
- **CC EAL 5+ (high), CIPURSE™ certified**

Potential applications

Optimized for **secure multi-application smart city and mobility cards**

About this document

Scope and purpose

This document describes the features, functionality, and operational characteristics of SLS 32TLC100(M).

Intended audience

This document is primarily intended for system and application designers.

Note: *For more details, CIPURSE™Security Controller V1.2.0 Extended Datasheet available under NDA can be requested from Infineon Technologies.*

Table of contents

Table of contents

List of tables

List of figures

1 Introduction

CIPURSE™Security Controller is a dedicated security controller for cost-optimized tickets, cards, and wearables in transport ticketing, physical access, micro-payment, and multi-applications featuring CIPURSE™ functionality and optional block oriented memory with NRG™ operation. It is therefore the ideal migration product to migrate existing NRG™ systems towards more advanced and state of the art CIPURSE™ security based on AES-128.

1.1 System overview

CIPURSE™Security Controller is designed to operate both in a CIPURSE™ and in an NRG™ system. The product, in the following also denoted as proximity integrated circuit card (PICC), is connected to a terminal, in the following also denoted as proximity coupling device (PCD), via contactless interface providing both energy for operation and data exchange. The terminal is application specific and may be either connected to a host system (online terminal) or work standalone (offline terminal).

After anticollision and selection as per ISO/IEC 14443-3 [9]. The PCD may proceed as follows:

- Enter the NRG™ operation state by performing the authentication procedure to any of the sectors by sending the command AUTHENTICATE

  or

- Enter ISO/IEC 14443-4 [10] transmission protocol processing (T=CL) by sending a request for answer to select (RATS) command

See Chapter 4.1 for details on further steps to operate in CIPURSE™ or NRG™ mode.

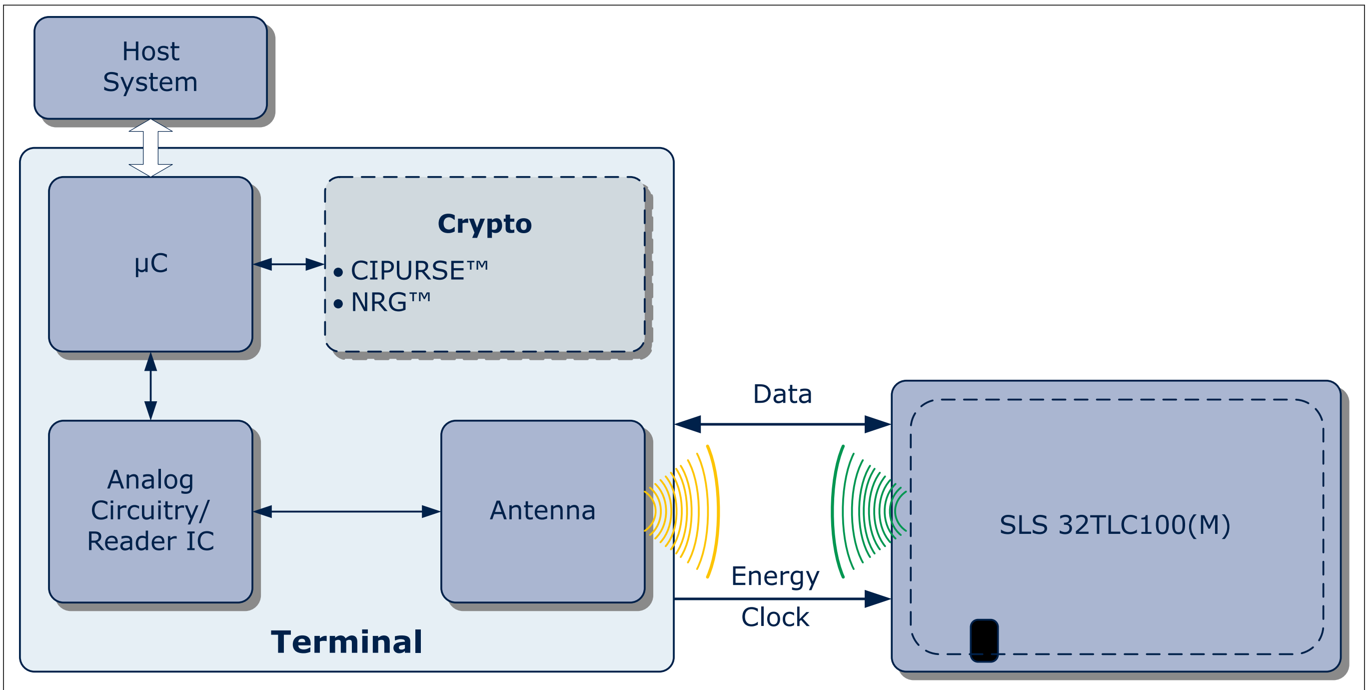

Figure 1 System overview

1.2 Product overview

CIPURSE™Security Controller is a cost-efficient implementation and designed for use in automatic fare collection systems, micro-payment, as access control token, and other smart card security applications. As a migration product, it also offers 1 KB and 4 KB block oriented memory with NRG™ operation. It is operated using the ISO/IEC 14443 Type A contactless interface.

The product allows handling a typical ticketing transaction in less than 100 ms. It is also suited for use in multi-application schemes, for example combining a transportation fare collection scheme and a ticketing system such as stadium ticketing. Further, the product offers robust contactless transmission which means that the card with CIPURSE™Security Controller may also remain in the wallet of the user even if there are coins in it.

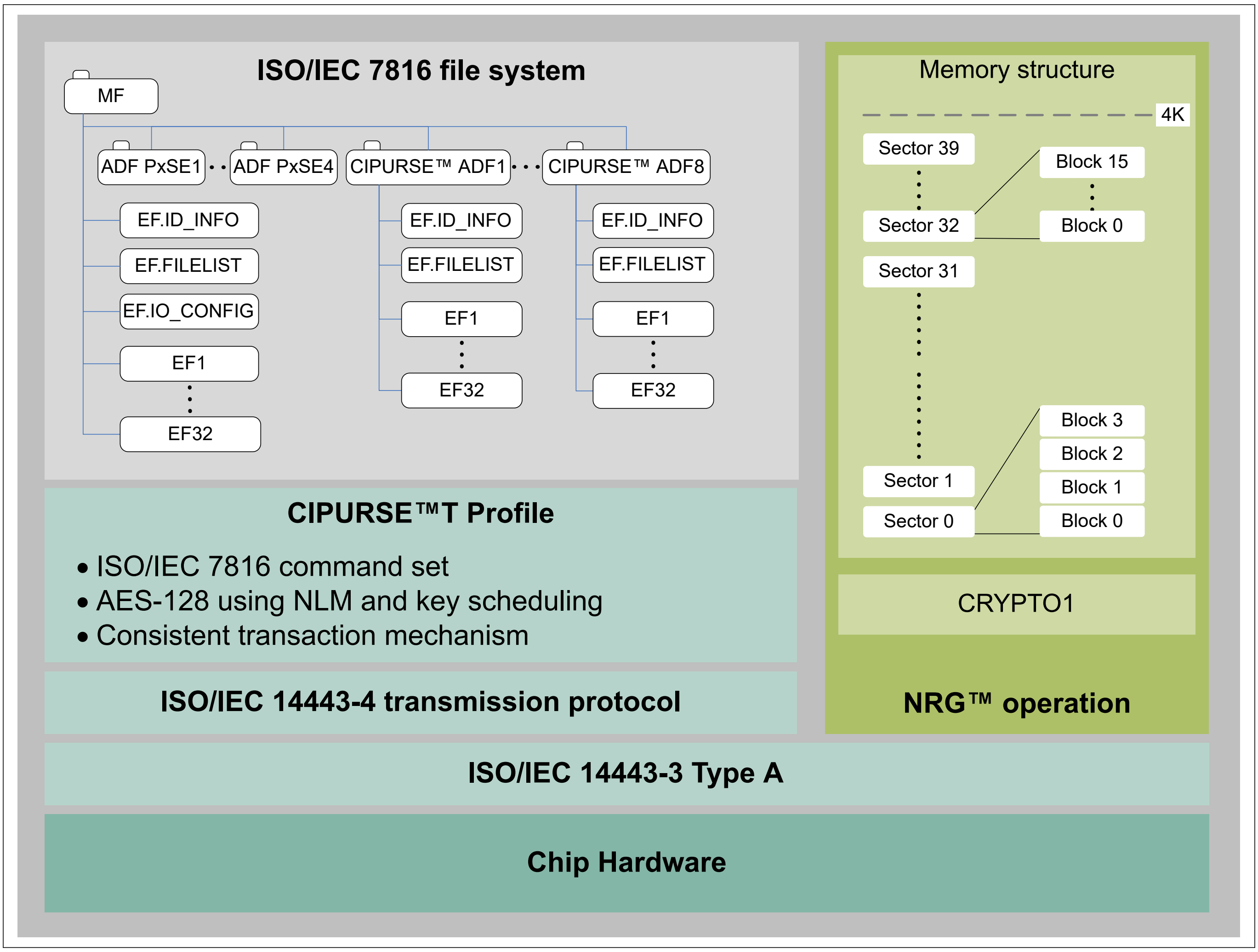

Figure 2 **Block diagram of CIPURSE™Security Controller**

General features

- 8 KB user memory
- Optional support of 1 KB and 4 KB block oriented memory with NRG™ operation
- 27/56/78 pF chip input capacitance
- Operating temperature range: -25°C to +85°C (for chip)[1)]
- Storage temperature range: -40°C to +125°C (for chip)[1)]

CIPURSE™application security

CIPURSE™Security Controller supports:

- Up to 8 128-bit Advanced Encryption Standard (AES) keys can be assigned to each application dedicated file (ADF)
- Flexible access rights and secure messaging rules configurable for each file
- Mutual authentication using AES-128
- Secure messaging with AES-message authentication code (MAC) and AES-encryption (ENC)

1 For modules according to module specification

- Secure messaging mode configurable for each data exchange
- Secure channel protocol inherently differential power analysis (DPA) and differential fault analysis (DFA) resistant, offering AES-MAC, AES-ENC and sequence integrity protection for application protocol data units (APDUs)
- Consistent transaction mechanism (CTM)
- Administrative functionality
  - 8 128-bit AES keys available for master file (MF) administration
  - MF security architecture is same as CIPURSE™ ADF security architecture

ISO/IEC 7816-4 file system

CIPURSE™Security Controller implements a CIPURSE™ compliant file system based on ISO/IEC 7816-4 [4]:

- Files are organized logically in form of two-level dedicated file (DF) tree structure (as described in Chapter 3.1)
- The MF forms the root of this structure. The MF hosts some predefined elementary files (EFs), up to 32 custom EFs, and up to 8 custom ADFs
- Support up to 4 ADF proximity system environments (PxSEs), in addition to 8 custom ADFs under the MF
- A CIPURSE™ application is represented by an ADF identified by its file identifier (FID) and DF name application identifier (AID). The ADF can host up to 32 custom EFs for application specific data
- Elementary file types supported are binary files, linear record files, cyclic record files, and linear value-record files
  - File size up to 4 KB
  - Up to 254 records per record oriented file
  - Record length up to 228 bytes
- Security attributes defining the access rights and secure messaging rules can be assigned to each ADF, to the MF, and to each EF
- Up to 64 bytes for proprietary security information per ADF
- Up to 64 bytes for proprietary security information for MF
- Consistent transaction mechanism for each file type
- Transaction buffer of at least 1 KB of data[2)]
- Up to 8 KB user memory is available to store an application data. Customers can configure the number of available ADFs, EFs, and the corresponding file size. The maximum file size of one EF is 4 KB

Block oriented memory with NRG™ operation features

As a migration product, CIPURSE™Security Controller is designed to operate in an NRG™ system to support the migration towards more advanced CIPURSE™ security based on AES-128. In addition, the support of NRG™ can be modified (see Chapter 3.1.5.3).

- **SLS 32TLC100M4/SLS 32TLC100M9/SLS 32TLC100ME** – supporting 4 KB block oriented memory with NRG™ operation
  - 32 sectors of 64 bytes (4 blocks)
  - 8 sectors of 256 bytes (16 blocks)
- Two keys per sector
- Mutual three pass authentication
- Encrypted data transfer

Near field communication (NFC) Forum Type 4 Tag

Supports NFC Forum Type 4 Tag functionality, see Chapter 3.1.2.3.

2 Subject to the size of data stored in the file system memory

CIPURSE™ command set

- Multi-level commands
  - SELECT
- Commands for personalization of file system oriented PICCs
  - CREATE_FILE
  - DELETE_FILE
  - FORMAT_ALL
- Commands for object management
  - ACTIVATE_FILE (ADF)
  - DEACTIVATE_FILE (ADF)
- Commands for file attribute management
  - READ_FILE_ATTRIBUTES
  - UPDATE_FILE_ATTRIBUTES
  - UPDATE_KEY
  - UPDATE_KEY_ATTRIBUTES
- Security-related commands
  - MUTUAL_AUTHENTICATE
  - GET_CHALLENGE
- Commands for file data management
  - READ_BINARY
  - UPDATE_BINARY
  - READ_RECORD
  - UPDATE_RECORD
  - APPEND_RECORD
  - READ_VALUE
  - INCREASE_VALUE
  - DECREASE_VALUE
- Commands for transaction management
  - PERFORM_TRANSACTION
  - CANCEL_TRANSACTION

Contactless interface

- Initialization and anticollision according to ISO/IEC 14443-3 [9] Type A using 4-byte reused-ID, 7-byte unique identifier (UID) (Double-Size UID), 10-byte UID (Triple-Size UID), or 4-byte random identification (ID) as defined in ISO/IEC 14443-3 [9]
- Transmission protocol according to ISO/IEC 14443-4 [10]
- Data rates in both directions up to 848 kbit/s

Security features

- Active shield technology
- Anti-snooping features
- Security attack countermeasures for all critical operations using both hardware and software controls
- Access limitation for manufacturer-specific data (configurable)

Certification level

- CIPURSE™V2 certification
- CC EAL 5+ (high)

1.3 Coding and notation conventions

All lengths are represented in bytes, unless otherwise specified.

Each byte is represented by bits b[8:1], where b[8] is the most significant bit and b[1] is the least significant bit, unless otherwise specified.

Multi-byte fields and values are presented in big endian order, unless otherwise specified.

Binary values are specified in brackets with suffix "B" (For example, 0101_B).

Hexadecimal values are specified with suffix "H" (For example, $B4_H$).

2 Ordering and packaging information

Note: The ordering codes for the individual sales code and package combination (For example, SLS 32TLCxxx – MCC8) are available on request.

Table 1 Ordering information

Type	Package
4 KB block oriented memory with NRG™ support, 27 pF chip capacitance	
SLS 32TLC100M4 – MCC8	P-MCC8-2-6[1)]
SLS 32TLC100M4 – MCS8	P-MCS-8-2-1[2)]
SLS 32TLC100M4 – NB	Unsawn/Sawn wafer, NiAu bump[3)]
SLS 32TLC100M4 – C	Unsawn/Sawn wafer, without bump[4)]
4 KB block oriented memory with NRG™ support, 56 pF chip capacitance	
SLS 32TLC100M9 – MCC8	P-MCC8-2-6[1)]
SLS 32TLC100M9 – MCS8	P-MCS-8-2-1[2)]
SLS 32TLC100M9 – NB	Unsawn/Sawn wafer, NiAu bump[3)]
SLS 32TLC100M9 – C	Unsawn/Sawn wafer, without bump[4)]
4 KB block oriented memory with NRG™ support, 78 pF chip capacitance	
SLS 32TLC100ME – MCC8	P-MCC8-2-6[1)]
SLS 32TLC100ME – MCS8	P-MCS-8-2-1[2)]
SLS 32TLC100ME – NB	Unsawn/Sawn wafer, NiAu bump[3)]
SLS 32TLC100ME – C	Unsawn/Sawn wafer, without bump[4)]

1) Pure contactless module (MCC8): for standard thickness inlays (330 µm)
2) Pure contactless module (MCS8): for very thin inlays (< 250 µm)
3) Wafer thickness: 55 µm, 75 µm, and 150 µm with NiAu bump 20 µm
4) Wafer thickness: 55 µm, 75 µm, and 150 µm

Table 2 UID configuration

Type	Delivery state	User configurable[1)]
SLS 32TLC100M4/ SLS 32TLC100M9/ SLS 32TLC100ME	4-byte reused-ID (xM band[2)])	4-byte reused-ID, 7-byte fixed UID, 4-byte random ID, and 10-byte fixed UID

1) The other UID variants can be configured by the customer. For more details, see Chapter 3.1.5.3.
2) M = 1, 5, 7, 9. Other values might be applicable without further notice

Figure 3 **Module contactless card - P-MCC8-2-6**

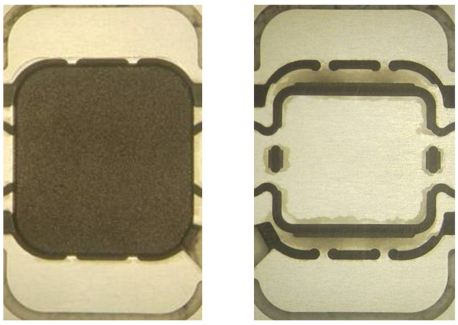

Figure 4 **Module contactless card - P-MCS-8-2-1 (top/bottom view)**

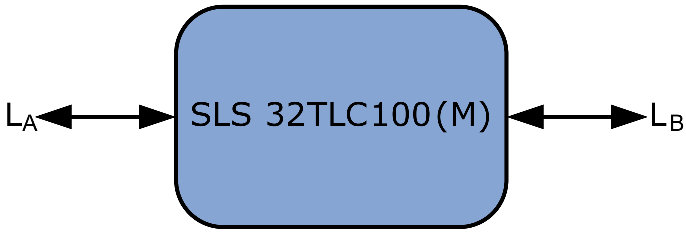

Figure 5 **Pin configuration**

Table 3 **Pin definitions and functions**

Symbol	Function
L_A	Coil connection pin L_A
L_B	Coil connection pin L_B

3 CIPURSE™ application support

3.1 File system of the PICC

The file system implemented by the product is compliant to the file system specified in ISO/IEC 7816-4 [4]. For example, Figure 6 shows the structure of the file system containing a number of CIPURSE™V2 applications and up to 4 PxSE applications.

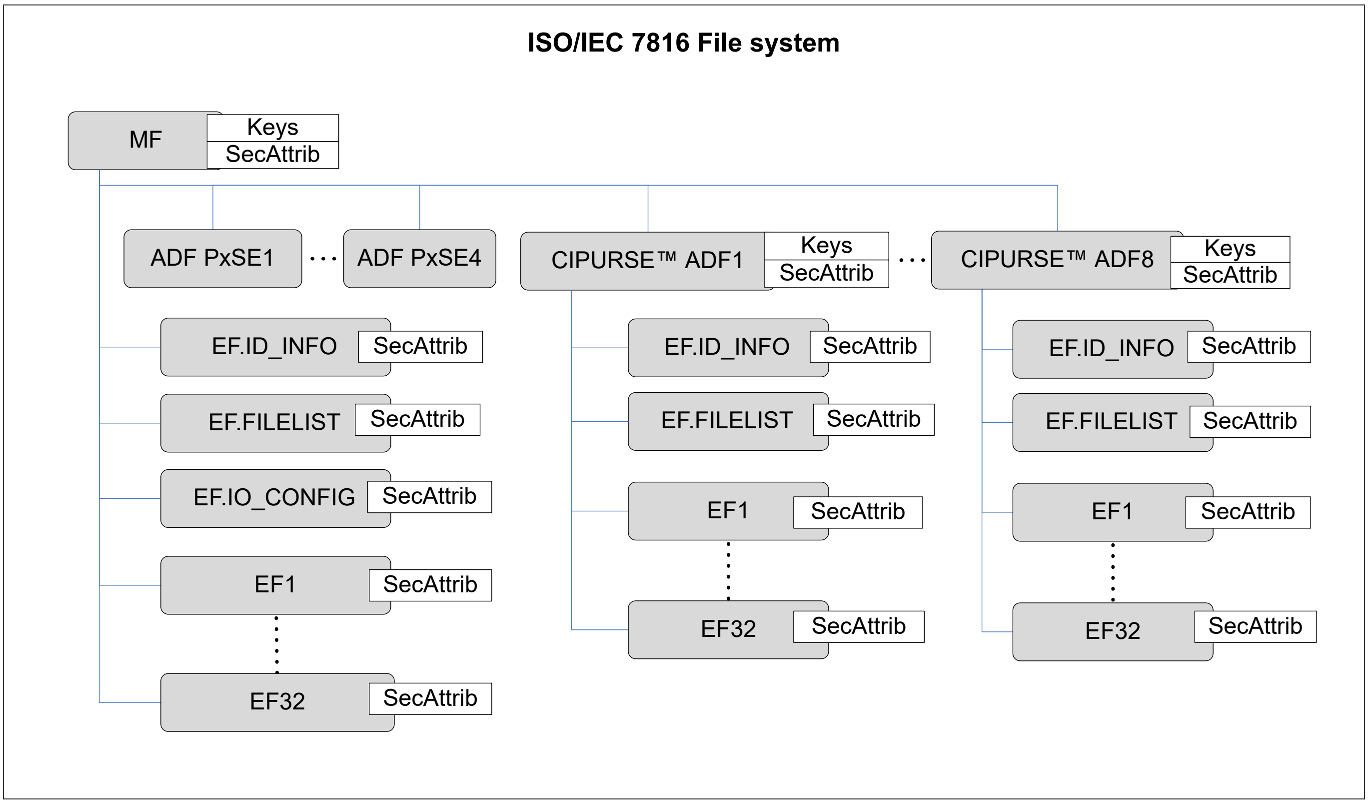

Figure 6 **Example of a CIPURSE™Security Controller file system structure**

For application operation, the files in the file system are organized logically in a form of two-level dedicated file (DF) tree structure. The MF forms the root of this structure.

The MF hosts three predefined EFs and 8 128-bit AES keys and it allows creation of up to 32 custom EFs, up to 4 ADF PxSEs, and up to 8 CIPURSE™ ADFs.

A CIPURSE™ application is represented by an ADF identified by its FID and AID. The ADF hosts two predefined EFs and up to 8 128-bit AES keys and it allows creation of up to 32 EFs.

A PxSE ADF is a specific application, which is created without child files and security attributes.

Security attributes defining the access rights and secure messaging rules may be assigned to each CIPURSE™ ADF, to the MF, and to each EF. The file system offers up to 8 KB memory to store the user data.

3.1.1 Master file

MF consists of keys, security attributes, and hosts custom ADFs (see Chapter 3.1.2) in addition to pre-defined EFs (see Chapter 3.1.5) and custom EFs (see Chapter 3.1.3).

The PICC supports implicit selection of the MF as a result of radio frequency (RF) initialization and anticollision process.

MF supports the following commands:

- CREATE_FILE (ADF/EF)

- DELETE_FILE (ADF/EF)
- FORMAT_ALL
- GET_CHALLENGE
- MUTUAL_AUTHENTICATE
- UPDATE_KEY
- UPDATE_KEY_ATTRIBUTES
- READ_FILE_ATTIRBUTES
- UPDATE_FILE_ATTRIBUTES
- SELECT (by FID/AID)

CTM (see Chapter 3.1.4.2) will also be applicable for commands manipulating MF attributes including the list of child EFs.

3.1.2 Application dedicated files

An ADF is identified by its AID or by its FID.

PICC supports three type of ADFs:

- CIPURSE™ ADF
- PxSE ADF
- NFC Type 4 Tag ADF

3.1.2.1 CIPURSE™ ADF

CIPURSE™ ADF consists of keys and security attributes, and it hosts the EFs with application-specific data as described in Chapter 3.1.3 in addition to pre-defined EFs (see Chapter 3.1.5).

CIPURSE™ ADF can be secured or unsecured based on the security attributes defining access conditions and secure messaging, and key values as described in Chapter 3.2.

CIPURSE™ ADF supports two operational states:

- ACTIVATED
- DEACTIVATED

Command ACTIVATE_FILE (ADF) activates the referenced CIPURSE™ ADF (and inherently all its child EFs) from its deactivated state.

An activated CIPURSE™ ADF supports the following commands:

- CREATE_FILE (EF)
- DELETE_FILE (this ADF/EF)
- GET_CHALLENGE
- MUTUAL_AUTHENTICATE
- UPDATE_KEY
- UPDATE_KEY_ATTRIBUTES
- READ_FILE_ATTIRBUTES
- UPDATE_FILE_ATTRIBUTES
- SELECT (by FID/AID)
- DEACTIVATE_FILE (ADF)

Command DEACTIVATE_FILE (ADF) deactivates the activated CIPURSE™ ADF (and implicitly all its child EFs).

A deactivated CIPURSE™ ADF supports the following operational commands:

- SELECT (by FID/AID)
- ACTIVATE_FILE (subject to access condition)

- GET_CHALLENGE
- MUTUAL_AUTHENTICATE

CIPURSE™ ADF supports a CTM (see Chapter 3.1.4.2); EF creation, new key values, key attributes, or file attributes become effective after successful execution of PERFORM_TRANSACTION.

3.1.2.2 PxSE ADF

PxSE application registers the segment specific CIPURSE™ applications such as dedicated to transport applications, event ticketing applications, and facility access applications.

PxSE application supports the SELECT (by AID) command only.

The response to SELECT PxSE provides the list of AIDs corresponding to its registered CIPURSE™ applications in ACTIVATED state and one of its registered applications might be implicitly selected.

3.1.2.3 NFC Type 4 Tag ADF

The product supports an NFC Type 4 Tag ADF [11] with the same functionality as a CIPURSE™ ADF with the following exceptions during ADF creation:

- EF.ID_INFO is not automatically created
- EF.FILELIST is not automatically created

The creation of EF with the same FID as EF.ID_INFO or EF.FILELIST is not allowed.

3.1.3 Supported elementary file types

EFs are used to store data and are identified by its FID or by short file identifier (SFID).

The file system supports the following elementary file types:

- Binary file
- Linear record file
- Cyclic record file
- Linear value-record file

Every elementary file type is available in the following two flavors:

- Version not supporting CTM
- Version supporting CTM

EFs can be secured or unsecured based on the security attributes as described in Chapter 3.2.

The commands READ_FILE_ATTRIBUTES and UPDATE_FILE_ATTRIBUTES can be used to read and update the EF attributes.

Binary file:

A binary file represents a series of sequential bytes without specific inner structure. Size of the file is defined at file creation.

On file creation, the data are created and initialized with zeros. The commands READ_BINARY and UPDATE_BINARY can be used to read and update the records.

The maximum size of the binary file is restricted to 4 KB.

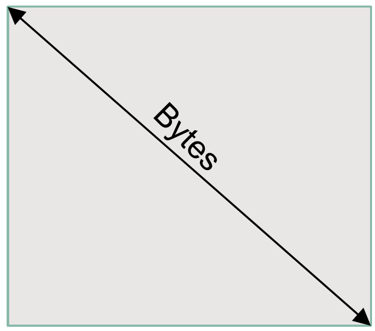

Figure 7 **Binary file**

Linear record file:

A linear record file represents a linear sequence of records of same size. Size and number of records are defined at file creation.

On file creation, all records are created and initialized with zeros. The commands READ_RECORD and UPDATE_RECORD can be used to read and update the records.

The maximum size of a record is 228 bytes. A file can contain maximum of 254 records. The maximum size of the linear record file (size of record x number of records) is restricted to 4 KB.

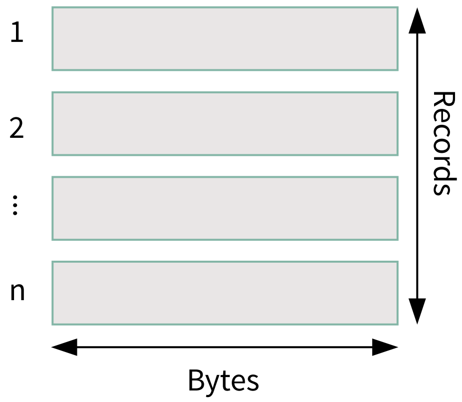

Figure 8 **Linear record file**

Cyclic record file:

A cyclic record file represents a cyclic sequence of records, where the oldest data will be overwritten, in case the list is full. The size and number of the records are defined at file creation.

On file creation, only the memory is reserved. No further initialization is performed. Each record must be created and initialized using command APPEND_RECORD before it can be read or updated. The commands READ_RECORD and UPDATE_RECORD can be used to read and update the records.

The maximum size of a record is 228 bytes. A file can contain maximum of 254 records. The maximum size of the cyclic record file (size of record x number of records) is restricted to 4 KB.

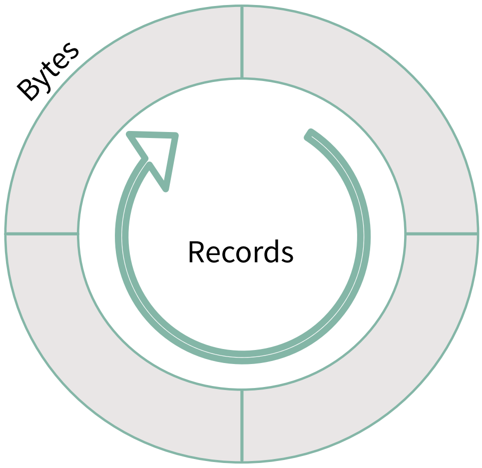

Figure 9 **Cyclic record file**

Value-record file:

A value-record file represents a linear sequence of records of 12 bytes. Each value-record contains maximum and minimum limit and a counter value field. Number of records is defined at file creation.

On file creation, all records are created and initialized with 0000 0000_H (counter value), 7FFF $FFFF_H$ (maximum limit), and 8000 0000_H (minimum limit). The commands READ_RECORD and UPDATE_RECORD can be used to read and update the records. The commands READ_VALUE, INCREASE_VALUE, and DECREASE_VALUE can be used to read and manipulate the counter values. If modification of the value violates the limits, the command will be rejected.

A file can contain maximum of 254 records.

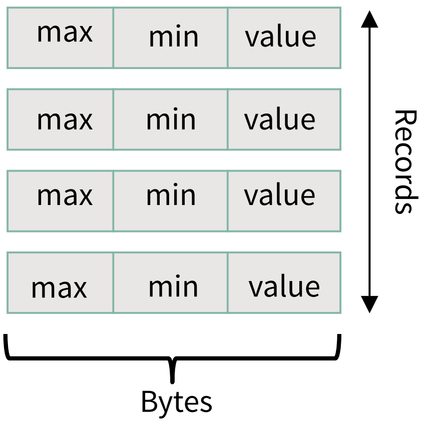

Figure 10 **Value-record file**

3.1.4 Consistent data update mechanisms

CIPURSE™Security Controller supports ‘command level atomicity’ and ‘consistent transaction mechanism’ to avoid inconsistent data update.

3.1.4.1 Command level atomicity

Either all data updates on the PICC are successful during the execution of a single command or no updates at all.

3.1.4.2 Consistent transaction mechanism

The CTM provides consistent data updates and protection from tearing, that is all updates on one or multiple files by sequence of commands are committed "at once".

This mechanism is implemented only on files supporting CTM.

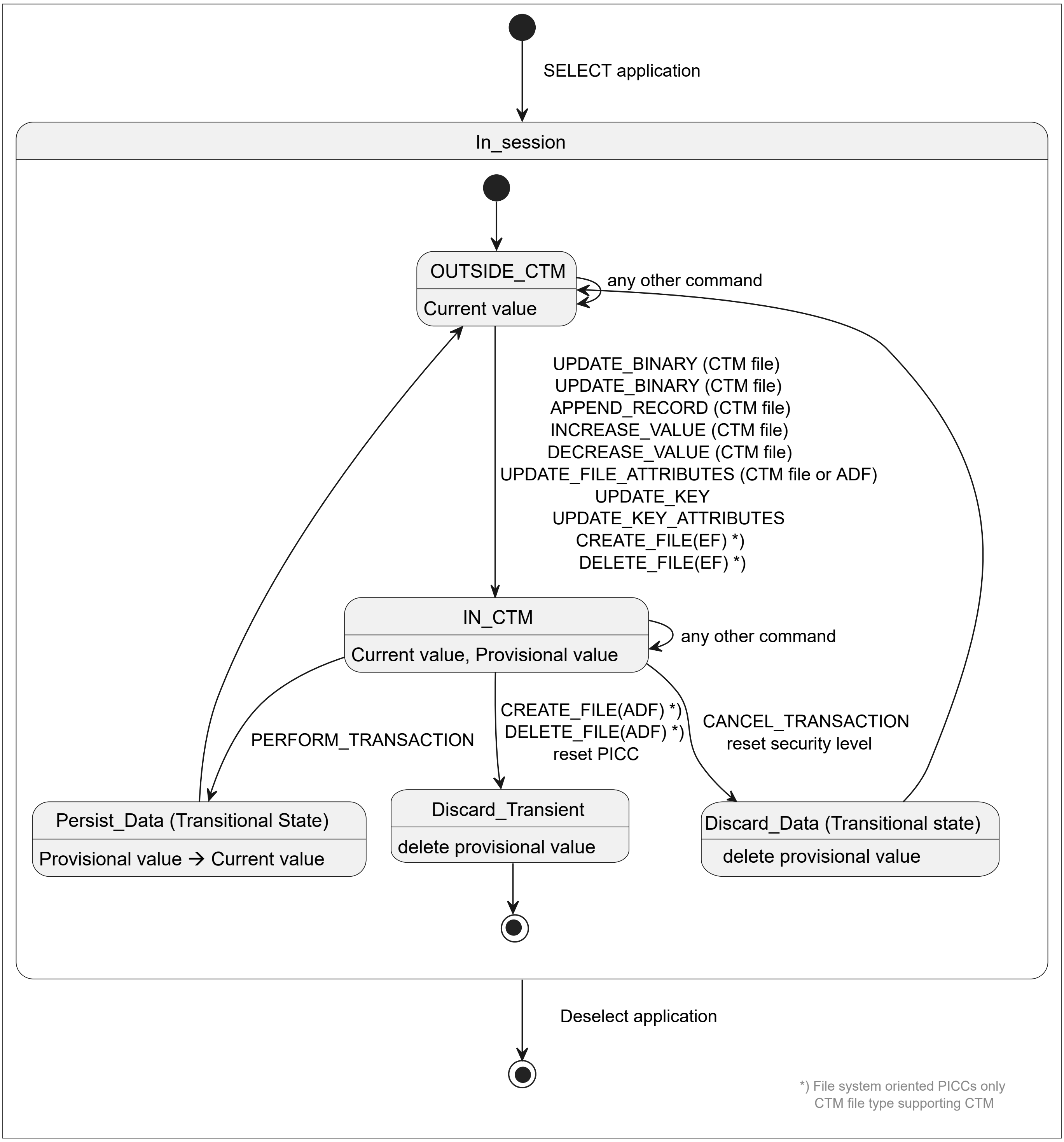

Figure 11 **CTM states diagram**

An application session starts after the selection in OUTSIDE_CTM state with no consistent transaction is in progress.

Command updating data or attributes of an EF supporting CTM or updating ADF attributes results in transition to IN_CTM state and manipulated data are stored as provisional values.

IN_CTM state is left in the following cases:

- Command PERFORM_TRANSACTION will persist all the provisional values and make them as current values before reaching the OUTSIDE_CTM state
- Command CANCEL_TRANSACTION or by resetting the security level to none will discard all the provisional values and retain the current values before reaching the OUTSIDE_CTM state
- Command CREATE_ADF (on MF level), deleting the current application, or resetting the PICC will change the state to Discard_Transient, delete the provisional values, and terminate the application session

3.1.5 Predefined elementary files

Predefined EFs under the MF are present at delivery state, need not be created and cannot be deleted. The security attributes can be modified.

Predefined EFs under the ADF are implicitly created during ADF creation. Deletion is only possible by deleting the parent ADF. The security attributes can be modified.

Table 4 List of predefined EFs

File name	File type	CTM support	Description
EF.FILELIST	Binary	No	Read-only file under the MF/ADF providing list of files under the MF/ADF
EF.ID_INFO	Binary	No	Read-only file under the MF/ADF providing information about the CIPURSE™ version and features along with the manufacturer specific information
EF.IO_CONFIG	Binary	No	File under the MF provides information about the interface configuration parameter and ATR content

3.1.5.1 EF.FILELIST

The EF.FILELIST (under the MF/ADF) is read-only file and provides a 4-byte file information for each file present under the MF/ADF. The size of EF.FILELIST varies depending on the number of files currently present in the MF/ADF.

Table 5 Structure and contents of EF.FILELIST

EF.FILELIST	Type: Binary, read-only		
Content		**Length [byte]**	**Description**
File #1	FID	2	File identifier of File #1
	SFID	1	Short file identifier of File #1
	FD	1	File descriptor byte of File #1
		Var.	Further FID || SFID || FD fields...
File #n	FID	2	File identifier of File #n
	SFID	1	Short file identifier of File #n
	FD	1	File descriptor byte of File #n

3.1.5.2 EF.ID_INFO

The predefined file EF.ID_INFO is a read-only file and is available under the MF and CIPURSE™ ADF. EF.ID_INFO files are identical across all applications in one PICC.

The structure and content of the EF.ID_INFO file are as described Table 6.

Table 6 Structure and content of EF.ID_INFO

EF.ID_INFO	Type: Binary, Read-only
Offset	**Description**
0-7	CIPURSE™ version along with features (CTM and file system oriented personalization) are supported
8	Integrated circuit manufacturer, as per ISO/IEC 7816-6 [5]: • 05_H: Infineon Technologies
9-23	Chip identification data
24-32	Reserved for further manufacturer information
33	Specifies whether 1 KB or 4 KB block oriented memory with NRG™ operations are supported
34-36	Software version
37-39	Product identifier

3.1.5.3 EF.IO_CONFIG

The EF.IO_CONFIG file under the MF contains IO configuration parameters as defined in the Table 7. The IO interface configuration of the product can be modified by updating the parameters in this file.

Table 7 Structure and contents of EF.IO_CONFIG

EF.IO_CONFIG	Type: Binary
Offset	**Description**
0-1	Tag and length for contactless I/O parameters
2	Protocol type and configurable UID mode[1)]
3	Configuration state of block oriented memory with NRG™ operation: • Block oriented memory with NRG™ operation is deactivated • Support for 1 KB block oriented memory with NRG™ operation is activated • Support for 4 KB block oriented memory with NRG™ operation is activated
4	Reserved for future use (RFU)
5	Interface bytes for Type A and frame size for proximity card integer (FSCI)
6	Baudrate
7	Frame waiting time integer (FWI) and start-up frame guard time integer
8	Node address (NAD) and card identifier (CID) support indicator
9-10	Tag for additional parameters. Length of this tag indicates the length of the historical bytes returned as part of answer to select (ATS). This value can be configured to be in the range 0 to 15 bytes. Default value is set to 7 bytes

(table continues...)

Table 7 (continued) Structure and contents of EF.IO_CONFIG

EF.IO_CONFIG	**Type: Binary**
Offset	**Description**
11-17	Initial historical bytes: • Controller control byte • Product identifier bytes • Software version bytes
18-25	Additional bytes to allow extending historical bytes. It is recommended to set these bytes to 00_H

1) 4-byte reused-ID, 7-byte fixed UID, 4-byte random ID, and 10-byte fixed UID

3.1.6 File referencing methods

To access the data, the files in a CIPURSE™ conforming PICC can be selected by using the following methods (Explicit selection or Implicit selection).

Explicit selection:

- A SELECT command is used for explicit selection mode
- A different combination of the parameters along with the SELECT command will perform the explicit selection such as:
  - For explicit selection of MF, the SELECT command with FID $3F00_H$ can be used
  - For explicit selection of ADF, the SELECT command with AID or an FID can be used
  - For explicit selection of EF, the SELECT command with FID or a command supporting addressing by SFID can be used

Implicit selection:

- RF initialization and anticollision process is used for implicit selection of MF
- Selection of a PxSE application may result in implicit selection of one of its registered ADFs
- Implicit selection of EF is not supported

3.1.7 Reserved file identifiers

Some of the FIDs are reserved to serve a special purpose such as file identifiers of MF and pre-defined EFs.

3.2 Security architecture

The security architecture of this product consists of keys representing the various roles, an authentication mechanism to check the availability of a key, and the file security attributes to grant access to entitled roles only.

The security architecture is intended to restrict the access and operations on the application's data to authorized entities only.

Before executing a command on a secured object, the PICC checks if the security requirements are met in terms of file security attributes which are access rights and secure messaging rules.

3.2.1 Keys

AES-128 bit keys are used for authentication. Keys are associated to ADF/MF.

Each key has a set of secure and non-secure attributes as defined below:

- Secure key attributes are used to control the operations permissible with/on this key such as if the key can be updated or is immutable, and if the key is valid or invalid
- Non-secure key attributes hold an additional key information and cryptographic algorithm identifier

3.2.2 Mutual authentication and security state

Figure 12 shows the states and resulting security levels reached when a terminal sends the commands GET_CHALLEGE and MUTUAL_AUTHENTICATE to mutually authenticate both terminal and PICC.

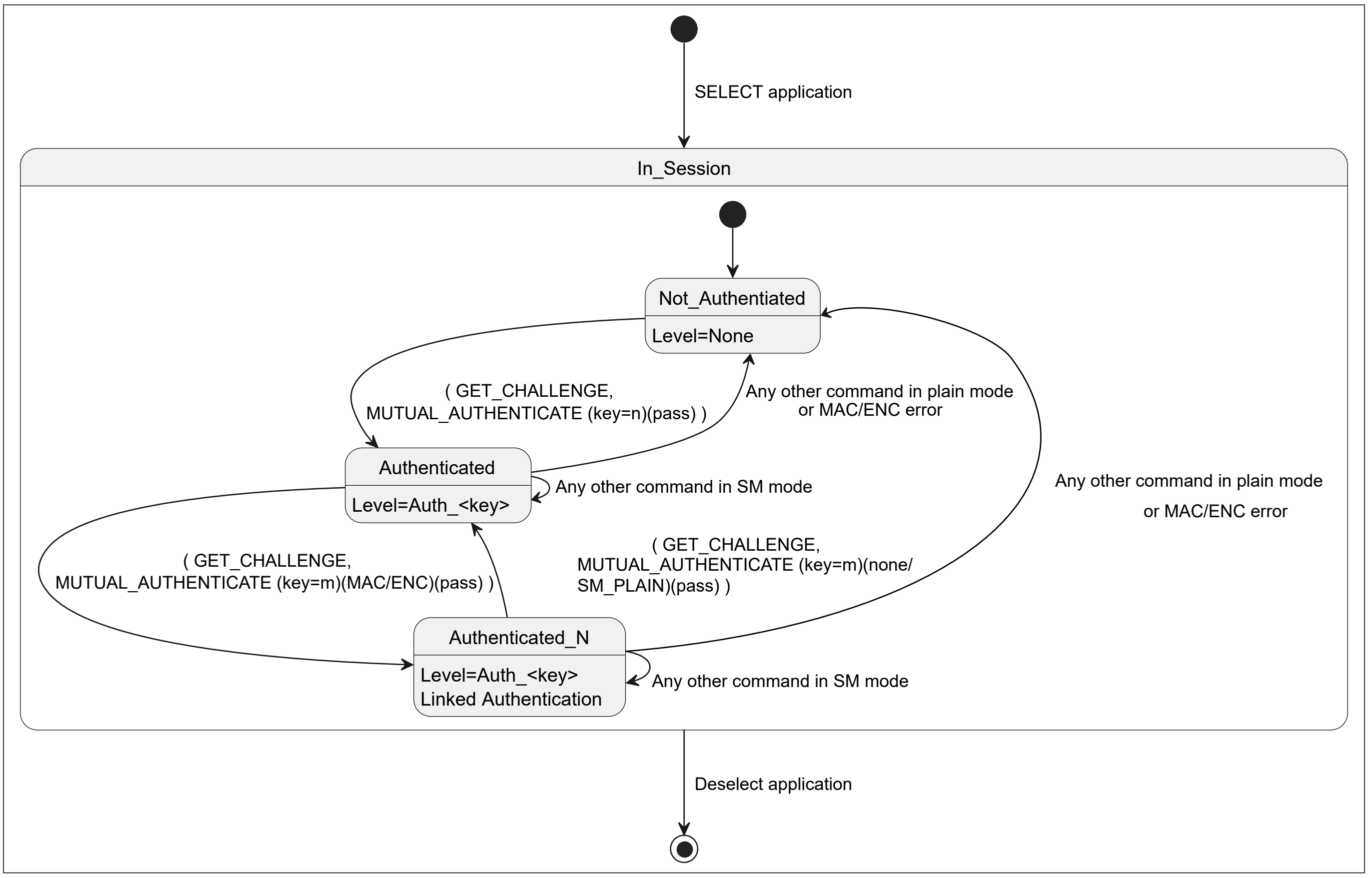

Figure 12 Authentication states and security level

After selection of the application owning the keys, the application is in Not_Authenticated state with security level none.

- A GET_CHALLENGE command followed by MUTUAL_AUTHENTICATE command with valid cryptogram results in a transition to Authenticated state with security level Auth_<key> referencing the key number used for authentication

In Authenticated state, all commands must be transmitted in secure channel mode.

- A GET_CHALLENGE command followed by a MUTUAL_AUTHENTICATE command with valid cryptogram, received in SM_MAC or SM_ENC mode, and referencing a new key will result in Authenticated_N state with "linked authentication" where the previous state's security level Auth_<key> is retained and the security level will change from Auth_<old key> to Auth_<new key>

In Authenticated_N state, all commands must be transmitted in secure channel mode.

- A GET_CHALLENGE command followed by a MUTUAL_AUTHENTICATE command with valid cryptogram, received without secure channel or secure messaging with plain data (SM_PLAIN), will result in Authenticated state with no "linked authentication" where the security level will reset to Auth_<new key>

Any command received in plain mode or in secure messaging (SM) mode with invalid cryptogram will reset the state to Not_Authenticated with security level none.

When a security level Auth_<key> is reached, the terminal acquires the right to execute the commands that are granted to this security level, as described in Chapter 3.2.3.

3.2.3 Access rights

Access rights grant each security level rights to execute various commands respective to a file type. Also, it defines unconditional access ("ALWAYS") to enable PCDs to execute commands irrespective of the security level reached and the secure messaging rules assigned to the file, see Chapter 3.2.4.

If none of the rights are enabled, the commands cannot be executed irrespective of the security level.

3.2.4 Secure messaging rules

Secure messaging rules (SMR) define for a file, the minimum secure messaging levels required to execute various commands respective to a file type.

There are three different secure messaging levels available, as follows:

- SM_PLAIN: Data is sent in plain and the transferred command does not include an integrity protection field
- SM_MAC: Integrity-protected communication with a field of MAC in the transferred command and the data is sent in plain
- SM_ENC: Confidential communication with encryption of data and integrity protection field in the transferred command

The PCD defines the communication security level applicable for exchanging the messages between PCD and PICC.

The PICC evaluates if the chosen security level is acceptable for the addressed file and operation.

3.3 Command set

This section defines all the commands available for operation of CIPURSE™ application.

Table 8 Overview of CIPURSE™ commands

Command	Description
Multi-level commands	
SELECT	Selects the file (MF, ADF, or EF)
Commands for personalization of file system oriented PICCs	
CREATE_FILE (ADF, EF)	Creates an ADF or an EF in the PICC file system
DELETE_FILE (ADF, EF)	Deletes an ADF or an EF from the PICC file system
FORMAT_ALL	Formats the file system to its initial data state The MF keys, MF key attributes, and the content and attributes of predefined EFs under the MF are not formatted
Commands for object management	
ACTIVATE_FILE (ADF)	Activates an ADF in the PICC file system
DEACTIVATE_FILE (ADF)	Deactivates an ADF in the PICC file system
Commands for file attribute management	
READ_FILE_ATTRIBUTES	Reads the MF, DF, or EF file attributes
UPDATE_FILE_ATTRIBUTES	Updates the MF, DF, or EF file attributes
UPDATE_KEY	Updates the value of a key in the PICC
UPDATE_KEY_ATTRIBUTES	Updates the attributes of a key in the PICC
Security related commands	
MUTUAL_AUTHENTICATE	Mutual authentication with the PICC

(table continues...)

Table 8 (continued) Overview of CIPURSE™ commands

Command	Description
GET_CHALLENGE	Retrieves the challenge information from the PICC in order to proceed with authentication
Commands for file data management	
READ_BINARY	Reads a data from a binary file
UPDATE_BINARY	Updates a data into a binary file
READ_RECORD	Reads a records from a record file or a value record file
UPDATE_RECORD	Updates a data into an existing record in a record file or a value record file
APPEND_RECORD	Appends a record to a cyclic record file that is not already full
READ_VALUE	Reads a value from a value record file
INCREASE_VALUE	Increases the value in a value record file
DECREASE_VALUE	Decreases the value in a value record file
Commands for transaction management	
PERFORM_TRANSACTION	Finalizes a transaction that is in progress
CANCEL_TRANSACTION	Cancels a transaction that is in progress

4 Contactless I/O functionality

CIPURSE™Security Controller supports contactless I/O communication according to ISO/IEC 14443-3 [9] and ISO/IEC 14443-4 [10] and as configured in EF.IO_CONFIG at the time of manufacturing of the product.

4.1 Communication principle

All operations on the PICC are initiated by an appropriate reader and controlled by the internal logic of CIPURSE™Security Controller. Prior to any application specific operations, the PICC has to be selected according to the ISO/IEC 14443-3 [9] Type A anticollision and selection scheme.

After selection, the PCD may proceed as follows:

- Enter the NRG™ operation state (CIPURSE™Security Controller devices supporting NRG™ operation only) or
- Enter ISO/IEC 14443-4 [10] transmission protocol processing (T=CL) by sending a RATS command

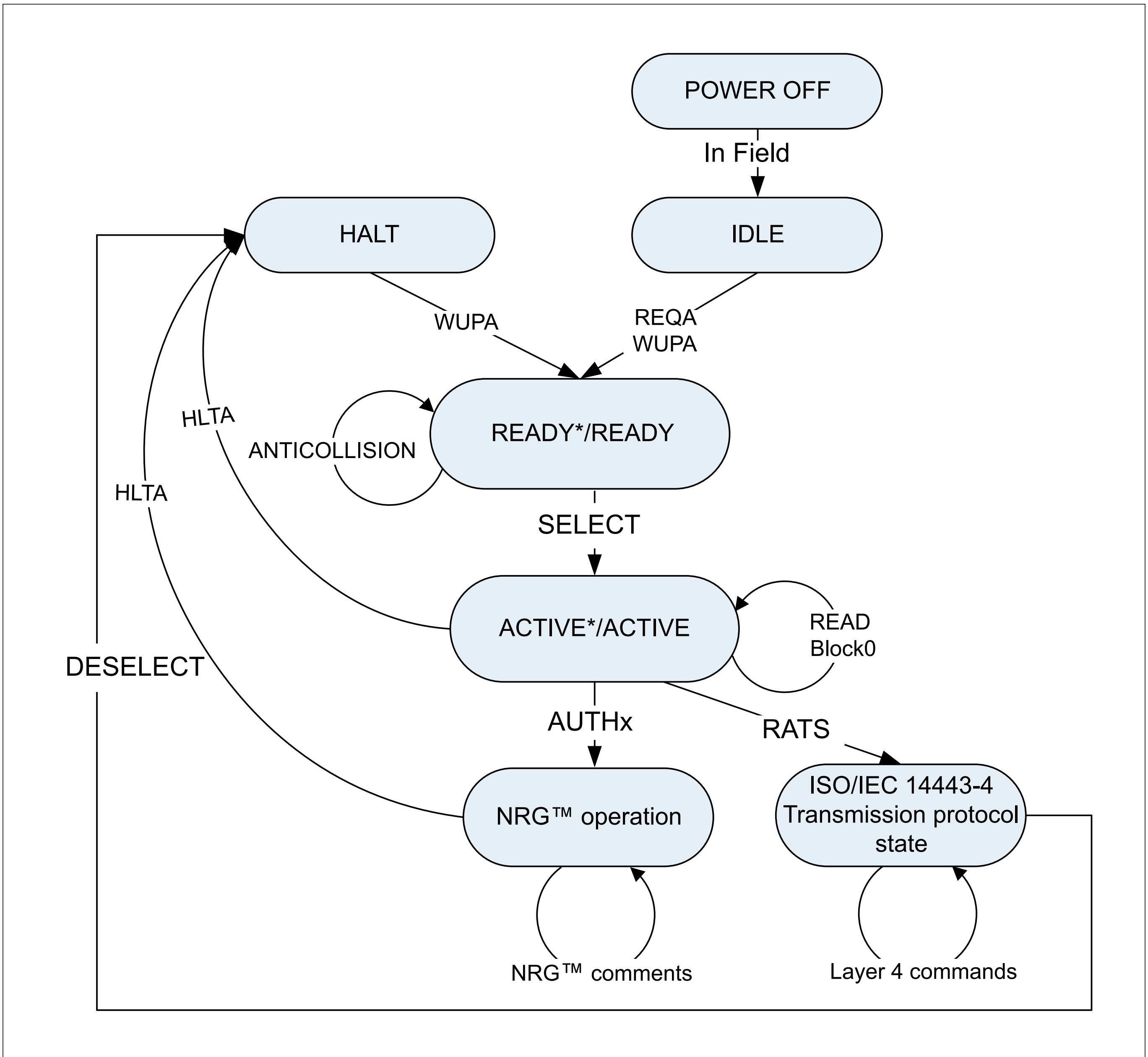

Figure 13 **CIPURSE™Security Controller communication state diagram according to ISO/IEC 14443-3 Type A**

4.2 ISO/IEC 14443 feature set

The following features and types of commands are available:

- Commands for radio frequency (RF) initialization and bit frame anticollision as per ISO/IEC 14443-3 [9], Type A
- Commands for operating the half-duplex block transmission protocol as per ISO/IEC 14443-4 [10], with the following feature profile:
  - Card identifier (CID) is supported, which enables the PCD to select and operate more than one PICC simultaneously
  - PICC and PCD chaining is supported
  - Node address (NAD) is supported
  - Power level indication inside the CID is not supported
- The error handling is performed as defined in ISO/IEC 14443-3 [9] and ISO/IEC 14443-4 [10]

5 Block oriented memory with NRG™

Block oriented memory communicating via ISO/IEC 14443-3 [9] Type A, and offers a proprietary command set for application operation. It features the confidential CRYPTO1 [12] stream cipher. Mutual authentication according to ISO/IEC 9798-2 [6] is used to set up the stream ciphering, which applies to the whole subsequent data exchanged over the RF link.

5.1 Operation of a block oriented memory with NRG™

The PCD and PICC must use a bit rate of $128/f_C$ (~106 kbit/s) in both directions for all commands and responses, with the characteristics as specified by ISO/IEC 14443-3 [9].

First, the PCD and the PICC perform the initialization and anticollision procedure as described in ISO/IEC 14443-3 [9].

With the PICC in ACTIVE/ACTIVE* state, the PCD can initiate the authentication procedure by sending the AUTHENTICATE command or send the READ block 00_H command plain (unencrypted) once or multiple times before initiating the authentication procedure. After completion of the authentication procedure, the PICC enters the authenticated state. So, all further communication in this state must be encrypted by the CRYPTO1 stream cipher.

The PICC exits the authenticated state on reception of the encrypted HLTA command, performing its transition to the HALT state, or in case of error.

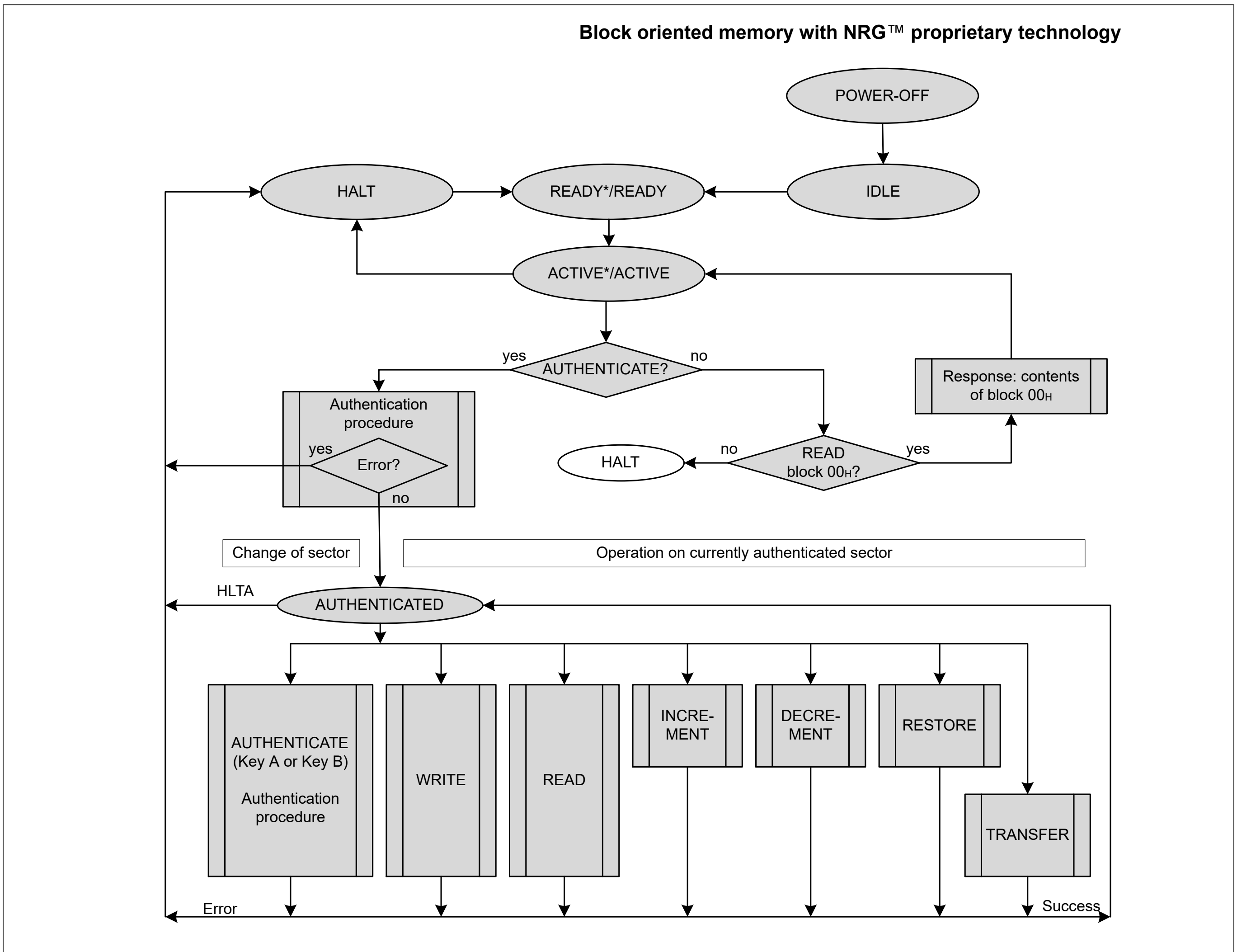

Figure 14 **Block oriented memory with NRG™ operation (initialization and anticollision procedure with 4-byte UID)**

Note: State transitions due to successful command execution are shown in this diagram.

5.2 Memory organization

Memory accessible in NRG™ mode is organized into blocks of 16 bytes. These blocks are accessible as elementary data units using the NRG™ command set (see Chapter 5.3) and thus no single byte level access is allowed. Further on, blocks are grouped into sectors as described below:

- SLS 32TLC100M4/SLS 32TLC100M9/SLS 32TLC100ME (4 KB block oriented memory with NRG™)
  - 32 sectors of 4 blocks each
  - 8 sectors of 16 blocks each

Supports both 1 KB and 4 KB NVM with NRG™. 4 KB NVM with NRG™ is the delivery configuration and 1 KB NVM with NRG™ can be configured by the user.

5.2.1 1 KB non-volatile memory

This section describes the PICCs offering 1 KB of non-volatile memory (NVM) available for the purpose of NRG™ operation.

Structure and properties

Sector Number	Block Address	Block Number	Byte Number within a Block																Description
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
15	$3F_H$	3	Authentication Key A						Access Bits			RFU	Authentication Key B (optional) or Data						Sector Trailer
	$3E_H$	2																	Data
	$3D_H$	1																	Data
	$3C_H$	0																	Data
14	$3B_H$	3	Authentication Key A						Access Bits			RFU	Authentication Key B (optional) or Data						Sector Trailer
	$3A_H$	2																	Data
	39_H	1																	Data
	38_H	0																	Data
...	...	...	...																...
1	07_H	3	Authentication Key A						Access Bits			RFU	Authentication Key B (optional) or Data						Sector Trailer
	06_H	2																	Data
	05_H	1																	Data
	04_H	0																	Data
0	03_H	3	Authentication Key A						Access Bits			RFU	Authentication Key B (optional) or Data						Sector Trailer
	02_H	2																	Data
	01_H	1																	Data
	00_H	0																	Manufacturer Data

Figure 15 Memory structure of 1 KB of NVM with NRG™

The memory of PICC offering 1 KB of NVM with NRG™ is structured as described below:

- The memory is organized in 16 sectors, each with 4 blocks with 16 bytes of data each. A block is the elementary unit addressable by NRG™ commands. The numbering of sectors and blocks is in ascending order of their addresses, as shown in Figure 15
- A successful authentication procedure to the sector where the addressed block is located must be carried out to allow the PCD to apply the appropriate commands to the block

- Blocks 0, 1, and 2 of each sector are available for application data, configurable in two ways:
  - Arbitrarily usable data blocks as specified in Data block
  - Blocks formatted as specified in Value block
- Block 3 of each sector (denoted as "sector trailer") has the following properties:
  - This block contains either one or two cryptographic keys of 6 byte each (Key A is mandatory, Key B is optional) for authentication to get access to the blocks in this sector, and 3 bytes of access bits forming the access conditions for all blocks in this sector as specified in Sector trailer

Data block

Data blocks offer to store the bytes in a sequential order. READ and WRITE commands are applicable to the data blocks.

Byte number within a block																Description
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Data																Data block

Figure 16 Structure of a data block

Value block

Value blocks offer to store and manage the dedicated "arithmetic values". The "Value" inside a value block is 4 bytes in length and stored two times in normal and one time in bit-inverted manner. Values must be stored in little endian order.

The arithmetic instructions INCREASE, DECREASE, and RESTORE are applicable to value blocks, where the result is temporarily stored in a volatile transfer buffer. To store the result in the NVM, the TRANSFER command must be used. Besides these commands, READ and WRITE commands are applicable to the value blocks.

Note: For the purpose of this document, the term "transfer buffer" is used in the command set description of the arithmetic instructions. This represents a volatile memory location in the PICC to perform the manipulation of arithmetic values. It cannot be directly accessed with any of the NRG™ commands.

Byte number within a block																Description
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Value				$\overline{\text{Value}}$				Value				Adr	$\overline{\text{Adr}}$	Adr	$\overline{\text{Adr}}$	Value block

Figure 17 Structure of a value block

Sector trailer

The sector trailer contains the authentication keys and the access bits as described below:

- Keys of each 6 byte (Key A is mandatory, Key B is optional) for authentication to all blocks in this sector
- 3 bytes of access bits forming the access condition information for the associated sector, that is access to the blocks along with the sector trailer
- 1 byte is reserved for future use and should not be used for other application data

For more details about sector trailer, see chapter 5.3.2 in [3].

Byte number within a block																Description
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Authentication key A						Access bits			RFU	Authentication key B (optional) or Data						Sector trailer

Figure 18 Structure of a sector trailer

Access condition

Depending on the access condition, the right to execute a particular command to the block results in one of the following conditions:

- Never: Command not granted
- Key A: Command granted when successfully authenticated with Key A of this sector
- Key B: Command granted when successfully authenticated with Key B of this sector
- Key A/B: Command granted when successfully authenticated with Key A or Key B of this sector

The access condition for blocks 0 to 2 and the sector trailer, of the associated sector are formed by the access bits.

Access bits define four access groups: one group for the sector trailer and the remaining groups for data or value blocks each.

5.2.2 4 KB non-volatile memory

This section describes the PICCs offering 4 KB of non-volatile memory (NVM) available for the purpose of NRG™ operation.

Structure and properties

A CIPURSE™Security Controller offering 4 KB of NVM available for the purpose of NRG™ operation is shown in Figure 19.

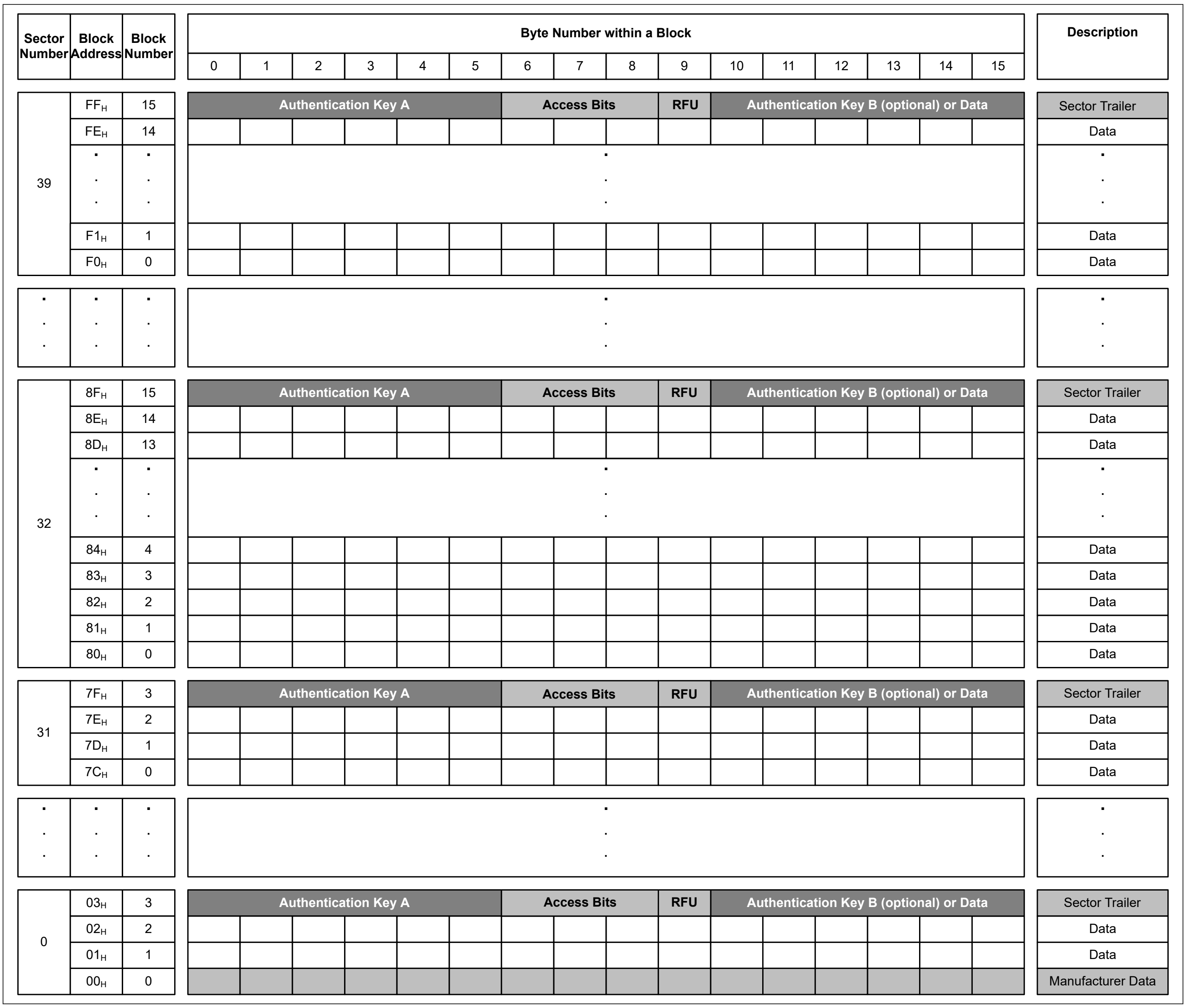

Figure 19 **Memory structure of 4 KB of NVM with NRG™**

The memory of PICC offering 4 KB of NVM with NRG™ is structured as described below:

- The memory is organized in 40 sectors, 32 of them consisting of 4 blocks with 16 bytes of data, and 8 of them consisting of 16 blocks of 16 bytes of data. A block is the elementary unit addressable by NRG™ commands. The numbering of sectors and blocks is in ascending order of their addresses, as shown in Figure 19
- For sectors 0 to 31 (sectors consisting of 4 blocks), the same definitions as for sectors 0 to 15 as specified by Figure 15 are applicable
- For sectors 32 to 39 (sectors consisting of 16 blocks):
  - Blocks 0 to 14 are available for application data, configurable in the same way as for sectors consisting of 4 blocks
  - Block 15 of each sector (denoted as "sector trailer") has similar properties as block 3 for sectors in 1 KB NRG™ (see Chapter 5.2.1) but four access groups are defined by the access bits forming access conditions: one group for the sector trailer and the remaining groups for 5 data or value blocks each

5.3 NRG™ command set

This section describes the commands supported by CIPURSE™Security Controller when it is in NRG™ operation state.

Table 9 Overview of NRG™ commands

Command[1)]	Description
AUTHENTICATE with Key A	Authentication with Key A to the sector in which the addressed block is located
AUTHENTICATE with Key B	Authentication with Key B to the sector in which the addressed block is located
READ	Reads out 16 bytes from memory block via NRG™
WRITE	Writes 16 bytes into memory block via NRG™
DECREMENT	Arithmetic instruction Loads the actual value from the addressed value block decremented by the operand into the transfer buffer
INCREMENT	Arithmetic instruction Loads the actual value from the addressed value block incremented by the operand into the transfer buffer
RESTORE	Arithmetic instruction Loads the actual value of the addressed value block into the transfer buffer
TRANSFER	Transfers the actual value in the transfer buffer to the addressed value block
HLTA	Transition to HALT state as per ISO/IEC 14443-3 [9]

1) For more details about the NRG™ command set, see chapter 7 in [3].

6 Operational characteristics

6.1 Absolute maximum ratings

Stresses above those listed may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this data sheet is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability, including electrically erasable programmable read-only memory (EEPROM) data retention and write/erase endurance.

Table 10 Absolute maximum ratings

Parameter	Symbol	Values			Unit	Note or test condition
		Min.	Typ.	Max.		
Junction temperature	T_J	-40		+110	°C	
Storage temperature	T_{stg}	-40		+125	°C	For chip. For modules according to module specification
ESD protection	V_{ESD}	-2		+2	kV	EIA/JESD22-A114-B

6.2 Electrical characteristics

Table 11 Operation range

Parameter	Symbol	Values			Unit	Note or test condition
		Min.	Typ.	Max.		
Ambient temperature	T_A	-25		+85	°C	T_J must not be exceeded
Endurance (write/erase cycles)[1)]		10^5				
Data retention (years)[1)]		10				T_A = 25°C

1) Values are temperature dependent. For further information please refer to your Infineon Technologies office or representative.

Table 12 Contactless interface characteristics

Parameter	Symbol	Values			Unit	Note or test condition
		Min.	Typ.	Max.		
Operating conditions	H	1.5		7.5	A/m	Reference setup according to ISO/IEC 14443-2 [8] and ISO/IEC 10373-1 [7]
Carrier frequency	f_C		13.56		MHz	±7 kHz
Chip input capacitance	C_{AB}		27/56/78		pF	
Recommended target resonance frequency	f_{res}		16.5		MHz	ID1 (Class 1) card size

References

CIPURSE™/OSPT

[1] OSPT Alliance: *CIPURSE™V2 , Operation and Interface Specification (Revision 2.0), 2013-12-20, incl. Errata and Precision List (Revision 3.0)*; 2017-09-27

[2] OSPT Alliance: *CIPURSE™V2 , CIPURSE™T Profile Specification (Revision 2.0), 2013-12-20, incl. Errata and Precision List (Revision 1.0)*; 2014-09-18

Infineon

[3] Infineon Technologies AG: *SLE66R35x, Extended datasheet (Revision 2.0)*; 2021-05-28

ISO/IEC

[4] ISO/IEC 7816-4:2020: *Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange (Fourth edition)*; 2020-05

[5] ISO/IEC 7816-6:2016: *Identification cards - Integrated circuit cards - Part 6: Interindustry data elements for interchange (Third edition)*; 2016-06

[6] ISO/IEC 9798-2: *Information technology - Security techniques - Entity authentication - Part 2: Mechanisms using symmetric encipherment algorithms (Third Edition)*; 2008-12-15, incl.

- Corrigendum 1, 2010-02-15
- Corrigendum 2, 2012-03-15
- Corrigendum 3, 2013-02-15

[7] ISO/IEC 10373-1:2020-10: *Cards and security devices for personal identification – Test methods - Part 1: General characteristics (Third edition)*; 2020-10

[8] ISO/IEC 14443-2:2020: *Cards and security devices for personal identification – Contactless proximity objects - Part 2: Radio frequency power and signal interface (Fourth edition)*; 2020-07

[9] ISO/IEC 14443-3:2018: *Cards and security devices for personal identification – Contactless proximity objects – Part 3: Initialization and anticollision (Fourth edition)*; 2018-07

[10] ISO/IEC 14443-4:2018: *Cards and security devices for personal identification – Contactless proximity objects – Part 4: Transmission protocols (Fourth edition)*; 2018-06

NFC Forum

[11] NFC Forum: *Type 4 Tag Technical Specification (Version 1.1)*; 2019-12-12

Siemens

[12] Siemens AG: *Semiconductors HL CC PD ID: Crypto-Unit CRYPTO1.DOC*; 1997

Glossary

ADF

application dedicated file (ADF)

AES

Advanced Encryption Standard (AES)
The standard for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001. The algorithm described by AES is a symmetric-key algorithm (i.e. the same key is used for both encryption and decryption).

AID

application identifier (AID)
Used to reference (select) an application.

APDU

application protocol data unit (APDU)
The communication unit between a smart card reader and a smart card.

ATS

answer to select (ATS)

CC

Common Criteria for Information Technology Security Evaluation (CC)
An international standard (ISO/IEC 15408) for computer security certification.

CID

card identifier (CID)

CIPURSE™

Open security standard for transit fare collection systems. CIPURSE™ is a trademark of the Open Standard for Public Transport Alliance.

CTM

consistent transaction mechanism (CTM)

DFA

differential fault analysis (DFA)
A class of side channel attacks in the field of cryptography, specifically cryptographic analysis. Faults are induced into cryptographic implementations with the intention of revealing information about their internal states.

DF

dedicated file (DF)

DPA

differential power analysis (DPA)
A class of attacks against smart cards and secure cryptographic tokens. The attack involves monitoring how much power a microprocessor uses as it functions, then using advanced statistical methods to determine secret keys or personal identification numbers involved in the computations.

EAL

evaluation assurance level (EAL)

EEPROM

electrically erasable programmable read-only memory (EEPROM)

EF

elementary file (EF)
A file system component containing (user) data.

EIA

Electronic Industry Alliance (EIA)

ENC

encryption (ENC)

ESD

electrostatic discharge (ESD)
The sudden draining of electrostatic charge. Even with small charges, it poses a considerable risk to small semiconductor structures, in particular MOS structures. It is therefore essential to take precautions when dealing with unprotected semiconductors.

FD

file descriptor (FD)
Defines the file type (MF, ADF, type of EF).

FID

file identifier (FID)
Used to reference an elementary file.

FWI

frame waiting time integer (FWI)

ID

identification (ID)

IEC

International Electrotechnical Commission (IEC)
The international committee responsible for drawing up electrotechnical standards.

ISO

International Organization for Standardization (ISO)

MAC

message authentication code (MAC)
Used to prove message integrity.

MCC

module contactless card (MCC)

MF

master file (MF)
The root of the CIPURSE™ file system.

NAD
node address (NAD)

NFC
near field communication (NFC)

NRG™

ISO/IEC 14443-3 type A with CRYPTO1

NVM
non-volatile memory (NVM)

OSPT
Open Standard for Public Transport (OSPT)

PCD
proximity coupling device (PCD)
A reader device for NFC cards.

PICC
proximity integrated circuit card (PICC)
A contactless smart card which can be read without inserting it into a reader device.

PxSE
proximity system environment (PxSE)
A generic term for various system-environment applications that are specific to the application family.

RATS
request for answer to select (RATS)

RF
radio frequency (RF)

RFU
reserved for future use (RFU)

SFID
short file identifier (SFID)

SMG
secure messaging group (SMG)
This belongs to the file security attributes. Commands are clustered into SMGs, where each of them lists one or more commands.

SMR
secure messaging rules (SMR)
Object-specific messaging rules combining four SMGs.

SM
secure messaging (SM)
A secure channel that is established between the secure element and a communication partner to ensure confidentiality and authenticity of the exchanged data.

SM_PLAIN

secure messaging with plain data (SM_PLAIN)
Communication with endpoint internal preparation for integrity verification. Data are sent plain, and the transferred frame does not include an integrity protection field.

UID

unique identifier (UID)

Revision history

Reference	Description
Revision 1.0, 2023-01-05 – Valid for product versions 1.2.0	
All	Initial release

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2023-01-05

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about any aspect of this document?

Email: **CSSCustomerService@infineon.com**

Document reference
IFX-fpk1661329163044