CIPURSE™SAM

Datasheet

CIPURSE™SAM is a ready-to-use secure access module and offers secure storage of keys in hardware, cryptographic operations for secure 3-pass mutual authentication, and secure communication between a reader and a card for a variety of applications

Key features

- **Open Standard** of the OSPT Alliance
  - **Interoperability and easy integration** of CIPURSE™ compliant products
  - Up to **512 (128-bit)** keys can be stored across all ADFs and key files for supporting **SDES, AES-128, and 2k-TDES**
  - Up to **80 (256-bit)** keys can be stored across all key files for supporting **AES-192, AES-256, and 3k-TDES**
  - **Mutual authentication** using AES
  - **Secure messaging** using AES-MAC and AES-ENC
- Compliant to **CIPURSE™SAM specification**
- **High-performance 16-bit SLE 78 security controller with Integrity Guard and CC EAL 6+ (high)**
- **CC EAL 6+, 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 CIPURSE™SAM.

Intended audience

This document is primarily intended for application and system designers.

Note: For more details, CIPURSE™SAM V1.2.4 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

The CIPURSE™SAM is a ready-to-use secure access module (SAM) and offers secure storage of keys in hardware, cryptographic operations for secure 3-pass mutual authentication, and secure communication between a reader and a card for a variety of applications like transport ticketing, automatic fare collection (AFC), access control, micro-payment, loyalty, and other related applications.

CIPURSE™SAM is based on the high-performance 16-bit SLE 78 security controller with Integrity Guard and CC EAL 6+ (High), which is used for eID documents of governments and successfully achieved common criteria EAL 6+ security certification as an independent evidence of its outstanding security level.

CIPURSE™SAM incorporates the CIPURSE™ security architecture, augmented by a combination of hardware and software security measures. Commands and transmitted data can be secured and inherently resistant against physical attacks like differential power analysis (DPA) and differential fault analysis (DFA).

CIPURSE™SAM can be used to communicate with the CIPURSE™ complaint products and 1 KB and 4 KB block oriented memory product(s) with NRG™ interface. Further, CIPURSE™SAM can be used to generate and verify cryptograms required for authentication of CIPURSE™ and NRG™ products within a subsequent secure channel.

Therefore, CIPURSE™SAM is the ideal product to support migration from existing non-security or NRG™ legacy systems towards a more advanced, state-of-the-art security architecture and open standard like CIPURSE™.

1.1 System overview

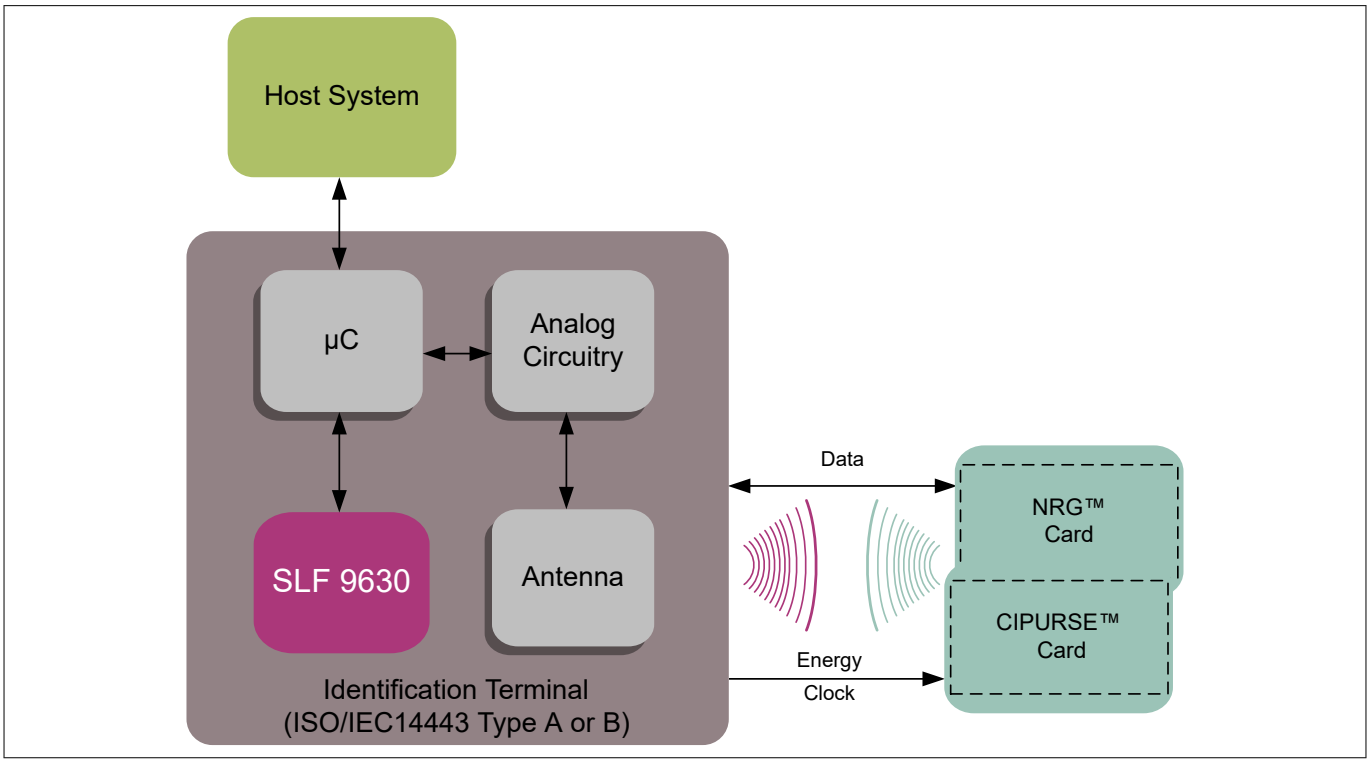

Figure 1 System overview

CIPURSE™SAM - SLF 9630 is connected to a terminal via ISO/IEC 7816-3 [8]. The application-specific terminal may be either connected to a host system (online terminal) or work stand-alone (offline terminal).

The CIPURSE™SAM can be used in an environment, where both CIPURSE™ and NRG™ cards are used.

1 Introduction

CIPURSE™SAM supports 3 modes:

- **Administration mode:** CIPURSE™SAM can be used as a LOAD_SAM (Master SAM) to generate keys and to personalize other client CIPURSE™SAMs by issuing DIVERSIFY_KEYSET and LOAD_KEY commands
  - A back up of the LOAD_SAM can be performed by issuing DIVERSIFY_KEYSET and LOAD_KEY commands to export the keys on LOAD_SAM without diversification
  - The BACK_OFFICE_SAM support administrative functions to verify and to decrypt transaction messages by issuing PERFORM_SYMCRYPTO command
- **Personalization mode:** CIPURSE™SAM can be used in a personalization environment to create cryptograms conveying keys for CREATE_FILE (ADF) and UPDATE_KEY commands to load applications and keys into CIPURSE™ products or products with NRG™ interface
- **Operational mode:** Personalized CIPURSE™ or NRG™ cards can be operated in secure sessions with SAM as follows:
  - AUTHENTICATE_SAM and AUTHENTICATE_CBP commands are used to generate and verify cryptograms required to establish secured session
  - GENERATE_SM_ELEMENTS and VERIFY_SM_ELEMENTS commands are used to generate and verify cryptograms required for data exchange

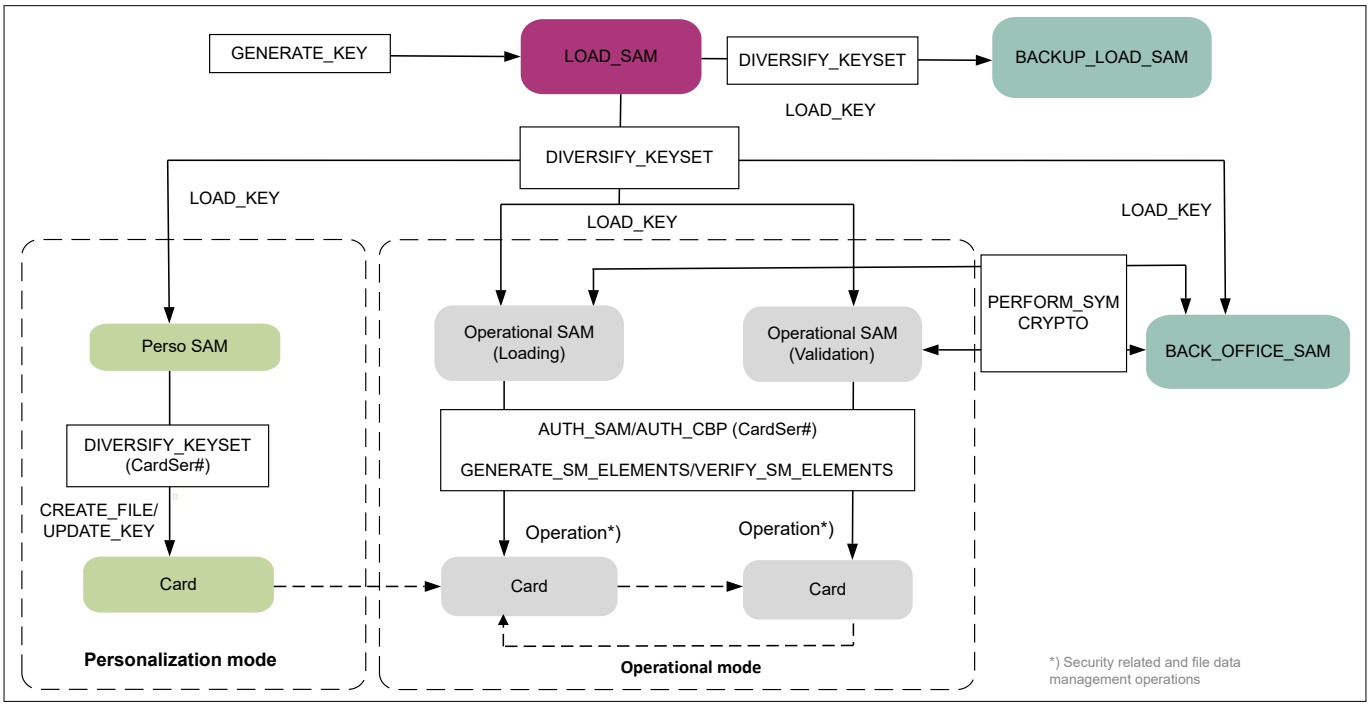

Figure 2 SAM types and key distribution

1.2 Product overview

This product implements SAM functionality for CIPURSE™ based products and products with NRG™ interface. Further this product offers support for generic symmetric cryptographic operations.

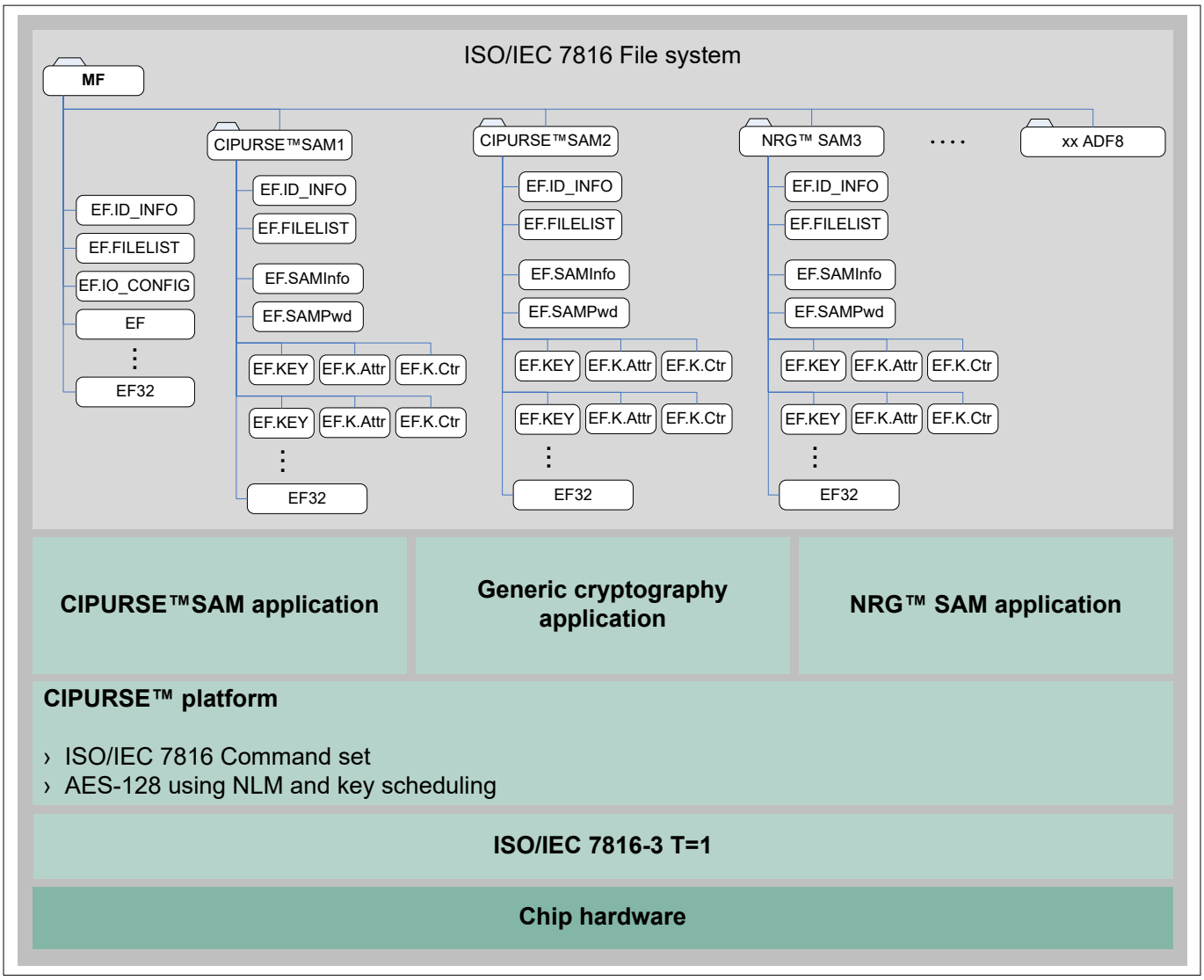

Figure 3 CIPURSE™SAM block diagram

1.2.1 Chip hardware

CIPURSE™SAM software is implemented on the high-performance 16-bit SLE 78 security controller with Integrity Guard and CC EAL 6+ (high).

1.2.2 I/O interfaces

CIPURSE™SAM supports the following interface:

- ISO/IEC 7816-3 T=1

1.2.3 CIPURSE™SAM security

CIPURSE™SAM supports:

- Up to 512 keys that are up to 128-bit long can be stored across all application dedicated files (ADFs) and key files for supporting single DES (SDES), AES-128 and 2k-Triple DES (TDES)
- Up to 80 keys that are up to 256-bit long can be stored across all key files for supporting AES-192, AES-256 and 3k-TDES
- CIPURSE™ application security:
  - Mutual authentication using 128-bit Advanced Encryption Standard (AES) keys
  - Flexible access rights and secure messaging rules can be configured for each file
  - Secure messaging, with AES-message authentication code (MAC) and AES-encryption (ENC)
  - Secure messaging mode configurable for each data exchange
  - Secure channel protocol inherently DPA and DFA offering AES-MAC, AES-ENC, and sequence integrity protection for application protocol data units (APDUs) (except NRG™ cryptography)
- SAM application specific security:
  - Access rights to execute SAM application specific command set are granted based on the SAM application specific security states as defined in Chapter 3.2.1.2, Chapter 3.2.2.3, and Chapter 3.2.3.2
  - Supports transitioning of all SAM application under MF to AUTHORIZED state in the following configurations of EF.SAM_ADMIN_CONFIG (see Chapter 3.6.1).
    - On power-up/reset
    - On VERIFY_SAM_PASSWORD
    - On authentication with a CIPURSE™ key under MF
- Administrative functionality:
  - 8 128-bit AES keys available for MF administration
  - MF security architecture is same as CIPURSE™ ADF security architecture

1.2.4 CIPURSE™SAM application

CIPURSE™SAM implements the CIPURSE™SAM application to provide the following SAM functionality for CIPURSE™ complaint products.

- Supports terminal side cryptography for CIPURSE™ proximity integrated circuit cards (PICCs) with:
  - Three-pass authentication
  - Encrypted data transfer
- Supports the following CIPURSE™ PICCs:
  - CIPURSE™Security Controller [7]
  - CIPURSE™4move [5]
  - CIPURSE™move [6]
- Supports multiple key diversification algorithms
- Supports generic cryptography as described in Chapter 1.2.6

1.2.5 NRG™ SAM application

CIPURSE™SAM implements NRG™ SAM application to provide SAM functionality for NRG™ products.

- Supports terminal side cryptography for NRG™ proximity integrated circuit cards (PICCs) with:
  - Three-pass authentication
  - Encrypted data transfer

- Supports the following NRG™ PICCs
  - 1 KB (16 sectors with 64 blocks in total)
  - 4 KB (40 sectors with 256 blocks in total)
- Supports multiple key diversification algorithms
- Supports generic cryptography as described in Chapter 1.2.6

1.2.6 Generic cryptography

CIPURSE™SAM implements generic cryptography to encrypt and decrypt the arbitrary data and to verify the integrity of arbitrary data. For example, application transaction data can be securely transferred from a terminal to a back office:

- Supports symmetric cryptography – SDES, TDES (2k and 3k), and AES (128, 192, and 256 bit keys)
  - Electronic code book (ECB) – encryption, decryption
  - Cipher block chaining (CBC) – encryption, decryption
  - Compute CBC-MAC
  - Compute retail-MAC (only for 2k-TDES)
  - Verify CBC-MAC
  - Verify retail-MAC (only for 2k-TDES)
  - Padding methods as per ISO/IEC 9797-1 [11] (M1 and M2) and no padding
  - Generic cryptographic operations in chaining mode

1.2.7 ISO/IEC 7816-4 file system

CIPURSE™SAM implements a CIPURSE™ compliant file system based on ISO/IEC 7816-4 [9]:

- Files are organized logically in the form of a two-level dedicated file (DF) tree structure
- The master file (MF) forms the root of this structure. The MF hosts some predefined elementary files (EFs), up to 32 custom EFs, and up to 8 customer-defined application dedicated files (ADFs)
- A CIPURSE™ application is represented by an ADF identified by its file identifier (FID) and DF name application identifier (AID). The ADF may host up to 32 custom EFs for application specific data
- Under each SAM, the following elementary file types are supported:
  - Binary files
  - Linear record files
  - Linear value-record files
  - Cyclic record files
  - Key files
  - Token files
- Security attributes defining the access rights and secure messaging rules may be assigned to the MF, to each ADF and to each EF
- Up to 64 bytes for proprietary security information per MF/ADF

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 with suffix "B" (For example, 0101_B).

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

2 Ordering and packaging information

Package information and ordering codes are defined in Table 1.

Table 1 Ordering information

Type	Package
SLF 9630 – ID1	ID-1/000 chip card with subscriber identity module (SIM) module

2.1 ID-1/000 chip card with SIM module

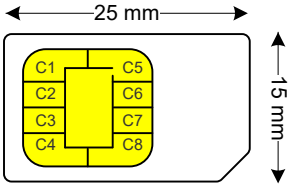

Figure 4 Pin configuration ID-1/000 chip card with SIM module

Table 2 Pin definitions and functions

Card contact	Symbol	Function
C1	V_{CC}	Supply voltage
C2	RST	Control input (reset signal)
C3	CLK	Clock input
C4	-	-
C5	GND	Ground
C6	N.C.	Not connected
C7	I/O	Bi-directional data line
C8	-	-

3 CIPURSE™SAM file system

The file system implemented by the product is compliant with the file system specified in ISO/IEC 7816-4 [9]. As an example, Figure 5 shows the structure of a file system on CIPURSE™SAM representing multiple CIPURSE™SAM applications, a NRG™ SAM application and a generic crypto SAM application.

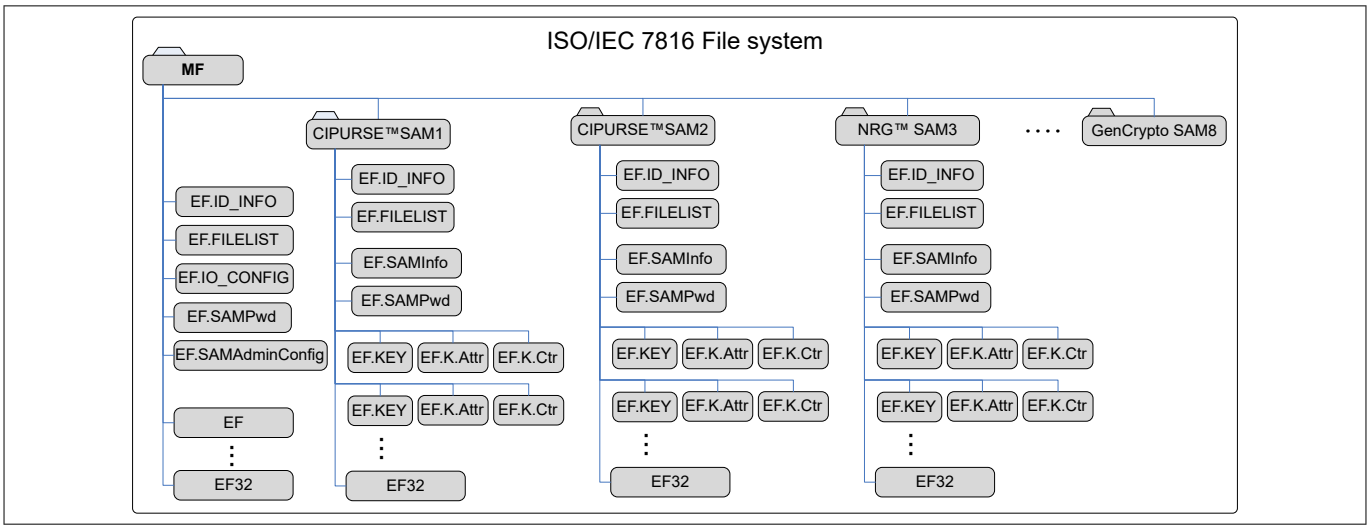

Figure 5 Example for CIPURSE™SAM file system structure

For application operations, the files in the file system are organized logically in form of a two-level DF tree structure. The MF forms the root of the file structure. The MF hosts 3 predefined EFs and up to 8 128-bit AES keys and it allows creation of up to 32 custom EFs and up to 8 custom ADFs.

CIPURSE™SAM application, NRG™ SAM application, and generic crypto SAM application are 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.

3.1 Master file

MF consists of keys, security attributes, and hosts custom ADFs (see Chapter 3.2) in addition to pre-defined EFs (see Chapter 3.5), SAM-specific EFs (see Chapter 3.6), and custom EFs (see Chapter 3.3).

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)
- VERIFY_SAM_PASSWORD

The MF supports transitioning of all SAM application to AUTHORIZED state in the following configurations of EF.SAM_ADMIN_CONFIG (see Chapter 3.6.1).

- On power-up/reset
- On VERIFY_SAM_PASSWORD
- On authentication with a CIPURSE™ key under MF

CTM (see Chapter 3.4.2) will also be applicable for commands manipulating MF attributes including the list of child EFs.

3.2 Application dedicated files

PICC supports six type of ADFs:

- CIPURSE™SAM ADF
- NRG™ SAM ADF
- Generic crypto SAM ADF
- CIPURSE™ ADF
- Proximity system environment (PxSE) ADF
- Near field communication (NFC) Type 4 Tag ADF

3.2.1 CIPURSE™SAM ADF

CIPURSE™SAM ADF hosts SAM-specific files (see Chapter 3.6) and key set elementary files (see Chapter 3.7) in addition to predefined EFs (see Chapter 3.5). CIPURSE™SAM ADF can be created using a standard CIPURSE™ CREATE_FILE command defined in Chapter 5. CIPURSE™SAM ADF supports the additional functionalities described in CIPURSE™ ADF (see Chapter 3.2.4)

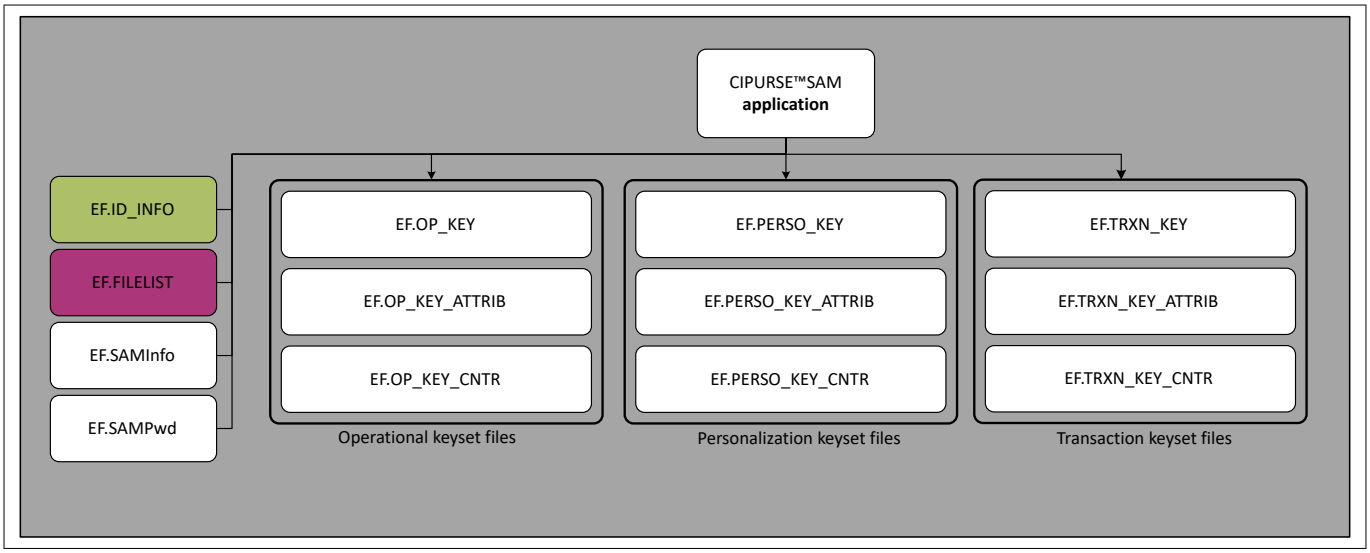

Figure 6 CIPURSE™SAM ADF

3.2.1.1 Command set

This ADF supports commands described in Chapter 5. Additional command set supported by CIPURSE™SAM application is described in this chapter. Table 3 lists the SAM application specific command set supported by this ADF.

Table 3 Command set supported by CIPURSE™SAM ADF

Command	Description
Operation commands	
AUTHENTICATE_SAM	Starts the terminal part of mutual authentication. It calculates the command data for the following MUTUAL_AUTHENTICATE command
AUTHENTICATE_CBP	Completes the terminal part of mutual authentication
END_SESSION	Allows to terminate a session between SAM and CIPURSE™ -based product(s) (CBP)
GENERATE_SM_ELEMENTS	Generates cryptographic relevant elements from the original APDU that are used to form SM_APDU
READ_SESSION_KEY	Allows an external entity like terminal to read out the current session key
VERIFY_SM_ELEMENTS	Decrypts and verifies cryptographic relevant elements of the SM_APDU and provides them in plain text
Personalization commands	
DIVERSIFY_KEYSET	Supports key diversification and personalization of CBP
GENERATE_KEY	Allows to create a new key in a CIPURSE™SAM
LOAD_KEY	Supports loading of keys into the CIPURSE™SAM
Back office admin commands	
PERFORM_SYMCRYPTO	Provides a general MAC and ENC functionality
General commands	
VERIFY_SAM_PASSWORD	Allows to verify the CIPURSE™SAM password
GET_KEY_INFO	Allows to retrieve key information from CIPURSE™SAM

3.2.1.2 State transitions

The CIPURSE™SAM supports two types of security states:

- The CIPURSE™ security state as defined in Chapter 4.2 for any CBP
- The CIPURSE™SAM application specific security states that control the operation of the SAM application specific command set described in this chapter and comprise:
  - Selected – The CIPURSE™SAM ADF is selected
  - Authorized – The CIPURSE™SAM password has been verified successfully
  - PICC_Initiated – The CIPURSE™SAM has responded with a terminal cryptogram to the challenge from the CBP
  - PICC_Authenticated – The CIPURSE™SAM has successfully verified the card cryptogram received from the CBP. The CBP is authenticated and the SAM is ready for secure messaging

Figure 7 shows the CIPURSE™SAM application specific security states and the commands that change the state or are restricted to a certain state.

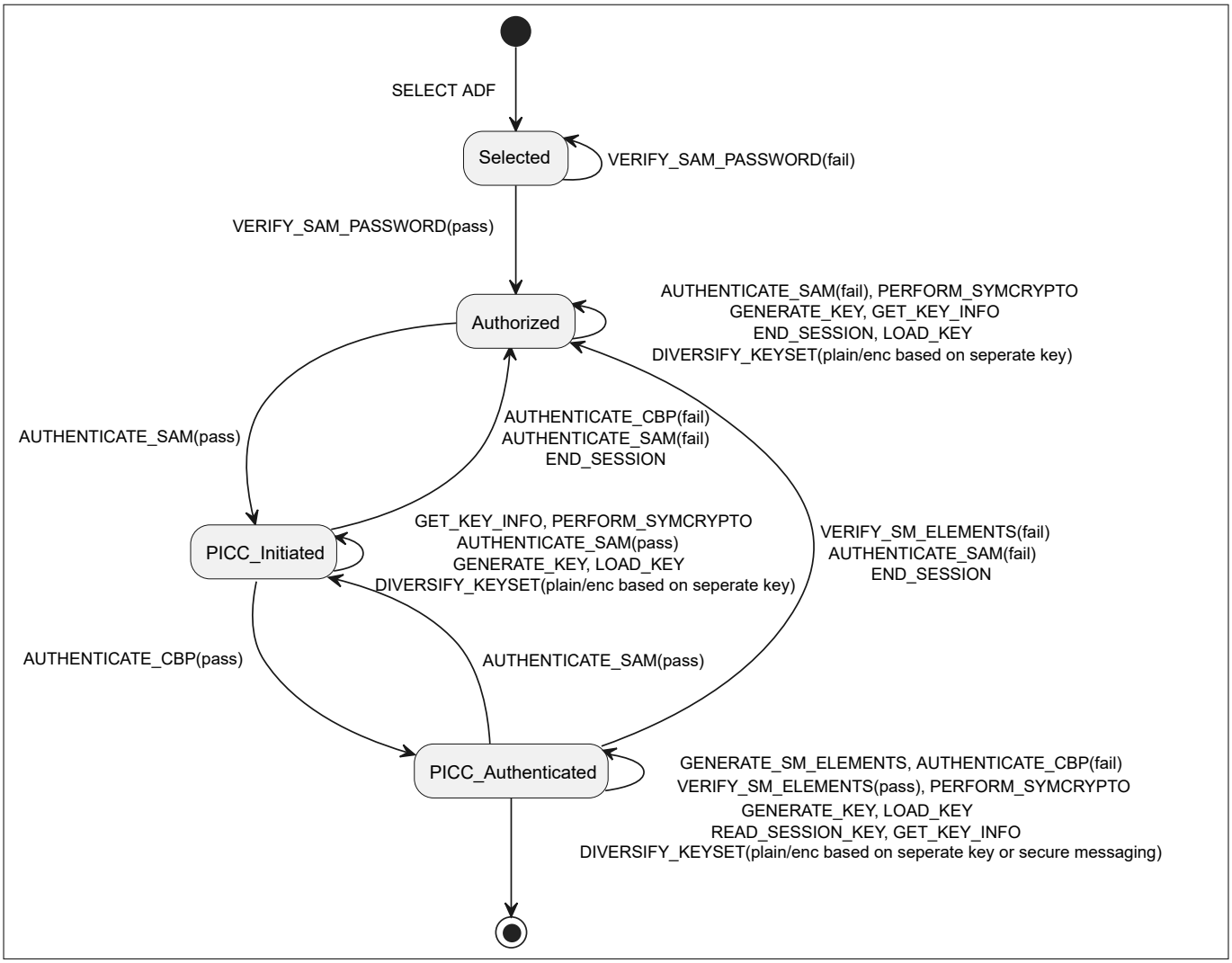

Figure 7 CIPURSE™SAM application specific security states and the commands

3.2.2 NRG™ SAM ADF

NRG™ SAM ADF hosts SAM-specific files (see Chapter 3.6) and key set elementary files (see Chapter 3.7) in addition to predefined EFs (see Chapter 3.5). NRG™ SAM ADF can be created using a standard CIPURSE™ CREATE_FILE command defined in Chapter 5. Structure of the ADF and the functionality of NRG™ SAM are similar to the CIPURSE™SAM ADF except where NRG™ specific functionality mandates deviations.

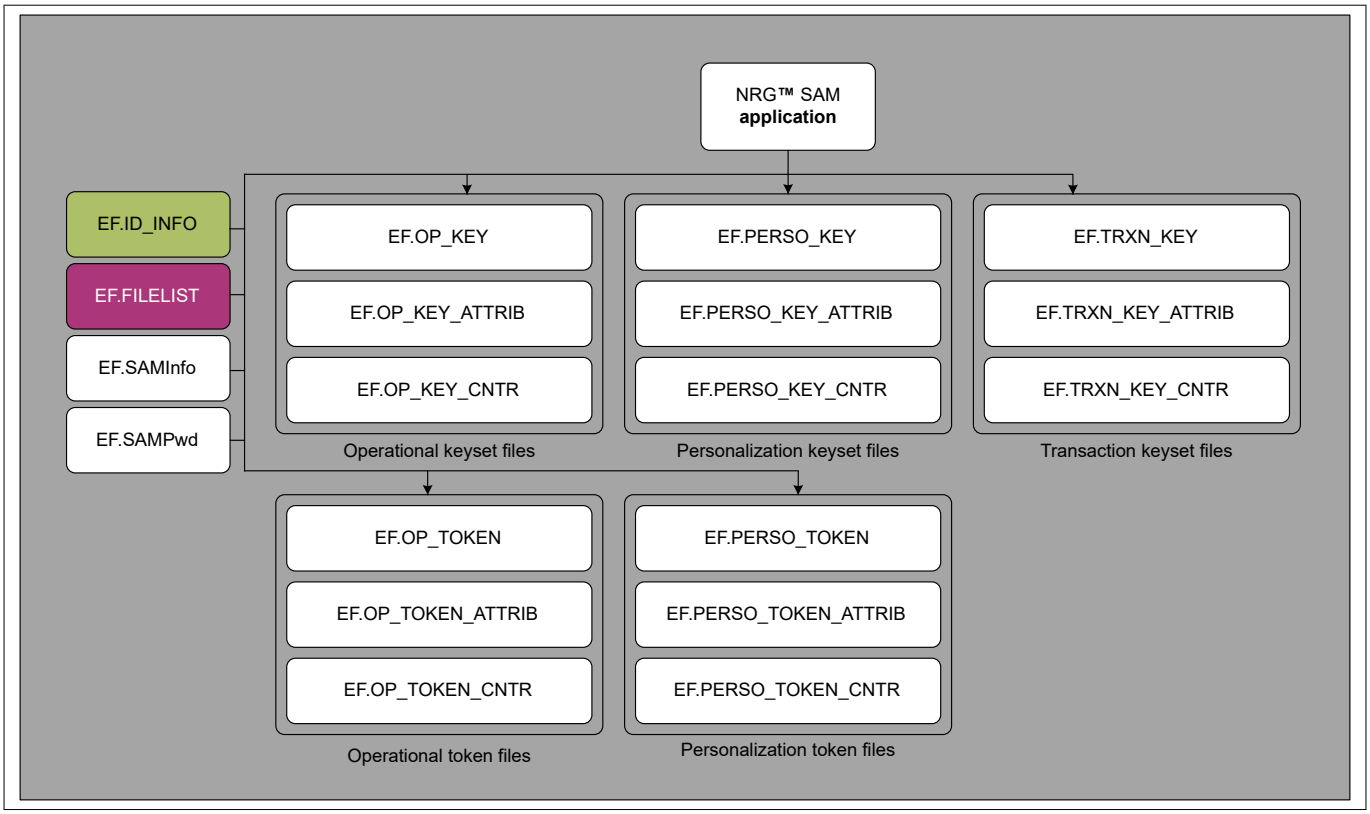

Figure 8 NRG™ SAM ADF

3.2.2.1 Token set elementary files

NRG™ SAM ADF supports token set elementary files in addition to key set elementary files defined in Chapter 3.7.

Token set elementary files (see Table 4) under NRG™ SAM ADF must be created manually during NRG™ SAM ADF personalization using object administration command CREATE_FILE defined in chapter 5 for the proper functioning of NRG™ SAM ADF. SAM application specific commands (see Chapter 3.2.2.2) require presence of these files.

Table 4 Token set elementary files under NRG™ SAM ADF

File	Type	Description
Operational token files	Token file	Contains NRG™ keys that are used for operational commands as described in Table 5 and cannot be read explicitly
Operational token attributes	Linear record	Contains attributes of NRG™ keys residing in operation token files. These files are referenced by operational commands as described in Table 5
Operational token counters	Value record	Contains NRG™ key usage counters. These counters are referred by NRG™ keys. These files are referenced by operational commands as described in Table 5
Personalization token files	Token file	Contains NRG™ keys that are used for personalization commands as described in Table 5 and cannot be read explicitly

(table continues...)

Table 4 (continued) Token set elementary files under NRG™ SAM ADF

File	Type	Description
Personalization token attributes	Linear record	Contains attributes of NRG™ keys residing in the personalization token files. These files are referenced by personalization commands as described in Table 5
Personalization token counters	Value record	Contains NRG™ key usage counters. These counters are referred by NRG™ keys. These files are referenced by personalization commands as described in Table 5

Token files

The token file holds NRG™ keys that can be used for NRG™ product authentication and personalization. These NRG™ keys are either diversified or used as such without diversification during NRG™ product authentication and personalization.

There are two sets of token files – operational and personalization files. Functionality and contents of these two sets are similar.

Token files are populated or updated using SAM commands either GENERATE_KEY or LOAD_KEY as described in 3.2.2.2 respectively. To protect confidentiality of keys stored in these files, reading the contents of these files is not allowed.

Depending on SAM commands issued (see Table 5), one of the sets is used to retrieve the direct NRG™ keys.

Token attribute files

Token attribute files are similar to key attributes files as described in Chapter 3.7.2.

Token counter files

Token counter files are similar to key counters files as described in Chapter 3.7.3.

3.2.2.2 Command set

This ADF supports the command set described in Chapter 5. Additional command set supported by NRG™ SAM ADF is described in this chapter. Table 5 lists the SAM application specific command set supported by this ADF.

Table 5 Command set supported by NRG™ SAM ADF

Command name	Description
Personalization commands	
DIVERSIFY_NRG_KEYSET	Supports key diversification and personalization of NRG™ product
LOAD_KEY	Supports loading of keys into the key and token files
GENERATE_KEY	Allows to create a new key in a key and token file
Operational commands	
AUTHENTICATE_NRGSAM	Starts the terminal part of mutual authentication
AUTHENTICATE_NRG	Completes the terminal part of mutual authentication
ENCRYPT_NRG	Used to perform NRG™ encryption
DECRYPT_NRG	Used to perform NRG™ decryption
END_SESSION	Allows to terminate a session between SAM and NRG™ product

(table continues...)

Table 5 (continued) Command set supported by NRG™ SAM ADF

Command name	Description
General commands	
VERIFY_SAM_PASSWORD	Allows to enable the operation of SAM after device reset
GET_KEY_INFO	Allows to retrieve key information from CIPURSE™SAM
Back office admin (transaction) commands	
PERFORM_SYMCRYPTO	Provides a general MAC and ENC functionality

3.2.2.3 State transitions

In addition to the security state defined in Chapter 4.2, NRG™ SAM application supports specific security states that control the operation of the SAM application specific command set described in this chapter and comprise:

- Selected – The NRG™ SAM ADF is selected
- Authorized – The NRG™ SAM password has been verified successfully
- PICC_Initiated – The NRG™ SAM has responded with a terminal cryptogram to the challenge from the NRG™ product
- PICC_Authenticated – The NRG™ SAM has successfully verified the card cryptogram received from the NRG™ product. The NRG™ product is authenticated and the SAM is ready for secure messaging

Figure 9 shows the NRG™ SAM application specific security states and the commands that change the state or restricted to a certain state

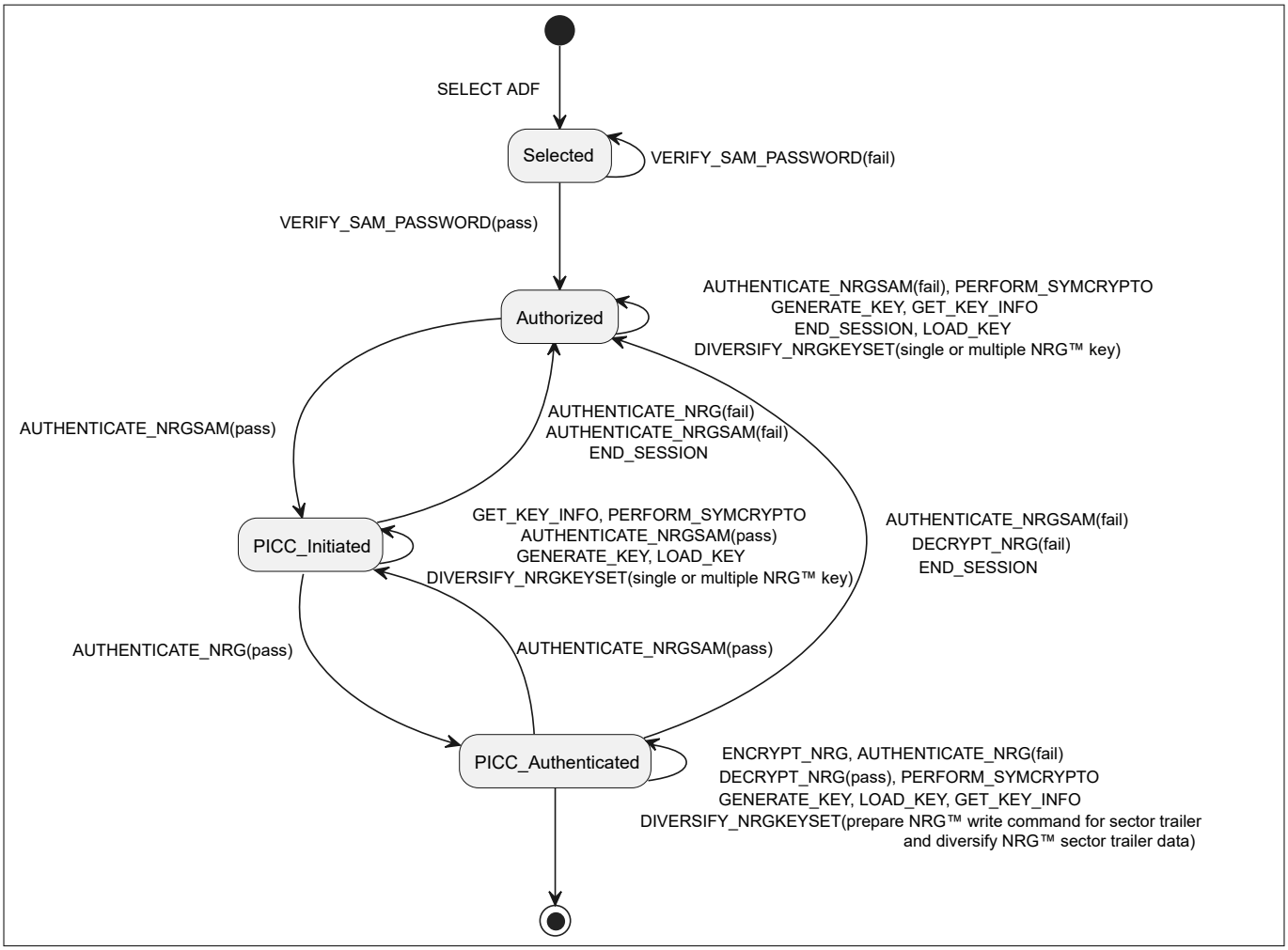

Figure 9 NRG™ SAM application specific security states and the commands

3.2.3 Generic crypto SAM ADF

Generic crypto SAM ADF hosts SAM-specific files (see Chapter 3.6) and key set elementary files (see Chapter 3.7) in addition to predefined EFs (see Chapter 3.5). Generic crypto SAM can be created using a standard CIPURSE™ CREATE_FILE command defined in Chapter 5.

Structure of the ADF and the functionality of generic crypto SAM ADF are similar to the CIPURSE™SAM ADF but the functionality is restricted to key loading and PERFORM_SYM_CRYPTO

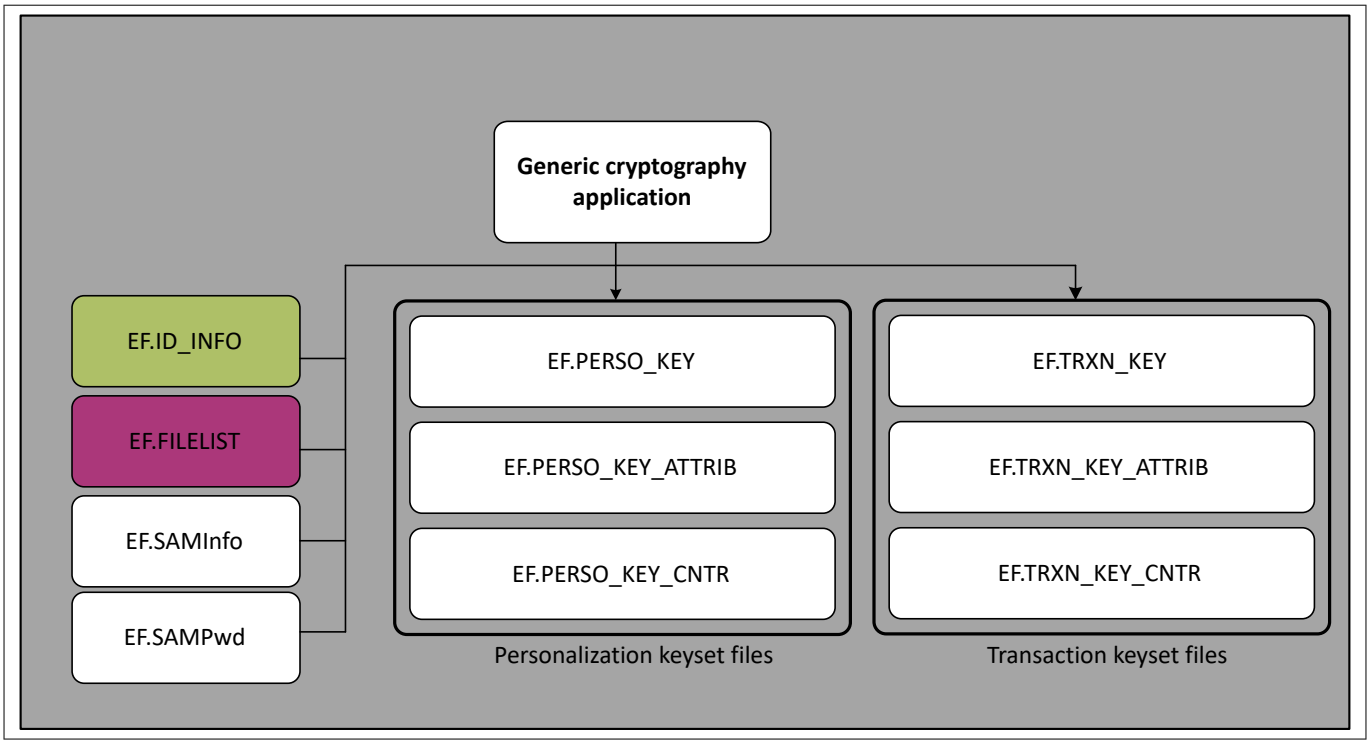

Figure 10 Generic crypto SAM ADF

3.2.3.1 Command set

This ADF supports commands described in Chapter 5. Additional command set supported by generic crypto ADF is described in this chapter. Table 6 lists the SAM specific command set supported by this ADF.

Table 6 Command set supported by generic crypto ADF

Command name	Description
Personalization commands	
LOAD_KEY	Supports loading of keys into the SAM key files
Back office admin (transaction) commands	
PERFORM_SYMCRYPTO	Performs symmetric crypto computations
General commands	
VERIFY_SAM_PASSWORD	Allows to enable the operation of SAM after device reset
GET_KEY_INFO	Allows to retrieve key information from SAM

3.2.3.2 State transitions

In addition to the security state defined in Chapter 4.2, generic crypto SAM application supports specific security states that control the operation of the SAM application specific command set described in this chapter and comprise:

- Selected – The generic crypto SAM ADF is selected
- Authorized – The generic crypto SAM password has been verified successfully

Figure 11 shows the generic crypto SAM application specific security states and the commands that change the state or restricted to a certain state.

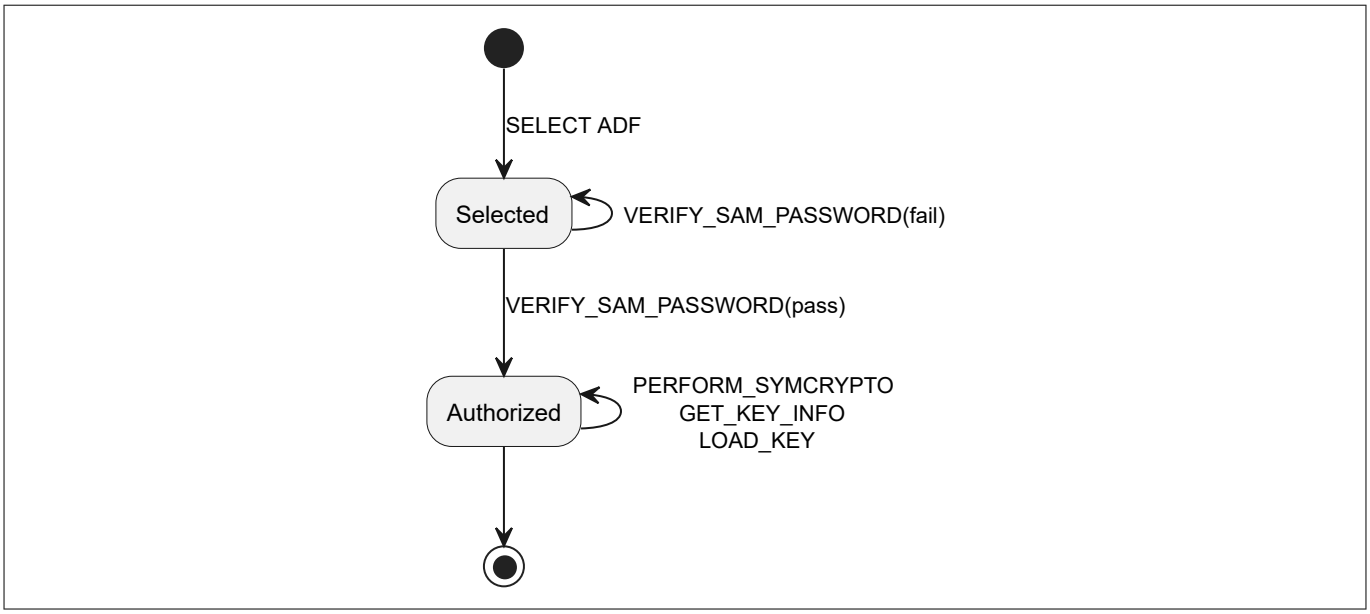

Figure 11 Generic crypto SAM application specific security states and the commands

3.2.4 CIPURSE™ ADF

CIPURSE™ ADF consists of keys and security attributes, and it hosts the EFs with application-specific data as described in Chapter 3.3 in addition to pre-defined EFs (see Chapter 3.5).

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 4.

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 consistent transaction mechanism (CTM) (see Chapter 3.4.2); EF creation, new key values, key attributes, or file attributes become effective after successful execution of PERFORM_TRANSACTION.

3.2.5 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.

3.2.6 NFC Type 4 Tag ADF

The product supports an NFC Type 4 Tag ADF [12] 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.

3.3 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 generic CIPURSE™ 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 4.

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 32768 bytes.

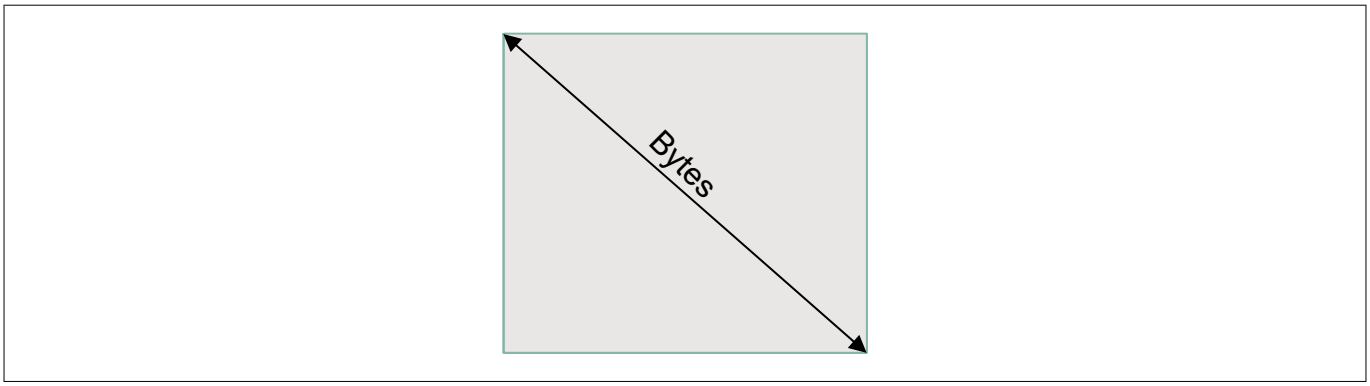

Figure 12 Binary file

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 32767 bytes.

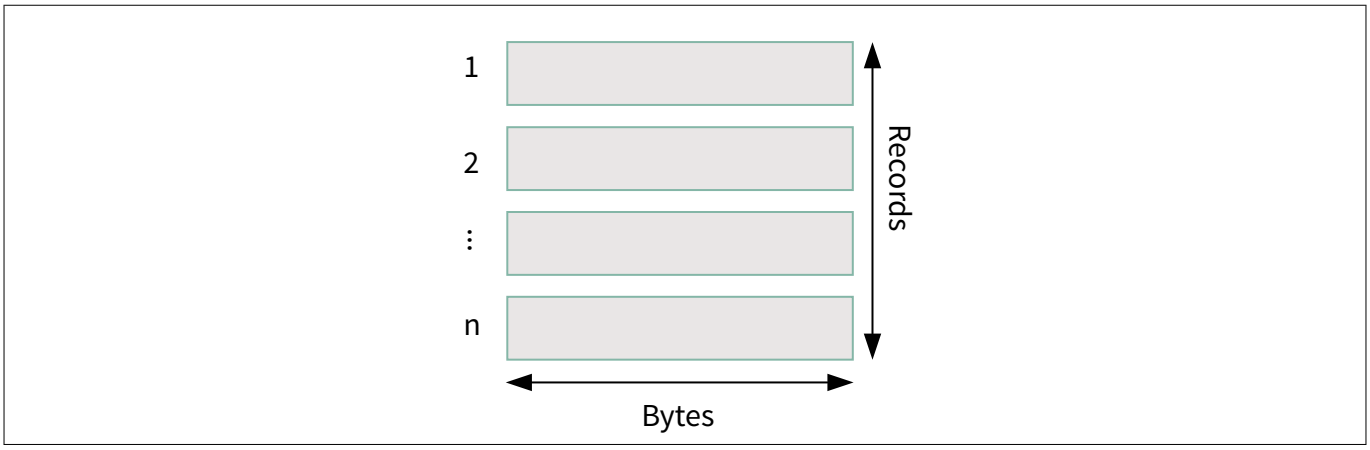

Figure 13 Linear record file

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 32767 bytes.

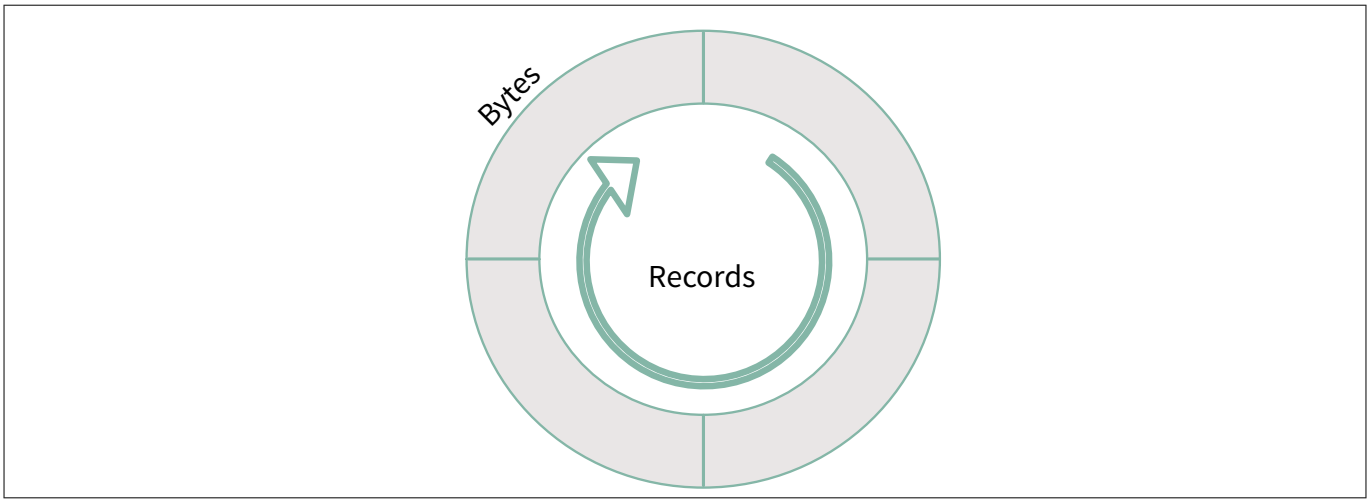

Figure 14 Cyclic record file

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.

The commands LIMITED_INCREASE_VALUE and LIMITED_DECREASE_VALUE can be used to offer a refund functionality that is limited to the number of tokens decreased/increased in last transaction. The value record remembers the last increase or decrease operation and enables refund up to the value that existed before increase or decrease. The commands UPDATE_RECORD, LIMITED_INCREASE_VALUE, and LIMITED_DECREASE_VALUE will reset the information granting limited refund functionality.

A file can contain maximum of 254 records.

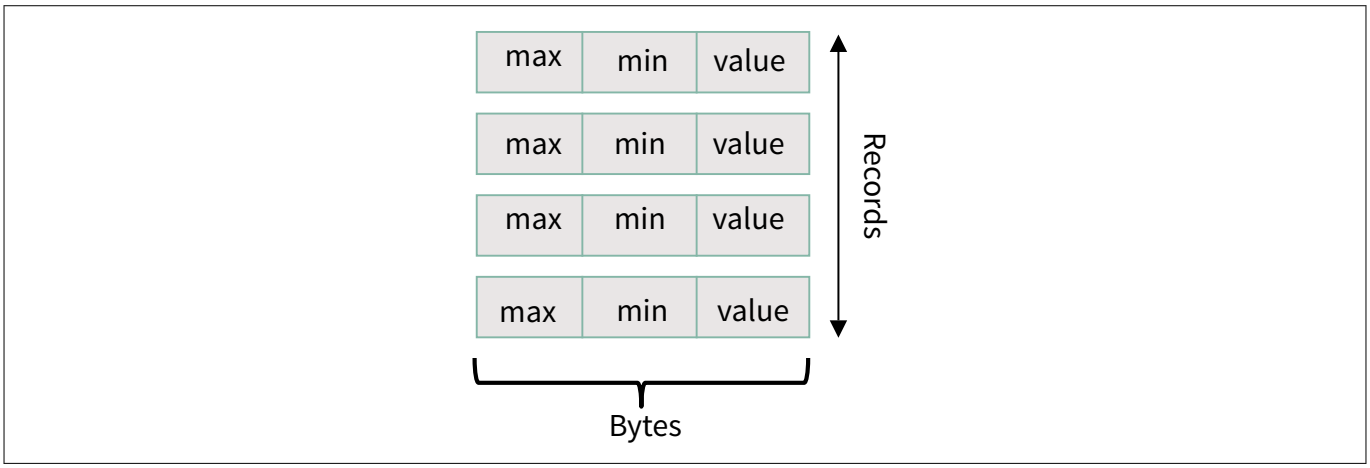

Figure 15 Value-record file

3.4 Consistent data update mechanisms

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

3.4.1 Command level atomicity

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

3.4.2 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.

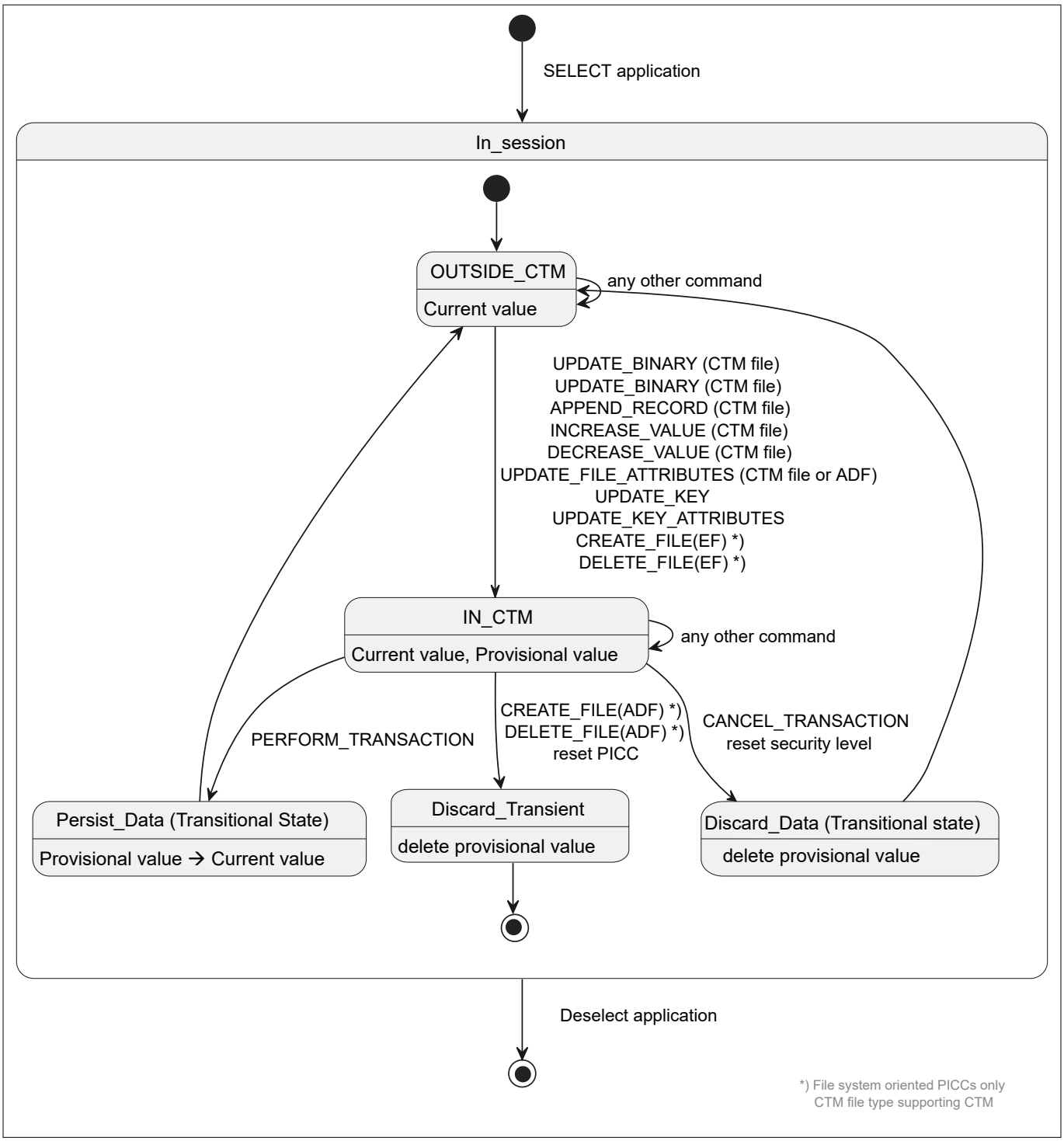

Figure 16 CTM states diagram

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

3.5 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 7 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 answer to reset (ATR) content

3.5.1 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 8 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

(table continues...)

Table 8 (continued) Structure and contents of EF.FILELIST

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

3.5.2 EF.ID_INFO

The predefined file EF.ID_INFO is a read-only file and is available under the MF and each 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 9.

Table 9 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 [10]: • 05_H: Infineon Technologies
9-23	Chip identification data
24-33	Reserved for further manufacturer information
34-36	Software version
37-39	Product identifier

3.5.3 EF.IO_CONFIG

The EF.IO_CONFIG file under the MF describes interface configuration parameters. This file allows the configuration of the interface parameters and the ATR content.

The structure and content of EF.IO_CONFIG file are described in Table 10.

Table 10 Structure and content of EF.IO_CONFIG file

Offset	Description
0-35	Reserved for future use (RFU)
36-47	Configuration data for T=1 communication interfaces (block waiting time index and stop bits)
48-81	Configuration data for ATR specific and historical bytes

3.6 SAM-specific elementary files

SAM-specific elementary files (see Table 11) under MF and each SAM ADF must be created manually during SAM personalization using object administration command CREATE_FILE defined in Chapter 5.

These files should be populated and configured using file data management and file attribute management commands defined in Chapter 5.

Table 11 List of SAM-specific elementary files

SAM-specific EFs	File type	Description
EF.SAMInfo	Binary	Must be manually created under each SAM ADF during personalization. Contains SAM configuration information that controls execution of SAM commands
EF.SAMPwd	Binary	Must be manually created under MF and each SAM ADF during personalization. Contains the password that controls SAM authorization
EF.SAM_CNTR_WARNG	Binary	May be manually created under each SAM ADF during personalization. Contains settings for key counters warning limits
EF.SAM_ADMN_CONFIG	Binary	May be manually created under MF during personalization. Contains global SAM configuration parameters that control the behavior of the product

If the above listed mandatory SAM-specific files are not present under respective SAM ADF, then the SAM application specific commands are not processed.

If the EF.SAM_CNTR_WARNG file is present under SAM ADF, then the warning status for the respective counter is applicable, otherwise it is ignored.

3.6.1 EF.SAM_ADMIN_CONFIG

The EF.SAM_ADMIN_CONFIG file under the MF describes global SAM configuration parameters that govern the behavior of the product.

The structure and content of this file are defined in Table 12.

Table 12 Structure and contents of EF.SAM_ADMIN_CONFIG

Offset	Description
0-1	Tag and length of configuration parameters
2	Configuration of the behavior of transitioning SAM applications to AUTHORIZED state: • On power-up/reset • On VERIFY_SAM_PASSWORD • On authentication with a CIPURSE™ key under MF
3	MF key number used in CIPURSE™ authentication to transition all SAM applications to AUTHORIZED state
4	Enable or disable the plain AES key support in AUTHENTICATE_SAM command, weak key check for Data Encryption Standard (DES) keys, and key usage byte validation for PERFORM_SYMCRYPTO
5	Enable or disable crypto algorithms and modes that are supported by PERFORM_SYMCRYPTO
6-10	RFU. Should be set to zeros

3.6.2 EF.SAMInfo

The EF.SAMInfo under SAM ADF provides SAM ADF identifier information and defines SAM ADF behavior during processing of SAM application specific commands (see Table 3, Table 5, and Table 6).

The "SAM use" byte in the EF.SAMInfo file defines the functionality that the CIPURSE™SAM supports.

The SAM ADF can be configured by updating the "SAM use" byte in the EF.SAMInfo file for one of the following uses:

- PERSO SAM: personalization – supports personalization of CIPURSE™ or NRG™ products
- STANDARD SAM: standard end-user product – supports standard functions in a terminal to support CIPURSE™ or NRG™ product applications
- LOAD SAM: key loading – supports functions to load keys onto CIPURSE™SAMs
- BACK SAM: back office admin – supports functions to verify and decrypt transaction messages
- GENERAL SAM: no restriction on operation. This allows for simple schemes to be easily configured

As the EF.SAMInfo contents are used during processing of SAM application specific commands, the EF.SAMInfo is required to be created and populated within the ADF to allow execution of the of SAM application specific commands.

3.6.3 EF.SAMPwd

The EF.SAMPwd file under the MF/SAM ADF holds the password to authorize SAM at MF/ADF level. The password issued in VERIFY_SAM_PASSWORD command is verified against the password residing in this file. When this file is present at MF level, issuing VERIFY_SAM_PASSWORD command at the MF level transitions all SAM applications residing under MF to the AUTHORIZED state, depending on the configuration of EF.SAM_ADMIN_CONFIG file (see Chapter 3.6.1).

Contents of this binary file are:

- Password
- Current retry counter value
- Maximum retry counter value

Verification of SAM password is implemented to withstand simple power analysis (SPA) attacks.

3.6.4 EF.SAM_CNTR_WARNG

The EF.SAM_CNTR_WARNG under SAM ADF is an CIPURSE™SAM specific configuration file that holds configuration information to set warning threshold for key counter usage. During SAM use, if this file is present and key counter falls below a threshold value set in respective field, then warning status word is issued in the response of the command that is using the respective key.

This warning status word implies the command itself is successful and indicates that key counter is below threshold value.

The warning threshold values are supported for the following key counters:

- Personalization key counters
- Operational key counters
- Transaction key counters
- Personalization token counters
- Operational token counters

3.7 Key set elementary files

Key set elementary files (see Table 13) under each SAM ADF must be created manually during SAM ADF personalization using object administration command CREATE_FILE defined in Chapter 5 for the proper functioning of CIPURSE™SAM. SAM application specific commands (see Table 3, Table 5, and Table 6) require presence of these files.

Table 13 Key set elementary files under SAM ADF

File	Type	Description
Operational key files	Key file	Contains SAM keys that are used for operational commands and cannot be read explicitly
Operational key attributes	Linear record	Contains attributes of operational keys defined in the corresponding key file. These files are referenced by operational commands
Operational key counters	Value record	Contains key usage counters. These counters are referred by operational keys. These files are referenced by operational commands
Personalization key files	Key file	Contains SAM keys that are used for personalization commands as and cannot be read explicitly
Personalization key attributes	Linear record	Contains attributes of personalization keys defined in the corresponding key file. These files are referenced by personalization commands
Personalization key counters	Value record	Contains key usage counters. These counters are referred by personalization keys These files are referenced by personalization commands
Transaction key files	Key file	Contains SAM keys that are used for transaction commands and cannot be read explicitly
Transaction key attributes	Linear record	Contains attributes of transaction keys defined in the corresponding key file. These files are referenced by back office admin commands
Transaction key counters	Value record	Contains key usage counters. These counters are referred by transaction keys. These files are referenced by back office admin commands

The commands which are involving key file functionality are described in respective SAM application specific commands (Table 3, Table 5, and Table 6):

- Operational key files, their attributes, and counters are referenced by operational commands
- Personalization key files, their attributes, and counters are referenced by personalization commands
- Transaction key files, their attributes, and counters are referenced by transaction commands

3.7.1 Key files

Key files hold keys. Keys are either diversified or used as such without diversification during personalization of PICC and operational phase of PICC.

There are three sets of key files – operational, personalization, and transaction key files. Functionality and contents of these three sets are similar.

Key files are populated or updated using SAM commands either GENERATE_KEY or LOAD_KEY. To protect confidentiality of keys stored in these files, reading the contents of these files is not allowed.

The key file is able to store the various key types that are listed below:

- DES Key
- 2k-TDES key
- 3k-TDES key
- AES-128 key

- AES-192 key
- AES-256 key

3.7.2 Key attribute files

Key attributes file is a linear record file that holds key attributes. Key attributes define the behavior of the keys residing in key files with the help of "key use" byte.

There are three sets of key attributes files – operational, personalization, and transaction key attributes files. Functionality and contents of these three sets are identical.

These files should be populated and configured using file data management and file attribute management commands defined in Chapter 5.

When a key is referenced to be used for an operation by a CIPURSE™SAM command, the key use byte is checked to ensure that the key can be used for a particular operation (key export, diversification, and encryption).

To execute SAM commands that access keys from a key file, it is mandatory for every key file, there must be a corresponding key attributes file and for every key in key file, there must be a corresponding key attributes record in key attributes file. There is a one-to-one mapping between keys in key files and key attributes records in key attributes files.

3.7.3 Key counter files

Key counters file is a linear value-record file[1)] that holds key usage counters. On every key usage, the counter associated with this key is decremented. Once the counter reaches its minimum value, the key cannot be used any further.

There may be many-to-one mapping between keys in key files and counter in key counter files.

There are three sets of key counters file – operational, personalization, and transaction key files. Functionality and contents of these three sets are similar.

These files should be populated and configured using file data management and file attribute management commands defined in Chapter 5.

3.8 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

1 Must be of a version not supporting CTM

3.9 Reserved file identifiers

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

For example, FIDs 60XX – 62XX are reserved for operational key set and are referenced by operational commands only.

4 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.

4.1 Keys

There are two different sets of keys residing in SAM.

- One set of keys (AES-128 bit) is associated with MF and ADF for authentication
  - Each key has a set of secure and non-secure key 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 immutable and if the key is valid or invalid
    - Non-secure key attributes hold an additional key information and cryptographic algorithm identifier
- The second set of keys is the keys residing in key files. Key management and usage functionality of these keys are defined in the SAM application specific command set defined in Table 3, Table 5, and Table 6

4.2 Mutual authentication and security state

Figure 17 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.

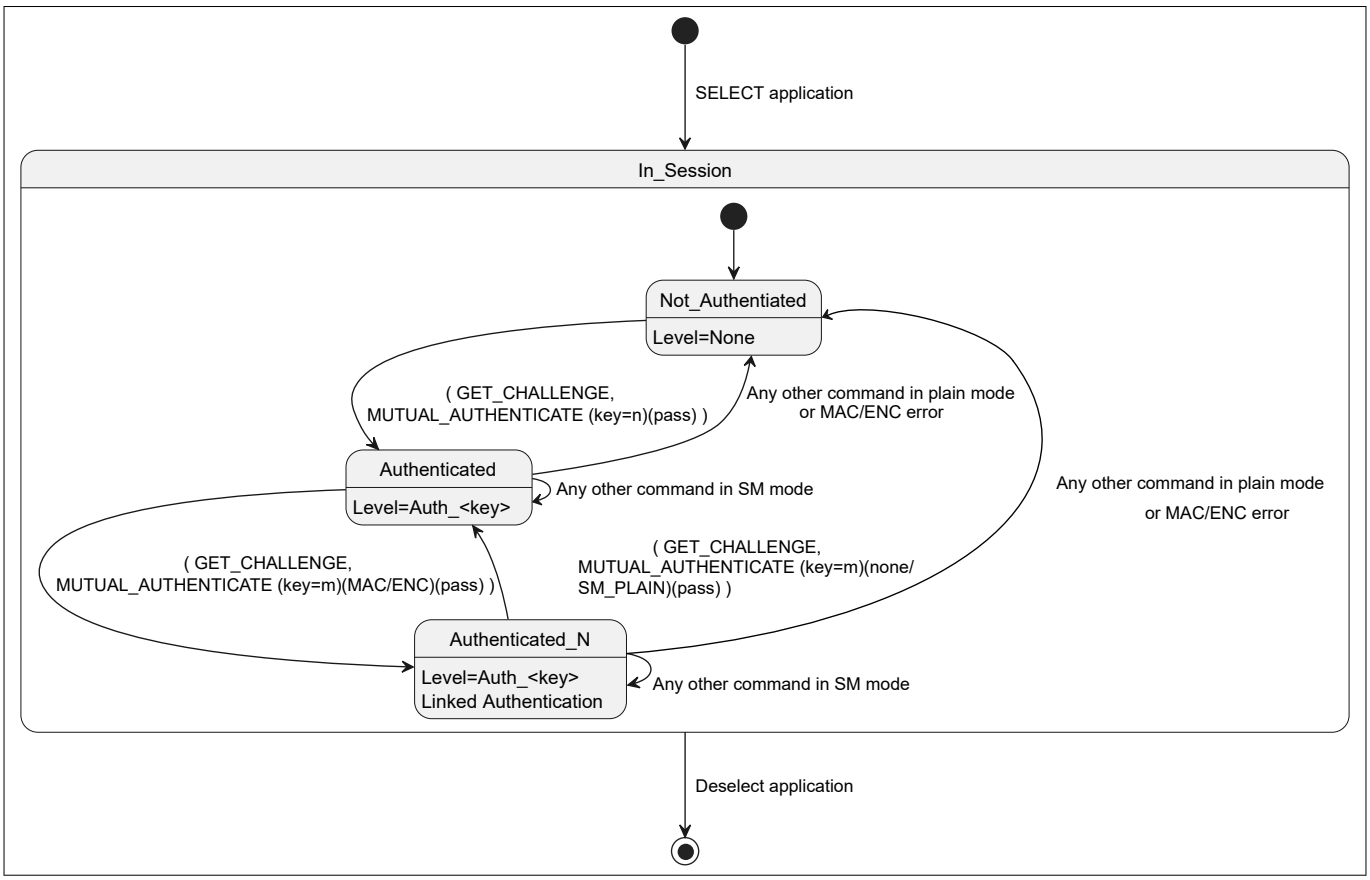

Figure 17 Authentication states and security level

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 4.3.

4.3 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 proximity coupling devices (PCDs) to execute commands irrespective of the security level reached and the secure messaging rules assigned to the file, see Chapter 4.4.

Except for the commands GENERATE_KEY and LOAD_KEY, all other SAM application specific commands defined in Table 3, Table 5, and Table 6 are not administrated by security level and secure messaging rules. Access rights to execute these commands are granted based on the SAM application specific security state as defined in Chapter 3.2.1.2, Chapter 3.2.2.3, and Chapter 3.2.3.2.

4.4 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.

5 Command set

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

Table 14 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
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
LIMITED_INCREASE_VALUE	Increases the value in a value record file within a limited range defined by the previous DECREASE_VALUE operation
LIMITED_DECREASE_VALUE	Decreases the value in a value record file by a limited amount
PERFORM_TRANSACTION	Finalizes a transaction that is in progress
CANCEL_TRANSACTION	Cancels a transaction that is in progress

6 Delivery image

The CIPURSE™SAM product is delivered with default delivery image.

The default delivery image comes with a default file structure. The file structure serves the purpose of general SAM use case for CBP and NRG™ products. It also comes with access condition set to unconditional ("ALWAYS"). The file structure is shown in Figure 18.

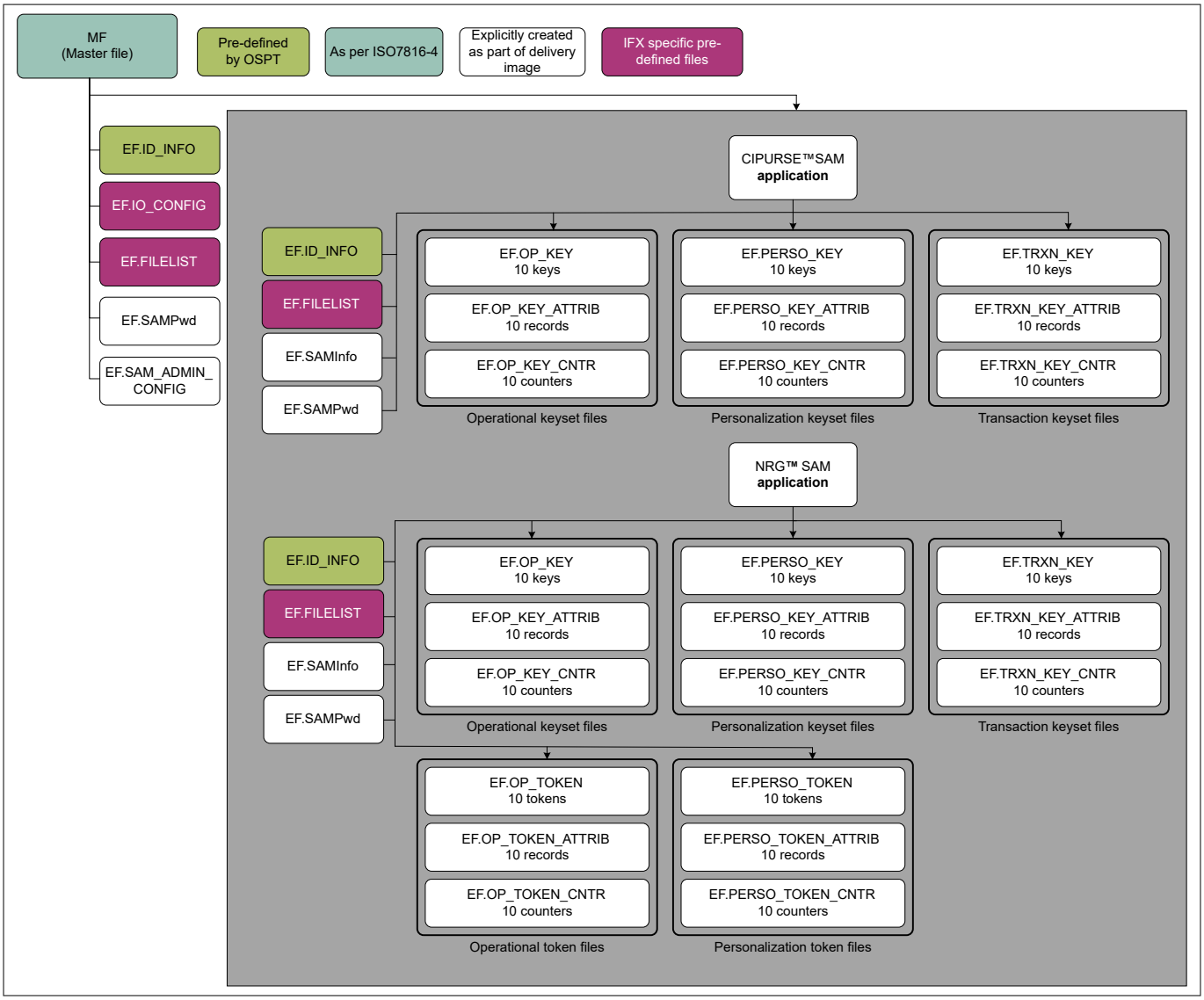

Figure 18 Default delivery image for CIPURSE™SAM product

Note: The FORMAT_ALL command at delivery state moves the card to empty state, which contains MF and predefined EFs under MF.

7 Operational characteristics

7.1 Absolute maximum ratings

Stresses above the values listed in Table 15 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 whose value exceeds 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 15 Absolute maximum ratings

Parameter	Symbol	Limit values			Unit	Note/test condition
		Min	Typ	Max		
Junction Temperature	TJ	-40	-	110	°C	-
Storage Temperature	T_S	-40	-	125	°C	-
Supply Voltage	V_{CC}	-0.3	-	7.0	V	-
ESD protection	VESD_ISO, HBM	-	-	4000	V	JESD22-A114C [3]
	VESD_ISO, CDM	-	-	500	V	JESD22-C101C [4]

Note: For further information on Table 15, please refer to your Infineon Technologies office or representative.

7.2 Electrical characteristics

Data retention is for minimum of 25 years at 25°C in non-volatile memory (NVM) cells that were never previously programmed. The product supports:

- At least 30 million updates on the key counter files
- At least 100k update operations for all other EFs

Table 16 Operation range

Parameter	Symbol	Limit values			Unit	Note/test condition
		Min	Typ	Max		
Ambient temperature	T_A	-40	-	85	°C	T_J must be kept

Over recommended operational temperature range.

Table 17 ISO/IEC 7816-3 card DC electrical characteristics

Parameter	Symbol	Values			Unit	Note/test condition
		Min	Typ	Max		
Supply voltage	V_{CC}	1.62	-	5.5	V	-

(table continues...)

Table 17 (continued) ISO/IEC 7816-3 card DC electrical characteristics

Parameter	Symbol	Values			Unit	Note/test condition
		Min	Typ	Max		
Pad input voltage [1)]	V_I	-0.3	-	V_{CC} + 0.3	V	-
Supply current	I_{CCAVG}	-	25	-	mA	-
RST						
Input high voltage	V_{IH}	0.7 * V_{CC}	-	V_{CC}	V	-
Input low voltage	V_{IL}	0	-	0.2 * V_{CC}	V	-
CLK						
Input high voltage	V_{IH}	0.7 * V_{CC}	-	V_{CC}	V	-
Input low voltage	V_{IL}	0	-	0.2 * V_{CC}	V	-
I/O						
Input high voltage	V_{IH}	0.7 * V_{CC}	-	V_{CC}	V	-
Input low voltage	V_{IL}	0	-	0.2 * V_{CC}	V	-
Output high voltage	V_{OH}	0.7 * V_{CC}	-	V_{CC}	V	I_{OH_max} = +20 µA, 20 kΩ to V_{CC}
Output low voltage	V_{OL}	0	-	0.15 * V_{CC}	V	I_{OL_max} = -1 mA

1) ISO/IEC 7816-3 card maximum rating

Table 18 ISO/IEC 7816-3 card AC electrical characteristics

Parameter	Symbol	Values			Unit	Note/test condition
		Min	Typ	Max		
V_{CC} rampup time	t_{R_VCC}	1	-	10^7	µs	0 to 100% of target voltage
RST						
Rise/fall time	t_R, t_F	-	-	400	µs	-
Input load capacitance	C_{LOAD}	-	-	30	pF	-
CLK						
External frequency	f_{UART_CLK}	1	-	10	MHz	@ duty cycle 40% to 60%

(table continues...)

Table 18 (continued) ISO/IEC 7816-3 card AC electrical characteristics

Parameter	Symbol	Values			Unit	Note/test condition
		Min	Typ	Max		
Rise/fall time	t_R, t_F	-	-	0.09 * (1/ f_{UART_CLK})	ns	Measured between 10% and 90% of signal amplitude
Input load capacitance	C_{LOAD}	-	-	30	pF	-
I/O						
Rise/fall time	t_R, t_F	-	-	1	µs	-
Input load capacitance	C_{LOAD}	-	-	30	pF	-

References

CIPURSE™/OSPT

[1] OSPT Alliance: *CIPURSE™V2 SAM Specification (Revision 1.0), 2013-10-14, incl. Errata and Precision List (Revision 1.0)*; 2015-02-06

[2] 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

JEDEC

[3] JEDEC JESD22-A114C: *Electrostatic Discharge (ESD) Sensitivity Testing Human Body Model (HBM)*; 2006

[4] JEDEC JESD22-C101C: *Field-Induced Charged-Device Model Test Method for Electrostatic-Discharge-Withstand Thresholds of Microelectronic Components*; 2004

Infineon

[5] Infineon Technologies AG: *CIPURSE™4move Datasheet (latest revision)*

[6] Infineon Technologies AG: *CIPURSE™move Datasheet (latest revision)*

[7] Infineon Technologies AG: *CIPURSE™Security Controller Datasheet (latest revision)*

ISO/IEC

[8] ISO/IEC 7816-3:2006: *Identification cards - Integrated circuit cards - Part 3: Cards with contacts - Electrical interface and transmission protocols (Third edition)*; 2006-11

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

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

[11] ISO/IEC 9797-1:2011: *Security techniques - Message Authentication Codes (MACs) - Part 1: Mechanisms using a block cipher (Second edition)*; 2011-03; www.iso.org

NFC Forum

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

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).

AFC

automatic fare collection (AFC)

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.

ATR

answer to reset (ATR)
A message conforming to ISO/IEC 7816 sent by the controller following a reset. It contains information on communication parameters, type and state of the chip.

CBC

cipher block chaining (CBC)

CC

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

CBP

CIPURSE™-based product(s) (CBP)

CDM

charged device model (CDM)

CLK

clock (CLK)

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)

DES

Data Encryption Standard (DES)
The standard referring to a symmetric-key algorithm for the encryption of electronic data.

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)

ECB

electronic code book (ECB)

EEPROM

electrically erasable programmable read-only memory (EEPROM)

EF

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

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.

FID

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

HBM

human body model (HBM)

IEC

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

I/O
input/output (I/O)

ISO
International Organization for Standardization (ISO)

JEDEC
Joint Electron Device Engineering Council (JEDEC)

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

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

NFC
near field communication (NFC)

NRG™
ISO/IEC 14443-3 type A with CRYPTO1

NVM
non-volatile memory (NVM)

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.

RF
radio frequency (RF)

RFU
reserved for future use (RFU)

RST
reset (RST)

SAM
secure access module (SAM)
A module based on smart card integrated circuits, and used to enhance the security and cryptography performance in devices. It is commonly used in smart card readers that need to perform secure transactions, for example, payment or ticketing terminals. The module is also referred to as a secure application module.

SDES
single DES (SDES)

SFID
short file identifier (SFID)

SIM
subscriber identity module (SIM)

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.

SPA
simple power analysis (SPA)

TDES
Triple DES (TDES)

UART
universal asynchronous receiver/transmitter (UART)
A universal asynchronous receiver transmitter is used for serial communications over a peripheral device serial port by translating data between parallel and serial forms.

Revision history

Reference	Description
Revison 1.0, 2023-01-05 - Valid for product version V1.2.3 or higher	
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-mvf1662527237878

Important notice

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.