MF3E(c)x3

Security Target Lite

Rev. 1.0 — 9 September 2024

NSCIB-2300018-01

Evaluation document

Document information

Information	Content
Keywords	Common Criteria, Security Target Lite, MF3E(c)x3
Abstract	Evaluation of the MF3E(c)x3 developed and provided by NXP Semiconductors, Business Line Secure Connected Edge (SCE), according to the Common Criteria for Information Technology Evaluation Version 3.1 R5 at EAL6 augmented.

Revision History

Rev.	Date	Description
1.0	2024-09-09	First release, based on full Security Target v1.3

1 Introduction

1.1 ST Reference

Table 1. ST Reference

Title	MF3E(c)x3 Security Target Lite
Revision	1.0
Date	9 September 2024

1.2 TOE Reference

Table 2. TOE Reference

TOE Name	MF3E(c)x3
IC Hardware	B0
IC Dedicated Software	Firmware: 3.0.11, Crypto Library: 2.4.2
Operating System	3.0.1
Hardware Major/Minor Version	0xA0 0x00
Software Major/Minor Version	0x00 0x01

Note: *The Hardware and Software Major/Minor Version identifiers listed in the table above represent the IC Hardware, IC Dedicated Software and Operating System version. The corresponding byte values can be checked on the TOE by the customer using the "GetVersion" APDU command as described in [8]. The complete certified response from this command is also documented in [8].*

1.3 TOE Overview

NXP has developed the MF3E(c)x3 to be used with Proximity Coupling Devices (PCDs, also called "terminal") according to ISO 14443 Type A. The communication protocol complies to part ISO 14443-4. Alternatively, in specific configurations the MF3E(c)x3 can be used with a host MCU through the I2C interface. The MF3E(c)x3 is primarily designed for secure contactless transport applications and related loyalty programs as well as access control management systems as well as closed loop payment systems. It fully complies with the requirements for fast and highly secure data transmission, flexible memory organisation and interoperability with existing infrastructure.

The TOE is a smart card IC comprising a hardware platform and a fixed software package. The software package is stored in ROM memory and provides an operating system with a set of functions, used to manage the various kinds of data files stored in Flash memory. The operating system supports a separation between the data of different applications and provides access control if required by the configuration.

The TOE includes also IC Dedicated Software to support its start-up and for test purposes after production. The Smart Card Controller hardware comprises a 32-bit CPU, volatile and non-volatile memories, cryptographic co-processors, security components and two communication interfaces.

The TOE includes a functional specification and a guidance document. This documentation contains a description of the hardware and software interface, the secure configuration and usage of the product by the terminal designer.

The security measures of the TOE are designed to act as an integral part of the combination of hardware platform and software package in order to strengthen the product as a whole. Several security measures are completely implemented in and controlled by the hardware. Other security measures are controlled by the combination of hardware and software.

1.3.1 Required non-TOE Hardware/Software/Firmware

The TOE requires an ISO 14443 card terminal to be provided with power and to receive adequate commands. Alternatively, when the TOE is configured to use the I2C interface, a host MCU is required to communicate with the device.

1.4 TOE Description

1.4.1 Physical Scope of the TOE

The Target of Evaluation (TOE) is the smart card IC named MF3E(c)x3 in combination with a fixed software package, the IC Dedicated Software. The TOE includes IC manufacturer proprietary IC Dedicated Test Software and IC Dedicated Support Software, according to the terminology used in the Security IC Protection Profile [6]. The TOE deliverables are mentioned in the table below.

Table 3. TOE deliverables

Type	Name	Version	Form of delivery
IC Hardware	MF3E(c)x3 Hardware	B0	Sawn wafer, packages
IC Dedicated Test Software	Test Software	3.0.11	On-chip software
IC Dedicated Support Software	Boot Software	3.0.11	On-chip software
	Firmware	3.0.11	On-chip software
	Crypto Library	2.4.2	On-chip software
	Operating System	3.0.1	On-chip software
Document	MF3E(H)x3, Product data sheet [8]	1.0	Electronic document (PDF via NXP DocStore)
Document	MF3E(H)x3 PDC, Data sheet addendum [9]	0.1	Electronic document (PDF via NXP DocStore)
Document	MF3E(c)x3, Wafer and Delivery Specification, Data sheet addendum [11]	1.2	Electronic document (PDF via NXP DocStore)
Document	MF3E(c)x3, User Guidance Manual [10]	1.2	Electronic document (PDF via NXP DocStore)

1.4.1.1 Evaluated Configurations

The TOE is available in various configurations. Each configuration has a different commerical type name. A commercial type name for the TOE has the following general format:

- MF3E(c)xeywdpp(p)/sv(ff)

The following table illustrates the commerical type names that are subject of the evaluation:

Table 4. Variable definitions for commercial type names

Identifier	Description	Assignment	Meaning
MF3E	product identifier	fixed	MF3E(c)x3
(c)	input capacitance	<omitted> A...Z	17 pF input capacitance other input capacitance (i.e. 50 pF, 70 pF, and others)
x	memory size	0...9, A...Z	memory size identifier, allowing up to 16 kB of memory
e	evolution	3	product evolution with additional ECC functionality

Table 4. Variable definitions for commercial type names*...continued*

Identifier	Description	Assignment	Meaning
y	UID size / market	0...9, A...Z	UID size allocation for different market segments (i.e., consumer market, automotive market, etc.)
w	wafer fab code	0...9	fab identification indicating the wafer fab production site
d	fixed value	D	
pp(p)	package type	AA...ZZ	package type option (i.e. MOA8 module, 120µm wafer with bumps, 75µm wafer with bumps, WLCSP package, and others)
/	separator		
s	SW minor version (higher nibble)	0	SW minor version information
v	SW minor version (lower nibble)	0	SW minor version information
(ff)	Type ID in GetVersion Frame 3	A...Z, 0...9 <omitted>	customer specific type identifier default type without customer data

All commercial type names are subject to this evaluation. However the identifier "MF3E(c)x3" will be used in the remainder of this document to make referencing easier. All information and security functionality described in this Security Target applies to all commercial types.

1.4.2 Logical Scope of the TOE

1.4.2.1 Hardware Description

The CPU of the MF3E(c)x3 has an 32-bit architecture. The on-chip hardware components are controlled by the software via Special Function Registers. These registers are correlated to the activities of the CPU, the memory protection unit, interrupt control, contactless communication, Flash, timers, the AES co-processor and the ECC co-processor. The communication with the MF3E(c)x3 can be performed through the contactless interface or in specific configurations using the I2C interface.

The ECC co-processor supports ECC operations with a key length of 256 bit over the NIST P-256 and brainpoolP256r1 curves. The AES co-processor supports AES operations with a key length of 128 and 256 bit.

A hardware Random Number Generator provides true random numbers which are used to seed deterministic random number generators, used internally by the operating system for security purposes.

1.4.2.2 Software Description

The IC Dedicated Test Software (Test ROM Software) located in ROM of the TOE is used by the TOE Manufacturer to test the functionality of the chip. The test functionality is disabled before the operational use of the smart card. The IC Dedicated Test Software includes the test operating system, test routines for the various blocks of the circuitry and shutdown functions to ensure that security relevant test operations cannot be executed illegally after phase 3 of the TOE Life cycle.

The TOE also contains IC Dedicated Support Software. The Boot Software which is stored in ROM is part of the IC Dedicated Support Software. This software is executed after each reset of the TOE, i.e. every time when the TOE starts. It sets up the TOE and does some basic configuration. The operating system is also part of the IC Dedicated Software and provides the main functionality of the TOE in the usage phase. The MF3E(c)x3 is primarily designed for secure contactless transport applications and related loyalty programs as well as access control systems. It fully complies with the requirements for fast and highly secure data transmission, flexible memory organization and interoperability with existing infrastructure. Its functionality consists of:

- Flexible file system that groups user data into applications and files within each application.
- Support for different file types like Data files, Value files, Record files, including definition of multiple file access conditions per file.
- ECC-based Mutual and Reader-Unilateral Authentication.
- ECC-based Card-Unilateral Authentication and generic ECDSA support.
- AES-based Mutual Authentication and Secure Messaging (EV2 Secure Messaging).
- Authentication on application level with fine-grained access conditions for files.
- Multi-application support that allows distributed management of applications and ensures application segregation.
- Delegated-application support that allows third party service providers to create their applications onto the issued TOE.
- Multiple application selection that allows transaction over files in two applications.
- Data encryption on the communication path.
- Message Authentication Codes (MAC) for replay attack protection.
- Flexible key management (for symmetric and asymmetric keys) on PICC and application level.
- ECC keypair generation.
- Transaction system with rollback that ensures consistency for complex transactions.
- Unique serial number for each device (UID) with optional random UID.
- Key set rolling feature per application to switch to a predefined symmetric key set.
- Transaction MAC feature (via AES-based CMAC or ECDSA signature) to prevent fraudulent merchant attacks.
- ECC-based originality functionality that allows verifying the authenticity of the TOE.
- Proximity check feature for protection against relay attacks on the TOE.
- Secure Dynamic Messaging feature which allows confidential (via AES-based encryption) and integrity protected data (via AES-based CMAC or ECDSA signature) exchange without requiring a preceding authentication.

Asymmetric cryptography features support 256-bit ECC over the NIST P-256 and brainpoolP256r1 curves. Symmetric cryptography features support both AES-128 and AES-256.

If privacy is an issue, the TOE can be configured not to disclose any privacy-related information to unauthorized users.

1.4.2.3 Documentation

The data sheet [8] and addendum [9] contain a functional description of the communication protocol and the commands implemented by the TOE. The provided documentation can be used by a customer to develop applications using the TOE.

The data sheet is supported by a user guidance manual [10] which gives additional guidance with regards to the secure usage of the TOE.

The Delivery specification data sheet addendum [11] gives additional information regarding the wafer dimensions, TOE identification and delivery processes.

1.4.3 Life Cycle and Delivery of the TOE

The life-cycle phases are organized according to the Security IC Platform Protection Profile with Augmentation Packages [6], Section 1.2.4:

- Phase 1: IC Embedded Software Development

- Phase 2: IC Development
- Phase 3: IC Manufacturing
- Phase 4: IC Packaging
- Phase 5: Composite Product Integration
- Phase 6: Personalisation
- Phase 7: Operational Usage

For the usage phase the MF3E(c)x3 chip will be embedded in a credit card (meaning ID-1 sized) plastic card (micro-module embedded into the plastic card) or another supported package. The module and card embedding of the TOE provide external security mechanisms because they make it harder for an attacker to access parts of the TOE for physical manipulation.

Regarding the Application Note 1 of the Protection Profile [6], NXP will deliver the TOE at the end of Phase 6. Therefore the TOE evaluation perimeter comprising the development and production environment of the TOE, consists of life-cycle phases 1 - 6. The TOE is a fully integrated composite product comprised of the underlying security IC hardware combined with the embedded software developed by NXP. Therefore, Phase 5 is fully under control of NXP and does not involve data exchange with other parties.

NXP also provides a commercial option to configure the TOE on behalf of the customer in order to personalize before the usage. Alternatively, the customer can also finalize the partially personalized TOE after delivery. In case that all required security anchors (key material) are already installed during personalization by NXP, the customer can finalize the personalization of the file system content relying on the operational security features of the TOE.

The TOE is able to control two different logical phases. After production of the chip every start-up will lead to the initial operating mode. In the initial operating mode the production test shall be performed and the TOE is trimmed and initialized. The selection of the required variant is part of the initialization. At the end of the production test, the access to the test and initialization software is disabled. Subsequent start-ups of the chip will always enter the user operating mode with the CPU executing the TOE operating system software. The TOE will stay in the user operating mode until the end of its life-time.

The TOE is being locked to the user operating mode before TOE delivery at the end of Phase 6.

1.4.4 TOE Intended Usage

The TOE user environment is the environment from TOE Delivery to Phase 7. At the phases up to 6, the TOE user environment must be a controlled environment. The only exception is that customer specific keys can be installed using trust provisioning services in Phase 6. In this case the customer can finalize the personalization at the end of Phase 6, already relying on the TOE provided operational security services. Regarding to Phase 7, the TOE is used by the end-user. The method of use of the product in this phase depends on the application. The TOE is intended to be used in an unsecured environment that does not avoid a threat.

The TOE is developed for high-end safeguarded applications, and is designed for embedding into contactless smart cards according to ISO 14443. Alternatively, the TOE may be embedded in a device with an MCU communicating with the TOE over I2C. Usually the device is assigned to a single individual only and may be used for multiple applications in a multi-provider environment. The secret data shall be used as input for the calculation of authentication data, encryption and integrity protection of data for communication.

In the end-user environment (Phase 7) smart card ICs are used in a wide range of applications to assure authorized conditional access. Examples of such are transportation or access management. The end-user environment therefore covers a wide spectrum of very different functions, thus making it difficult to avoid and monitor any abuse of the TOE.

The system integrators such as the terminal software developer may use samples of the TOE during the development phases for their testing purposes. These samples do not differ from the TOE and do not have any additional functionality used for testing.

1.4.5 Interface of the TOE

The electrical interface of the TOE are the pads to connect the RF antenna, which allows communication according to ISO 14443 Type A. The communication protocol complies to part ISO 14443-3.

Alternatively, the TOE can be connected to a host MCU via the pads dedicated for I2C communication. The functional interface is defined by the commands implemented by the TOE and described in the product data sheet.

2 Conformance Claims

2.1 CC Conformance Claim

This Security Target claims to be conformant to the Common Criteria version 3.1:

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 3.1, Revision 5, CCMB-2017-04-001, April 2017 [2].
- Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Version 3.1, Revision 5, CCMB-2017-04-002, April 2017 [3].
- Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, Version 3.1, Revision 5, CCMB-2017-04-003, April 2017 [4].

For the evaluation the following methodology will be used:

- Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Version 3.1, Revision 5, CCMB-2017-04-004, April 2017 [5].

This Security Target claims to be CC Part 2 extended and CC Part 3 conformant. The extended Security Functional Requirements are defined in Section 5.

2.2 Protection Profile Claim

This Security Target claims strict conformance to the following Protection Profile:

- Security IC Platform Protection Profile with Augmentation Packages, Registered and Certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-CC-PP-0084-2014, Version 1.0, 13 January 2014 [6].

2.3 Package Claim

This Security Target claims conformance to the assurance package EAL6 augmented with ASE_TSS.2.

2.4 Conformance Claim Rationale

As the Protection Profile [6] requires strict conformance, no conformance claim requirement is needed in this Security Target.

3 Security Problem Definition

This section lists the assets, threats, organisational security policies and assumptions from the Protection Profile [6] and describes extensions to these elements in detail.

3.1 Description of Assets

The assets to be protected (related to standard functionality) are described in Section 3.1 of the Protection Profile [6] and are listed below:

- The user data of the Composite TOE.
- The Security IC Embedded Software, stored and in operation.
- The security services provided by the TOE for the Security IC Embedded Software.

These assets are related to the following high-level security concerns:

- Integrity of user data of the Composite TOE.
- Confidentiality of user data of the Composite TOE being stored in the TOE's protected memory areas.
- Correct operation of the security services provided by the TOE for the Security IC Embedded Software.
- Deficiency of random numbers.

To be able to protect these assets the TOE shall self-protect its security functionality. Critical information about the security functionality shall be protected by the development environment and the operational environment. Critical information may include:

- Logical design data, physical design data, IC Dedicated Software, and configuration data.
- Initialisation Data and Pre-personalisation Data, specific development aids, test and characterisation related data, material for software development support, and photomasks.

For details see Section 3.1 of the Protection Profile [6].

3.2 Threats

All threats for the TOE which are defined in section 3.2 of the Protection Profile are applied to this Security Target and are listed in Table 5.

Table 5. Threats defined in the Protection Profile (PP-0084)

Name	Title
T.Leak-Inherent	Inherent Information Leakage
T.Phys-Probing	Physical Probing
T.Malfunction	Malfunction due to Environmental Stress
T.Phys-Manipulation	Physical Manipulation
T.Leak-Forced	Forced Information Leakage
T.Abuse-Func	Abuse of Functionality
T.RND	Deficiency of Random Numbers

For details see Section 3.1 of the Protection Profile [6].

The following additional threats are defined in this Security Target:

Table 6. Additional threats defined in this Security Target

Name	Title
T.Data-Modification	Unauthorised Data Modification
T.Impersonate	Impersonating authorised users during authentication
T.Cloning	Cloning

T.Data-Modification

Unauthorised Data Modification

User data stored by the TOE may be modified by unauthorised subjects. This threat applies to the processing of modification commands received by the TOE, it is not concerned with verification of authenticity.

T.Impersonate

Impersonating authorised users during authentication

An unauthorised subject may try to impersonate an authorised subject during the authentication sequence, e.g. by a man-in-the-middle or replay attack.

T.Cloning

Cloning

User and TSF data stored on the TOE (including keys) may be read out by an unauthorised subject in order to create a duplicate.

3.3 Organisational Security Policies

All organisational security policies defined in the Protection Profile are valid for this Security Target and are listed in Table 7.

Table 7. Organisational security policies defined in the Protection Profile (PP-0084)

Name	Title
P.Process-TOE	Identification during TOE Development and Production

For details see Section 3.3 of the Protection Profile [6].

This Security Target defines additional organisational security policies as detailed in the following.

The TOE provides specific security functionality which can used by the operating system. In the following, specific security functionality is listed which is not derived from threats identified for the TOE's environment because it can only be decided in the context of the application against which threats the TOE will use the specific security functionality.

The IC Developer / Manufacturer therefore applies the following policies as specified below.

Table 8. Additional organisational security policies defined in this Security Target

Name	Title
P.Encryption	Confidentiality during communication
P.Integrity	Authenticated integrity during communication
P.Transaction	Transaction mechanism
P.No-Trace	Untraceability of end-users

P.Encryption

Confidentiality during communication

The TOE shall provide the possibility to protect selected data elements from eavesdropping during contactless communication.

P.Integrity

Authenticated integrity during communication

The TOE shall provide the possibility to protect the contactless communication from modification or injections. This includes especially the possibility to detect replay or man-in-the-middle attacks within a session.

P.Transaction

Transaction mechanism

The TOE shall provide the possibility to combine a number of data modification operations in one transaction, so that either all operations or no operation at all is performed.

P.No-Trace

Untraceability of end-users

The TOE shall provide the ability that authorised subjects can prevent that end-user of TOE may be traced by unauthorised subjects without consent. Tracing of end-users may happen by performing a contactless communication with the TOE when the end-user is not aware of it. Typically this involves retrieving the UID or any freely accessible data element.

3.4 Assumptions

All assumptions defined in the Protection Profile are valid for this Security Target and are listed in Table 9.

Table 9. Assumptions defined in the Protection Profile (PP-0084)

Name	Title
A.Process-Sec-IC	Protection during Packaging, Finishing and Personalisation
A.Resp-Appl	Treatment of user data of the Composite TOE

For details see Section 3.4 of the Protection Profile [6].

In compliance with Application Notes 6 and 7 in the Protection Profile [6], this Security Target defines two additional assumptions as follows:

Table 10. Additional assumptions defined in this Security Target

Name	Title
A.Secure-Values	Usage of secure values
A.Terminal-Support	Terminal Support

A.Secure-Values

Usage of secure values

Only confidential and secure cryptographically strong keys shall be used to set up the authentication. These values are generated outside the TOE and they are downloaded to the TOE. Additionally, asymmetric keys may also be generated on the TOE, only exporting the public key. It is assumed that related public keys are properly registered within the system, e.g. by complementing them with a certificate.

A.Terminal-Support

Terminal Support

The terminal verifies information sent by the TOE in order to ensure integrity and confidentiality of the communication. In case of asymmetric authentication, this may include the verification of a certificate provided by the TOE or via other mechanisms. Furthermore the terminal shall provide random numbers and/or ephemeral ECC keys according to AIS20/31 for the authentication.

The additional assumptions as defined above are required for the correct functioning of the operating system's security functionality. As the Protection Profile [6] does not cover this kind of functionality, the additional assumptions neither mitigate a threat (or a part of a threat) meant to be addressed by security objectives for the TOE in the Protection Profile [6], nor fulfil an OSP (or part of an OSP) meant to be addressed by security objectives for the TOE in the Protection Profile [6].

4 Security Objectives

4.1 Security Objectives for the TOE

All security ojectives for the TOE which are defined in section 4.1 of the Protection Profile are applied to this Security Target and are listed in Table 11.

Table 11. Security Objectives of the TOE (PP-0084)

Name	Title
O.Leak-Inherent	Protection against Inherent Information Leakage
O.Phys-Probing	Protection against Physical Probing
O.Malfunction	Protection against Malfunctions
O.Phys-Manipulation	Protection against Physical Manipulation
O.Leak-Forced	Protection against Forced Information Leakage
O.Abuse-Func	Protection against Abuse of Functionality
O.Identification	TOE Identification
O.RND	Random Numbers

Regarding the Application Notes 8 and 9 in the Protection Profile [6], additional security objectives that are based on additional functionality provided by the TOE are defined below:

Table 12. Additional security objectives defined in this Security Target

Name	Title
O.Access-Control	Access Control
O.Authentication	Authentication
O.Encryption	Confidential Communication
O.Integrity	Integrity-Protected Communication
O.No-Trace	Preventing Traceability
O.Transaction	Transaction Mechanism
O.Type-Consistency	Data Type Consistency

O.Access-Control

Access Control

The TOE must provide an access control mechanism for data stored by it. The access control mechanism shall apply to read, modify, create and delete operations for data elements and to reading and modifying security attributes as well as authentication data. It shall be possible to limit the right to perform a specific operation to a specific user. The security attributes (keys) used for authentication shall never be output.

O.Authentication

Authentication

The TOE must provide an authentication mechanism in order to be able to authenticate authorised users. The authentication mechanism shall be resistant against replay and man-in-the-middle attacks.

O.Encryption

Confidential Communication

The TOE must be able to protect the communication by encryption. This shall be implemented by security attributes that enforce encrypted communication for the respective data elements.

O.Integrity

Integrity-Protected Communication

The TOE must be able to protect the communication by adding a MAC or signature, ensuring integrity and authentication of the transferred data. This shall be implemented by security attributes that enforce integrity protected communication for the respective data elements. Usage of the protected communication shall also support the detection of injected and bogus commands within the communication session before the protected data transfer.

O.No-Trace

Preventing Traceability

The TOE must be able to prevent that the TOE end-user can be traced. This shall be done by providing an option that disables the transfer of privacy-related information that is suitable for tracing an end-user by an unauthorised subject.

O.Transaction

Transaction Mechanism

The TOE must be able to provide a transaction mechanism that allows to update multiple data elements either all in common or none of them.

O.Type-Consistency

Data Type Consistency

The TOE must provide a consistent handling of the different supported data types. This comprises over- and underflow checking for values, for data file sizes and record handling.

4.2 Security Objectives for the Security IC Embedded Software

All security objectives for the Security IC Embedded Software which are defined in section 4.2 of the Protection Profile are applied to this Security Target and are listed in Table 13.

Table 13. Security Objectives for the Security IC Embedded Software (PP-0084)

Name	Title
OE.Resp-Appl	Treatment of User Data

4.3 Security Objectives for the Operational Environment

All security objectives for the operational environment which are defined in section 4.3 of the Protection Profile are applied to this Security Target and are listed in Table 14.

Table 14. Security Objectives for the Operational Environment (PP-0084)

Name	Title
OE.Process-Sec-IC	Protection during composite product manufacturing

The following additional security objectives for the operational environment are defined in this Security Target:

Table 15. Additional security objectives for the operational environment defined in this Security Target

Name	Title
OE.Secure-Values	Generation of secure values

Table 15. Additional security objectives for the operational environment defined in this Security Target*...continued*

Name	Title
OE.Terminal-Support	Terminal support to ensure integrity, confidentiality and use of random numbers

The TOE provides specific functionality that requires the TOE Manufacturer to implement measures for the unique identification of the TOE. Therefore, OE.Secure-Values is defined to allow a TOE specific implementation (refer also to A.Secure-Values).

OE.Secure-Values

Generation of Secure Values

The environment shall generate confidential and cryptographically strong keys for authentication purpose. These keys may comprise symmetric keys, asymmetric TOE key pairs from which the ECC Private Key is stored on the TOE, and asymmetric key pairs protecting the access to the TOE, i.e. the key pair from which the CA Root Public Key is stored on the TOE, but also the further key pairs that are certified by the CA. These values are generated outside the TOE and are downloaded to the TOE during the personalisation or usage in phase 5 to 7. Asymmetric TOE key pairs can also be generated by the TOE. In this case the environment shall protect the registration of public keys in the system, e.g. by providing the TOE with a certificate. The environment shall ensure that the generated secure values are kept confidential.

The TOE provides specific functionality to verify the success of the application download process. Therefore, OE.Terminal-Support is defined to allow triggering the verification process.

OE.Terminal-Support

Terminal support to ensure integrity, confidentiality and use of random numbers

The terminal shall verify information sent by the TOE in order to ensure integrity and confidentiality of the communication. This may involve the checking of MAC values, signatures and certificates sent by the TOE, and secure closing of the communication session. Furthermore the terminal shall provide random numbers and/or ephemeral ECC keys according to AIS20/31 [1] for the authentication.

The additional security objectives for the operational environment as defined above are required for the correct functioning of the TOE's security functionality. As the Protection Profile [6] does not cover this kind of functionality, the additional objectives neither mitigate a threat (or a part of a threat) meant to be addressed by security objectives for the TOE in the Protection Profile [6], nor fulfil an OSP (or part of an OSP) meant to be addressed by security objectives for the TOE in the Protection Profile [6].

4.4 Security Objectives Rationale

Section 4.4 in the Protection Profile [6] provides a rationale how the threats, organisational security policies and assumptions are addressed by the security objectives defined in the Protection Profile. This rationale is not repeated here.

The following table summarizes how threats, organisational security policies and assumptions are addressed by the security objectives with respect to those items defined in the Security Target. All these items are in line with those in the Protection Profile [6].

Table 16. Security Problem Definition mapping to Security Objective

Security Problem Definition	Security Objective
T.Data-Modification	O.Access-Control

Table 16. Security Problem Definition mapping to Security Objective*...continued*

Security Problem Definition	Security Objective
	O.Type-Consistency
T.Impersonate	O.Authentication
T.Cloning	O.Access-Control O.Authentication
P.Encryption	O.Encryption
P.Integrity	O.Integrity
P.Transaction	O.Transaction
P.No-Trace	O.Access-Control O.Authentication O.No-Trace
A.Secure-Values	OE.Secure-Values
A.Terminal-Support	OE.Terminal-Support

The rationale for the mapping is given below:

Justification related to T.Data-Modification:

Security Objective	Rationale
O.Access-Control	This objective requires an access control mechanism that limits the ability to modify data and code elements stored by the TOE.
O.Type-Consistency	This objective ensures that data types are adhered, so that TOE data can not be modified by abusing type-specific operations.

Justification related to T.Impersonate:

Security Objective	Rationale
O.Authentication	This objective requires that the authentication mechanism provided by the TOE shall be resistant against attack scenarios targeting the impersonation of authorized users.

Justification related to T.Cloning:

Security Objective	Rationale
O.Access-Control	This objective requires that unauthorized users can not read any information that is restricted to the authorized subjects. The cryptographic keys used for the authentication are stored inside the TOE and are protected by this objective. This objective states that no keys used for authentication shall ever be output.
O.Authentication	This objective requires that users are authenticated before they can read any information that is restricted to authorized users.

Justification related to A.Secure-Values:

Security Objective	Rationale
OE.Secure-Values	This objective is an immediate transformation of the assumption, therefore it covers the assumption.

Justification related to A.Terminal-Support:

Security Objective	Rationale
OE.Terminal-Support	This objective is an immediate transformation of the assumption, therefore it covers the assumption. The TOE can only check the integrity of data received from the terminal. For data transferred to the terminal the receiver must verify the integrity of the received data. Furthermore the TOE cannot verify the entropy of the random number sent by the terminal. The terminal itself must ensure that random numbers are generated with appropriate entropy for the authentication. This is assumed by the related assumption, therefore the assumption is covered.

Justification related to P.Encryption:

Security Objective	Rationale
O.Encryption	This objective is an immediate transformation of the security policy, therefore it covers the security policy.

Justification related to P.Integrity:

Security Objective	Rationale
O.Integrity	This objective is an immediate transformation of the security policy, therefore it covers the security policy.

Justification related to P.Transaction:

Security Objective	Rationale
O.Transaction	This objective is an immediate transformation of the security policy, therefore it covers the security policy.

Justification related to P.No-Trace:

Security Objective	Rationale
O.Access-Control	This objective provides means to implement access control to data elements on the TOE in order to prevent tracing based on freely accessible data elements.
O.Authentication	This objective provides means to implement authentication on the TOE in order to prevent tracing based on freely accessible data elements.
O.No-Trace	This objective requires that the TOE shall provide an option to prevent the transfer of any information that is suitable for tracing an end-user by an unauthorized subject. This objective includes the UID.

The justification of the additional policies and the additional assumptions show that they do not contradict the rationale already given in the Protection Profile [6] for the assumptions, policy and threats defined there.

5 Extended Components Definition

This Security Target defines two additional extended components which are described in the following sections.

Note that the Protection Profile [6] defines extended security functional requirements FCS_RNG.1, FMT_LIM.1, FMT_LIM.2, FAU_SAS.1 and FDP_SDC.1 in chapter 5, which are used in this Security Target but their definitions are not duplicated in this section.

5.1 Export of user data in unauthenticated state (FDP_ETC.3)

To define the Secure Dynamic Messaging functionality of the TOE, an additional component FDP_ETC.3 of the family FDP_ETC (export from the TOE) of the class FDP (user data protection) is defined. The class and family behaviour of FDP_ETC are defined in CC Part 2 [3].

As defined in CC Part 2 [3], the FDP class addresses user data protection. The FDP_ETC family defines functions for TSF-mediated exporting of user data from the TOE such that its security attributes and protection either can be explicitly preserved or can be ignored once it has been exported. The extended component FDP_ETC.3 (Export of user data in unauthenticated state) addresses a similar concern but does not require a TOE enforcement of an access control SFP(s) and/or information flow control SFP(s) as the already defined components of the FDP_ETC family.

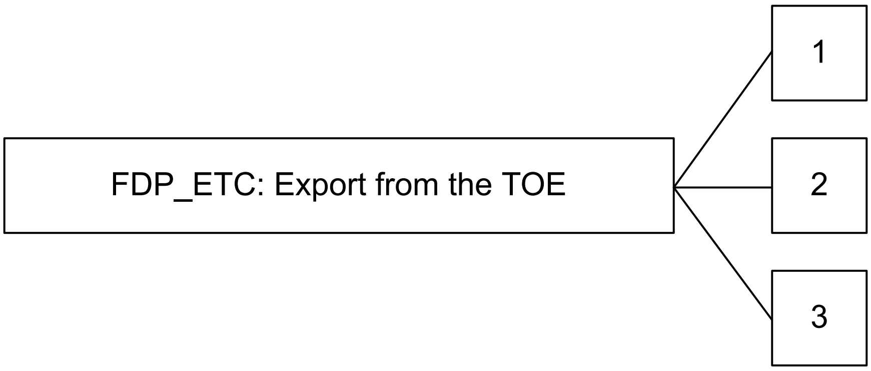

Figure 1. Component levelling of Extended Component FDP_ETC

FDP_ETC Export from the TOE

Management: FDP_ETC.3

There are no management activities foreseen.

Audit: FDP_ETC.3

There are no actions defined to be auditable.

FDP_ETC.3 Export of user data in unauthenticated state

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_ETC.3.1	**The TSF shall export the following pieces of user data: [assignment: *pieces of user data*] with the following user data's associated security attributes: [assignment: *list of security attributes*].**
FDP_ETC.3.2	**The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data.**
FDP_ETC.3.3	**The TSF shall enforce the following rules when user data is exported from the TOE: [assignment: *additional exportation control rules*]**

The extended component is defined to capture the Secure Dynamic Messaging feature provided by the TOE, which allows for the encrypted and authenticated extraction of user data without the need of establishing a trusted channel beforehand. Due to this specific property, the existing data export SFRs FDP_ETC.1 and FDP_ETC.2 did not apply well.

5.2 Authentication Proof of Identity (FIA_API.1)

To define the Transaction Signature functionality of the TOE, an additional family (FIA_API) of the class FIA (Identification and authentication) is taken from Protection Profile PP-0056 [7] and its definition is repeated below. The class behaviour of FIA is defined in CC Part 2 [3].

The family FIA_API describes the functional requirements for the proof of the claimed identity for the authentication verification by an external entity where the other families of the class FIA address the verification of the identity of an external entity.

Family behaviour:

This family defines functions provided by the TOE to prove their identity and to be verified by an external entity in the TOE IT environment.

Component leveling:

FIA_API: Authentication Proof of Identity
1

Figure 2. Component levelling of Extended Component FIA_API

FIA_API	Authentication Proof of Identity
Management:	FIA_API.1

The following actions could be considered for the management functions in FMT: Management of authentication information used to prove the claimed identity.

Audit: FIA_API.1

There are no actions defined to be auditable.

FIA_API.1 **Authentication Proof of Identity**

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_API.1.1 **The TSF shall provide a [assignment: *authentication mechanism*] to prove the identity of the [assignment: *authorized user or role*].**

6 Security Requirements

This chapter defines the security requirements that shall be met by the TOE. These security requirements are composed of the security functional requirements and the security assurance requirements that the TOE must meet in order to achieve its security objectives.

CC allows several operations to be performed on security requirements (on the component level); refinement, selection, assignment, and iteration are defined in section 8.1 of CC Part 1 [2]. These operations are used in this Security Target.

The refinement operation is used to add details to requirements, and thus, further intensifies a requirement.

The selection operation is used to select one or more options provided by the Protection Profile or CC in stating a requirement. Selections having been made are denoted as italic text.

The assignment operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments having been made are denoted as italic text.

The iteration operation is used when a component is repeated with varying operations. For the sake of a better readability, the iteration operation may also be applied to some single components (being not repeated) in order to indicate belonging of such SFRs to same functional cluster. In such a case, the iteration operation is applied to only one single component.

6.1 Security Functional Requirements

6.1.1 Security Functional Requirements from the Protection Profile

6.1.1.1 FAU_SAS.1

The TOE shall meet the requirement "Audit storage" as defined in the PP [6], and as specified below.

FAU_SAS.1	**Audit storage**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FAU_SAS.1.1	The TSF shall provide *the test process before TOE Delivery* with the capability to store *the Initialisation Data, Pre-personalisation Data, Customer-specific Data*[1] in the *non-volatile memory*[2].

6.1.1.2 FCS_RNG.1/PTG2

The TOE shall meet the requirement "Random number generation (Class PTG.2)" as defined in the PP [6] according to [1] , and as specified below.

FCS_RNG.1/PTG2	**Random number generation (Class PTG.2)**
Hierarchical to:	No other components.

[1] [selection: *the Initialisation Data, Pre-personalisation Data, [assignment: other data]*]

[2] [assignment: *type of persistent memory*]

Dependencies: No dependencies.

FCS_RNG.1.1/PTG2 The TSF shall provide a *physical*[3] random number generator that implements:[4]

(PTG.2.1) A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.

(PTG.2.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG *prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source*[5].

(PTG.2.3) The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected.

(PTG.2.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.

(PTG.2.5) The online test procedure checks the quality of the raw random number sequence. It is triggered *at regular intervals or continuously*[6]. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.

FCS_RNG.1.2/PTG2 The TSF shall provide *octets of bits*[7] that meet:

(PTG.2.6) Test procedure A[8] does not distinguish the internal random numbers from output sequences of an ideal RNG.

(PTG.2.7) The average Shannon entropy per internal random bit exceeds 0.997.

6.1.1.3 FCS_RNG.1/DRG4

The TOE shall meet the requirement "Random number generation (Class DRG.4)" as defined in the PP [6] according to [1] , and as specified below.

FCS_RNG.1/DRG4 **Random number generation (Class DRG.4)**

[3] [selection: *physical, hybrid physical, hybrid deterministic*]
[4] [assignment: *list of security capabilities*]
[5] [selection: *prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source, generates the internal random numbers with a post-processing algorithm of class DRG.2 as long as its internal state entropy guarantees the claimed output entropy*]
[6] [selection: *externally, at regular intervals, continuously, applied upon specified internal events*]
[7] [selection: *bits, octets of bits, numbers [assignment: format of the numbers]*]
[8] [assignment: *additional standard test suites*]. Assignment is empty as per Application Note 44 of the PP.

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1/DRG4 The TSF shall provide a *hybrid deterministic*[9] random number generator that implements:[10]

(DRG.4.1)The internal state of the RNG shall *use PTRNG of class PTG.2 as random source*[11].

(DRG.4.2) The RNG provides forward secrecy.

(DRG.4.3) The RNG provides backward secrecy even if the current internal state is known.

(DRG.4.4) The RNG provides enhanced forward secrecy *on demand*[12].

(DRG.4.5) The internal state of the RNG is seeded by an *PTRNG of class PTG.2*[13].

FCS_RNG.1.2/DRG4 The TSF shall provide random numbers that meet:

(DRG.4.6) The RNG generates output for which 2^{48} strings[14] of bit length 128 are mutually different with probability *of at least* $1 - 2^{-24}$[15].

(DRG.4.7) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A *and no additional test suites*[16].

6.1.1.4 FDP_SDC.1

The TOE shall meet the requirement "Stored data confidentiality" as defined in the PP [6], and as specified below.

FDP_SDC.1 **Stored data confidentiality**

Hierarchical to: No other components.

Dependencies: No dependencies.

[9] [selection: *physical, hybrid physical, hybrid deterministic*]
[10] [assignment: *list of security capabilities*]
[11] [selection: *use PTRNG of class PTG.2 as random source, have [assignment: work factor], require [assignment: guess work]*]
[12] [selection: *on demand, on condition [assignment: condition], after [assignment: time]*]
[13] [selection: *internal entropy source, PTRNG of class PTG.2, PTRNG of class PTG.3, [other selection]*]
[14] [assignment: *number of strings*]
[15] [assignment: *probability*]
[16] [assignment: additional test suites]

FDP_SDC.1.1	The TSF shall ensure the confidentiality of the information of the user data while it is stored in the *RAM and non-volatile memory*[17].

6.1.1.5 FDP_SDI.2

The TOE shall meet the requirement "Stored data integrity monitoring and action" as defined in the PP [6], and as specified below.

FDP_SDI.2	**Stored data integrity monitoring and action**
Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring
Dependencies:	No dependencies.
FDP_SDI.2.1	The TSF shall monitor user data stored in containers controlled by the TSF for *modification, deletion, repetition or loss of data*[18] on all objects, based on the following attributes: *integrity check information associated with the data stored in memories*[19].
FDP_SDI.2.2	Upon detection of a data integrity error, the TSF shall *perform an error correction if possible or trigger a security reset if not*[20].

6.1.2 Security Functional Requirements regarding Access Control

6.1.2.1 TOE Access Control Policy

The Security Functional Policy (SFP) *TOE Access Control Policy* uses the definitions listed in this paragraph. The defined subjects are:

Subject	Admin Administrator
Info	The Admin is the subject that owns or has access to the PICCMasterKey, or has equivalent access rights granted by an AdminCA.
Info	The Admin is the subject that distributes the PICCDAMAuthKey, DAMMACs, and DAMENCs containing the AppDAMDefaultKey, to the DelAppMgr.
Info	The Admin is the subject that owns or delegates the right to change the PICCCARootKeys and their related access rights.

Subject	AdminCA Administrator CA
Info	The AdminCA is the subject knows the private key related to an PICCCARootKey, and therefore can issue certificates holding (a subset of) the access rights related to that PICCCARootKey.

[17] [assignment: *memory area*]
[18] [assignment: *integrity errors*]
[19] [assignment: *user data attributes*]
[20] [assignment: *action to be taken*]

Subject	AdminCA	Administrator CA
Info	The AdminCA is the subject that, if his PICCCARootKey is associated with the ECC-based Delegated Application Management access right, can distribute delegated application management access rights by issuing such certificates to a DelAppMgr.	
Info	Note that this subject cannot directly authenticate against the TOE, but rather can grant the possiblity to authentication to other subjects via issuing certificates.	

Subject	PICCUser	PICC User
Info	The PICCUser is the subject that owns or has access to the VCConfigurationKey, or has this or other PICC level access rights granted by an AdminCA.	
Info	Note that the TOE supports multiple PICCUser at the PICC level and the assigned rights to the PICCUser can be different, which allows to have more or less powerful PICCUser.	

Subject	AppMgr	Application Manager
Info	The AppMgr is the subject that owns or has access to an AppMasterKey, or has equivalent access rights granted by an AppCA. Note that the TOE supports multiple Applications and therefore multiple AppMgr. Within one Application the role can also be issued to multiple instances, e.g. through certificates associated with different readers.	
Info	The AppMgr is the subject that owns or delegates the right to change the AppCARootKeys and their related access rights	

Subject	DelAppMgr	Delegated Application Manager
Info	The DelAppMgr is the subject that has received delegated applicaton mangement access rights. This can be either through a valid DAMMAC, the PICCDAMAuthKey, and a DAMENC containing the AppDAMDefaultKey, or by knowing the private key related to a delegated application management certificate	
Info	Note that the TOE supports multiple DelApplications and therefore multiple DelAppMgr.	

Subject	AppUser	Application User
Info	The AppUser is the subject that owns or has access to an AppKey, or has one or more equivalent access rights granted by an AppCA.	
Info	Note that the TOE supports multiple AppUser within each Application and the assigned rights to the AppUser can be different, which allows to have more or less powerful AppUser.	

Subject	AppChangeUser	Application Change User
Info	The AppChangeUser is the subject that owns or has access to an AppChangeKey, or has equivalent access rights granted by an AppCA.	

Subject	AppRollUser	Application Roll Key Set User
Info	The AppRollUser is the subject that owns or has access to an AppRollKey, or has equivalent access rights granted by an AppCA.	

Subject	AppCA	Application CA
Info	The AppCA is the subject knows the private key related to an AppCARootKey, and therefore can issue certificates holding (a subset of) the access rights related to that AppCARootKey.	
Info	Note that this subject cannot directly authenticate against the TOE, but rather can grant the possiblity to authentication to other subjects via issuing certificates.	

Subject	Anybody	Anybody
Info	Any subject that does not belong to one of the roles Admin, PICCUser, AppMgr, DelAppMgr, AppUser, AppChangeUser or AppRollUser, belongs to the role Anybody. This role includes the card holder (also referred to as end-user), and any other subject like an attacker for instance. The subjects belonging to Anybody do not possess any key and therefore are not able to perform any operation that is restricted to one of the roles which are explicitely excluded from the role Anybody.	
Info	Additionally, in product configurations with dual interface (I2C and NFC) certain access rights can be granted to Anybody but restricted to one of the interfaces. If only free access over I2C is configured, it means the access right is granted to Anybody over the I2C interface, but not over the NFC interface.	

Subject	Nobody	Nobody
Info	Any subject that does not belong to one of the roles Admin, PICCUser, AppMgr, DelAppMgr, App User, AppChangeUser, AppRollUser or Anybody, belongs to the role Nobody. Due to the definition of Anybody, the set of all subjects belonging to the role Nobody is the empty set.	

The objects defined for the *TOE Access Control Policy* are:

Object	PICCLevelData	PICC Level Data
Info	The PICC level is the lowest level of the TOE (PICC level, Application level, File level). On the PICC level Applications, DelApplications and Files can be created or deleted. Hence to the PICCLevelData belong Applications, DelApplications and Files.	
Operation	Modify	Modify attributes of PICCLevelData.
Operation	Freeze	Freeze attributes of PICCLevelData.
Attribute	PICCKeySettings	Generic PICC key settings.

Object	Application	Application
Info	The card can store a number of Applications. An Application can store a number of Files.	
Operation	Modify	Modify attribute of an Application.
Operation	Freeze	Freeze attribute of an Application.
Operation	Create	Create an Application.

Object	Application	Application
Operation	Delete	Delete an Application.
Operation	Select	Select an Application.
Attribute	AppKeySettings	Generic application key settings.
Attribute	ECCKeyManagement	ECC key management access conditions.

Object	DelApplication	Delegated Application
Info	The card can store a number of DelApplications. After creation the DelApplication has the same attributes as a Application.	
Operation	Create	Create a DelApplication.
Operation	Delete	Delete a DelApplication.

Object	File	File
Info	An Application can store a number of Files of different types. Also the PICC level can store a number of Files, but this is restricted to StandardData Files.	
Operation	Create	Create a File.
Operation	Delete	Delete a File.
Operation	Freeze	Freeze attributes of a File.
Operation	Read	Read operations accessing the content of a File.
Operation	Write	Write operations accessing the content of a File.
Operation	ReadWrite	ReadWrite operations accessing the content of a File.
Operation	Change	Change operation to change the attribute File.AccessRights.
Attribute	AccessRights	Generic access rights for a File.
App Note	Certificates used to authenticate the TOE are stored within data files. Therefore, the regular operations apply and they are not specified as a separate Object type.	
App Note	The CRLFile as mentioned in FMT_REV.1 is a specific type of data File.	

Object	PICCMasterKey	PICC Master Key
Info	The Card Master Key.	
Operation	Change	Change the PICCMasterKey.
Operation	Freeze	Freeze the PICCMasterKey.

Object	VCConfigurationKey	VC Configuration Key
Info	The VC Configuration Key.	
Operation	Change	Change the VCConfigurationKey.

Object	PICCAppDefaultKey	PICC Application Default Key
Info	The Default Application Master Key and Application Keys that are used when an application is created and when a KeySet is initialized.	
Operation	Change	Change the PICCAppDefaultKey.

Object	PICCDAMAuthKey	PICC DAM Authentication Key
Info	Delegated Application Management Authentication Key.	
Operation	Change	Change the PICCDAMAuthKey.

Object	PICCDAMENCKey	PICC DAM Encryption Key
Info	Delegated Application Management Encryption Key to generate DAMENC.	
Operation	Change	Change the PICCDAMENCKey.

Object	PICCDAMMACKey	PICC DAM MAC Key
Info	Delegated Application Management MAC Key to generate DAMMAC.	
Operation	Change	Change the PICCDAMMACKey.

Object	PICCCARootKey	PICC CA Root Key
Info	CA Root Key at PICC level	
Operation	Create	Create the PICCCARootKey and its related attributes.
Operation	Change	Change the PICCCARootKey and its related attributes.
Attribute	AccessRights	Access rights granted to this PICCCARootKey.
Attribute	WriteAccess	Access condition for PICCCARootKey.Change.

Object	PICCECCPrivateKey	PICC ECC Private Key
Info	ECC Private Key at PICC level	
Operation	Change	Change the PICCECCPrivateKey and/or its related attributes.
Attribute	KeyPolicy	Key policy defining the operations allowed with this PICCECCPrivateKey.
Attribute	WriteAccess	Access condition for PICCECCPrivateKey.Change.

Object	AppMasterKey	Application Master Key
Info	The Application Master Key.	

Object	AppMasterKey	Application Master Key
Operation	Change	Change the AppMasterKey.
Operation	Freeze	Freeze the AppMasterKey.

Object	AppChangeKey	Application Change Key
Info	Application Change Key.	
Operation	Change	Change the AppChangeKey.

Object	AppKey	Application Key
Info	Application Key.	
Operation	Change	Change the AppKey.

Object	AppTransactionMACKey	Application Transaction MAC Key
Info	Application Transaction MAC Key.	
Operation	Create	Create the AppTransactionMACKey.
Operation	Delete	Delete the AppTransactionMACKey.

Object	AppRollKey	Application Roll Key Set Key
Info	Application Roll Key Set Key.	
Operation	Change	Change the AppRollKey.

Object	AppDAMDefaultKey	Application DAM Default Key
Info	Delegated Application Management Default Authentication Key	

Object	AppCARootKey	Application CA Root Key
Info	CA Root Key at Application level	
Operation	Create	Create the AppCARootKey and its related attributes.
Operation	Change	Change the AppCARootKey and its related attributes.
Attribute	AccessRights	Access rights granted to this AppCARootKey
Attribute	WriteAccess	Access condition for AppCARootKey.Change.

Object	AppECCPrivateKey	Application ECC Private Key
Info	ECC Private Key at Application level	
Operation	Change	Change the AppECCPrivateKey and/or its related attributes.
Attribute	KeyPolicy	Key policy defining the operations allowed with this App ECCPrivateKey.
Attribute	WriteAccess	Access condition for AppECCPrivateKey.Change.

Object	KeySet	Key Set
Info	AppKeys are grouped into KeySets.	
Operation	Roll	Roll the KeySet.

Note that subjects are authorized by cryptographic keys and certificates. These keys are considered as authentication data and not as security attributes of the subjects. The card has a card master key PICCMasterKey and up to two PICCCARootKeys. Every Application has a variable number of AppKeys (from which the one with KeyNo 0x0 is the AppMasterKey) organized in KeySets, and up to 5 AppCARootKeys. These keys are used to authorise operations on Files.

6.1.2.2 FDP_ACC.1

The TOE shall meet the requirement "Subset access control" as specified below.

FDP_ACC.1	**Subset access control**
Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1	The TSF shall enforce the *TOE Access Control Policy*[21] on *all subjects, objects, operations and attributes defined by the TOE Access Control Policy*[22].

6.1.2.3 FDP_ACF.1

The TOE shall meet the requirement "Security attribute based access control" as specified below.

FDP_ACF.1	**Security attribute based access control**
Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1	The TSF shall enforce the *TOE Access Control Policy*[23] to objects based on the following: *all subjects, objects and attributes*[24].

[21] [assignment: *access control SFP*]
[22] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]
[23] [assignment: *access control SFP*]

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:[25]

1. *The Admin is allowed to perform Application.Create and Application.Delete.*
2. *The Admin is allowed to perform DelApplication.Delete.*
3. *The Admin is allowed to perform File.Create and File.Delete.*
4. *The AppMgr is allowed to perform File.Create and File.Delete.*
5. *The DelAppMgr is allowed to perform DelApplication.Create or DelApplication.Delete with either a valid DAMMAC and for DelApplication.Create a valid DAMENC, or a valid delegated application management certificate.*

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:[26]

1. *The Admin or PICCUser is allowed to perform File.Read or File.Write or File.ReadWrite or File.Change on a File at PICC level if the File.AccessRights grant these rights.*
2. *The AppMgr is allowed to Application.Delete if the attribute PICCLevelData.PICCKeySettings grant this right.*
3. *The AppMgr, AppChangeUser, AppRollUser or AppUser is allowed to perform File.Read or File.Write or File.ReadWrite or File.Change on a File at Application level if the File.AccessRights grant these rights.*
4. *The Anybody is allowed to perform Application.Create if the PICCLevelData.PICCKeySettings grant this right.*
5. *The Anybody is allowed to perform File.Create and File.Delete if the Application.AppKeySettings grant these rights.*
6. *The Anybody is allowed to perform File.Read or File.Write or File.ReadWrite or File.Change if the File.AccessRights grant these rights.*

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules:[27]

1. *No one but Nobody is allowed to perform File.Read or File.Write or File.ReadWrite or File.Change if the File.AccessRights do not grant this right.*

6.1.2.4 FDP_ITC.2

The TOE shall meet the requirement "Import of user data with security attributes" as specified below.

FDP_ITC.2 **Import of user data with security attributes**

Hierarchical to: No other components.

[24] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

[25] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

[26] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

[27] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path], FPT_TDC.1 Inter-TSF basic TSF data consistency
FDP_ITC.2.1	The TSF shall enforce the *TOE Access Control Policy*[28] when importing user data, controlled under the SFP, from outside of the TOE.
FDP_ITC.2.2	The TSF shall use the security attributes associated with the imported user data.
FDP_ITC.2.3	The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.
FDP_ITC.2.4	The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.
FDP_ITC.2.5	The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: *no additional rules*[29].

6.1.2.5 FMT_MSA.1

The TOE shall meet the requirement "Management of security attributes" as specified below.

FMT_MSA.1	**Management of security attributes**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1	The TSF shall enforce the *TOE Access Control Policy*[30] to restrict the ability to *modify, change or freeze*[31] the security attributes *of the objects PICCLevelData, Application and the security attribute File.AccessRights*[32] to *the Admin, PICCUser, AppMgr, AppChangeUser, AppRollUser or AppUser respectively*[33].
Refinement:	The detailed management abilities are: 1. *Only the Admin is allowed to perform PICCLevelData.Modify or PICCLevelData.Freeze on PICCLevelData.PICCKeySettings.* 2. *Only the AppMgr is allowed to perform Application.Modify or Application.Freeze on Application.AppKeySettings and Application.ECCKeyManagement.*

[28] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[29] [assignment: *additional importation control rules*]
[30] [assignment: *access control SFP(s), information flow control SFP(s)*]
[31] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[32] [assignment: *list of security attributes*]
[33] [assignment: *the authorised identified roles*]

3. *The Admin or PICCUser with Change access rights is allowed to perform File.Change and File.Freeze on File.AccessRights at PICC level.*
4. *The AppMgr, AppChangeUser, AppRollUser or AppUser with Change access rights is allowed to perform File.Change and File.Freeze on File.AccessRights at Application level.*

6.1.2.6 FMT_MSA.3

The TOE shall meet the requirement "Static attribute initialization" as specified below.

FMT_MSA.3	**Static attribute initialization**
Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes, FMT_SMR.1 Security roles
FMT_MSA.3.1	The TSF shall enforce the *TOE Access Control Policy*[34] to provide *permissive*[35] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow the *no one but Nobody*[36] to specify alternative initial values to override the default values when an object or information is created.
Application Note:	In the default configuration, the only initial attributes are the card attributes. All other attributes have to be defined at the same time the respective object is created. Upon customer request, the file system can be further instantiated during the initialization of the product. Also then, the TOE Access Control Policy does not allow the creation and consequently the manipulation of the default values in operational mode.

6.1.2.7 FMT_MTD.1

The TOE shall meet the requirement "Management of TSF data" as specified below.

FMT_MTD.1	**Management of TSF data**
Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions
FMT_MTD.1.1	The TSF shall restrict the ability to *perform*[37] the *Create, Change and Freeze operations for Keys*[38] to *specific roles depending on the targeted Key and certain attributes.*[39].

[34] [assignment: *access control SFP, information flow control SFP*]
[35] [selection, choose one of: *restrictive, permissive, [assignment: other property]*]
[36] [assignment: *the authorised identified roles*]
[37] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[38] [assignment: *list of TSF data*]
[39] [assignment: *the authorised identified roles*]

Refinement: The detailed management abilities are:

1. *Only the Admin is allowed to perform PICCMasterKey.Change or PICCMasterKey.Freeze.*
2. *The Admin is allowed to perform PICCAppDefaultKey.Change.*
3. *The Admin is allowed to perform PICCDAMAuthKey.Change.*
4. *The Admin is allowed to perform PICCDAMENCKey.Change.*
5. *The Admin is allowed to perform PICCDAMMACKey.Change.*
6. *The Admin or PICCUser with VCConfigurationKey access rights is allowed to perform VCConfigurationKey.Change.*
7. *The Admin or PICCUser is allowed to perform PICCCARootKey.Change if PICCCARootKey.WriteAccess grants this right.*
8. *The Anybody is allowed to perform PICCCARootKey.Change if PICCLevelData.WriteAccess grants this right.*
9. *The Admin or PICCUser is allowed to perform PICCECCPrivateKey.Change if PICCECCPrivateKey.WriteAccess grants this right.*
10. *The Anybody is allowed to perform PICCECCPrivateKey.Change if PICCECCPrivateKey.WriteAccess grants this right.*
11. *The AppMgr is allowed to perform AppMasterKey.Change and AppMasterKey.Freeze.*
12. *The AppMgr is allowed to perform AppChangeKey.Change.*
13. *The AppMgr is allowed to perform AppRollKey.Change.*
14. *The AppMgr is allowed to perform AppTransactionMACKey.Create and AppTransactionMACKey.Delete.*
15. *The AppChangeUser is allowed to perform AppKey.Change.*
16. *The AppUser is allowed to perform AppKey.Change on AppKey if Application.AppKeySettings grant this right.*
17. *The Anybody is allowed to perform AppTransactionMACKey.Create and AppTransactionMACKey.Delete on AppTransactionMACKey if Application.AppKeySettings grant this right.*
18. *The AppRollUser is allowed to perform KeySet.Roll.*
19. *The AppMgr, AppChangeUser, AppRollUser or AppUser is allowed to perform AppCARootKey.Create if Application.ECCKeyManagement grants this right. In the default configuration, this is granted to the AppMgr.*
20. *The Anybody is allowed to perform AppCARootKey.Create if Application.ECCKeyManagement grants this right.*
21. *The AppMgr, AppChangeUser, AppRollUser or AppUser is allowed to perform AppCARootKey.Change if AppCARootKey.WriteAccess grants this right.*
22. *The Anybody is allowed to perform AppCARootKey.Change if AppCARootKey.WriteAccess grants this right.*
23. *The AppMgr, AppChangeUser, AppRollUser or AppUser is allowed to perform AppECCPrivateKey.Create if Application.ECCKeyManagement grants this right. In the default configuration, this is granted to the AppMgr.*
24. *The Anybody is allowed to perform AppECCPrivateKey.Create if Application.ECCKeyManagement grants this right.*
25. *The AppMgr, AppChangeUser, AppRollUser or AppUser is allowed to perform AppECCPrivateKey.Change if AppECCPrivateKey.WriteAccess grants this right.*

26. *The Anybody is allowed to perform AppECCPrivateKey.Change if AppECCPrivateKey.WriteAccess grants this right.*

6.1.2.8 FMT_SMF.1

The TOE shall meet the requirement "Specification of Management Functions" as specified below.

FMT_SMF.1	**Specification of Management Functions**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions:[40] • *Authenticate a user.* • *Invalidating the current authentication state based on the functions: Selecting an application or the card, Changing the key corresponding to the current authentication, Occurrence of any error during the execution of a command, starting a new authentication, Rolling key set, Failed Proximity Check, Deleting an Application as AppMgr, Reset.* • *Changing a security attribute.* • *Rolling the Key Set.* • *Creating or deleting an application, a delegated application or a file.*

6.1.2.9 FMT_SMR.1

The TOE shall meet the requirement "Security roles" as specified below.

FMT_SMR.1	**Security roles**
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification
FMT_SMR.1.1	The TSF shall maintain the roles *Admin, PICCUser, AppMgr, DelAppMgr, AppUser, AppChangeUser, AppRollUser and Anybody*[41].
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

6.1.2.10 Implications of the TOE Access Control Policy

The *TOE Access Control Policy* has some implications, that can be drawn from the policy and that are essential parts of the TOE security functions:

- The TOE end-user does normally not belong to the group of authorised users, but regarded as Anybody by the TOE. This means that the TOE cannot determine if it is used by its intended end-user.

[40] [assignment: *list of management functions to be provided by the TSF*]

[41] [assignment: *the authorised identified roles*]

- The TOE does not offer any functionality to read out symmetric keys or asymmetric private keys.

6.1.3 Security Functional Requirements regarding Confidentiality, Authentication and Integrity

6.1.3.1 FCS_COP.1/AES

The TOE shall meet the requirement "Cryptographic Operation (AES)" as specified below.

FCS_COP.1/AES	**Cryptographic Operation (AES)**
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/AES	The TSF shall perform *encryption and decryption and cipher based MAC for authentication and communication*[42] in accordance with the specified cryptographic algorithm *Advanced Encryption Standard AES in one of the following modes of operation: CBC, CMAC*[43] and cryptographic key sizes *128 bits and 256 bits*[44] that meet the following:[45] • *FIPS PUB 197 [14] (AES)* • *NIST SP 800-38A [15] (CBC mode)* • *NIST SP 800-38B [16] (CMAC mode)*

6.1.3.2 FCS_COP.1/ECDSA

The TOE shall meet the requirement "Cryptographic Operation (ECDSA)" as specified below.

FCS_COP.1/ECDSA	**Cryptographic Operation (ECDSA)**
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/ECDSA	The TSF shall perform *signature generation and verification*[46] in accordance with the specified cryptographic algorithm *ECDSA / ECC over GF(p) (i.e. NIST P-256 or brainpoolP256r1)*[47] and cryptographic key sizes *256 bits*[48] that meet the following:[49] *FIPS PUB 186-5 [13]*.

[42] [assignment: *list of cryptographic operations*]
[43] [assignment: *cryptographic algorithm*]
[44] [assignment: *cryptographic key sizes*]
[45] [assignment: *list of standards*]
[46] [assignment: *list of cryptographic operations*]
[47] [assignment: *cryptographic algorithm*]
[48] [assignment: *cryptographic key sizes*]

6.1.3.3 FCS_COP.1/ECDH

The TOE shall meet the requirement "Cryptographic Operation (ECDH)" as specified below.

FCS_COP.1/ECDH	**Cryptographic Operation (ECDH)**
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/ECDH	The TSF shall perform *Diffie-Hellman Key Exchange*[50] in accordance with the specified cryptographic algorithm *ECDH / ECC over GF(p) (i.e. NIST P-256 or brainpoolP256r1)*[51] and cryptographic key sizes *256 bits*[52] that meet the following:[53] *NIST SP800-56A [17]*.

6.1.3.4 FCS_COP.1/SHA

The TOE shall meet the requirement "Cryptographic Operation (SHA)" as specified below.

FCS_COP.1/SHA	**Cryptographic Operation (SHA)**
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/SHA	The TSF shall perform *hashing*[54] in accordance with the specified cryptographic algorithm *SHA-256*[55] and cryptographic key sizes *none*[56] that meet the following:[57] *FIPS 180-4 [12]*.

6.1.3.5 FCS_CKM.1/Session_AES

The TOE shall meet the requirement "Cryptographic key generation (Session AES)" as specified below.

FCS_CKM.1/ Session_AES	**Cryptographic key generation (Session AES)**

[49] [assignment: *list of standards*]
[50] [assignment: *list of cryptographic operations*]
[51] [assignment: *cryptographic algorithm*]
[52] [assignment: *cryptographic key sizes*]
[53] [assignment: *list of standards*]
[54] [assignment: *list of cryptographic operations*]
[55] [assignment: *cryptographic algorithm*]
[56] [assignment: *cryptographic key sizes*]
[57] [assignment: *list of standards*]

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/ Session_AES The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *AES-based Symmetric Authentication Session Key Generation*[58] and specified cryptographic key sizes *128 bits and 256 bits*[59] that meets the following: *MF ECC refarch section 4.8.4 [8]*[60].

6.1.3.6 FCS_CKM.1/Session_ECC

The TOE shall meet the requirement "Cryptographic key generation (Session ECC)" as specified below.

FCS_CKM.1/ Session_ECC **Cryptographic key generation (Session ECC)**

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/ Session_ECC The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *ECC-based Mutual and Reader-Unilateral Authentication Session Key Generation*[61] and specified cryptographic key sizes *128 bits*[62] that meets the following: *MF ECC refarch section 4.6.4 [8]*[63].

6.1.3.7 FCS_CKM.1/ECC

The TOE shall meet the requirement "Cryptographic key generation (ECC)" as specified below.

FCS_CKM.1/ECC **Cryptographic key generation (ECC)**

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/ECC The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *ECDSA (ECC over GF(p))*[64] and specified cryptographic key sizes *256 bits*[65] that meets the following: *FIPS PUB 186-5 [13]*[66].

[58] [assignment: *cryptographic key generation algorithm*]
[59] [assignment: *cryptographic key sizes*]
[60] [assignment: *list of standards*]
[61] [assignment: *cryptographic key generation algorithm*]
[62] [assignment: *cryptographic key sizes*]
[63] [assignment: *list of standards*]
[64] [assignment: *cryptographic key generation algorithm*]

6.1.3.8 FCS_CKM.4

The TOE shall meet the requirement "Cryptographic key destruction" as specified below.

FCS_CKM.4	**Cryptographic key destruction**
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *overwriting*[67] that meets the following: *none*[68].

6.1.3.9 FIA_UAU.2

The TOE shall meet the requirement "User authentication before any action" as specified below.

FIA_UAU.2	**User authentication before any action**
Hierarchical to:	FIA_UAU.1 Timing of authentication
Dependencies:	FIA_UID.1 Timing of identification
FIA_UAU.2.1	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

6.1.3.10 FIA_UAU.3

The TOE shall meet the requirement "Unforgeable authentication" as specified below.

FIA_UAU.3	**Unforgeable authentication**
Hierarchical to:	No other components
Dependencies:	No dependencies
FIA_UAU.3.1	The TSF shall *detect and prevent*[69] use of authentication data that has been forged by any user of the TSF.

[65] [assignment: *cryptographic key sizes*]
[66] [assignment: *list of standards*]
[67] [assignment: *cryptographic key destruction method*]
[68] [assignment: *list of standards*]
[69] [selection: *detect, prevent*]

FIA_UAU.3.2	The TSF shall *detect and prevent*[70] use of authentication data that has been copied from any other user of the TSF.

6.1.3.11 FIA_UAU.5

The TOE shall meet the requirement "Multiple authentication mechanisms" as specified below.

FIA_UAU.5	**Multiple authentication mechanisms**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UAU.5.1	The TSF shall provide *'none', AES-based symmetric mutual authentication, ECC-based mutual and reader-unilateral authentication*[71] to support user authentication.
FIA_UAU.5.2	The TSF shall authenticate any user's claimed identity according to the *following rules:*[72] • *The 'none' authentication is performed with anyone who communicates with the TOE without issuing an explicit authentication request. The 'none' authentication implicitly and solely authorizes the 'Everybody' subject.* • *The AES-based symmetric mutual authentication and ECC-based mutual and reader-unilateral authentication is used to authorise the Administrator, PICC User, Application Manager, Delegated Application Manager and Application User.*

6.1.3.12 FIA_UID.2

The TOE shall meet the requirement "User identification before any action" as specified below.

FIA_UID.2	**User identification before any action**
Hierarchical to:	FIA_UID.1 Timing of identification
Dependencies:	No dependencies.
FIA_UID.2.1	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.
Application Note:	Identification of a user is performed upon an authentication request based on the currently selected context and: • for AES-based symmetric mutual authentication: the key number. For example, if an authentication request for key number 0 is issued after selecting a specific

[70] [selection: *detect, prevent*]
[71] [assignment: *list of multiple authentication mechanisms*]
[72] [assignment: *rules describing how the multiple authentication mechanisms provide authentication*]

application, the user is identified as the Application Manager of the respective application.

- for ECC-based mutual and reader-unilateral authentication: the access rights granted from the targeted CARootKey and presented certificates. For example, if an authentication request, issued after selecting a specific application, targets a CARootKey that is associated with ACMap where bit 0 is set, and also the presented certificates either implicitly inherit or have this access right explicitly encoded, the user is identified as the Application Manager of the respective application.

Before any authentication request is issued the user is identified as "Everybody".

6.1.3.13 FIA_API.1/ECDSA

The TOE shall meet the requirement "Authentication Proof of Identity (ECDSA)" as specified below.

FIA_API.1/ECDSA	**Authentication Proof of Identity (ECDSA)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_API.1.1/ECDSA	The TSF shall provide a *generic ECDSA signature functionality*[73] to prove the identity of the *TOE*[74].

6.1.3.14 FIA_API.1/InternAuth

The TOE shall meet the requirement "Authentication Proof of Identity (ISOInternalAuthenticate)" as specified below.

FIA_API.1/InternAuth	**Authentication Proof of Identity (ISOInternalAuthenticate)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_API.1.1/InternAuth	The TSF shall provide a *ECC-based Card Unilateral Authentication*[75] to prove the identity of the *TOE*[76].

6.1.3.15 FMT_REV.1

The TOE shall meet the requirement "Revocation" as specified below.

FMT_REV.1	**Revocation**

[73] [assignment: *authentication mechanism*]

[74] [assignment: *authorized user or role*]

[75] [assignment: *authentication mechanism*]

[76] [assignment: *authorized user or role*]

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles
FMT_REV.1.1	The TSF shall restrict the ability to revoke *certificates*[77] associated with the *individual subjects as identified by the certificates granted by the AppCA or AdminCA*[78] under the control of the TSF to *an AppUser or PICCUser with Write or ReadWrite access to the CRLFile*[79].
FMT_REV.1.2	The TSF shall enforce the rules:[80] • *Depending on the configuration, revocation updates will only be allowed if signed by respectively the AppCA at Application level, or AdminCA at PICC level.* • *Depending on the configuration, revocation updates will only be allowed if the CRLVersion is strictly increasing or strictly increasing by one.* • *Revocation updates will revoke single subjects or groups of subjects via their individual Certificate Serial Number (CSN) or ranges of consecutive CSNs.*

6.1.3.16 FPT_TDC.1

The TOE shall meet the requirement "Inter-TSF basic TSF data consistency" as specified below.

FPT_TDC.1	**Inter-TSF basic TSF data consistency**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_TDC.1.1	The TSF shall provide the capability to consistently interpret *data files and values*[81] when shared between the TSF and another trusted IT product.
FPT_TDC.1.2	The TSF shall use *the following rules:*[82] • *data files or values can only be modified by their dedicated type-specific operations honouring the type-specific boundaries.* when interpreting the TSF data from another trusted IT product.

6.1.3.17 FTP_TRP.1

The TOE shall meet the requirement "Trusted path" as specified below.

FTP_TRP.1	**Trusted path**

[77] [assignment: *list of security attributes*]
[78] [selection: *users, subjects, objects, [assignment: other additional resources]*]
[79] [assignment: *the authorised identified roles*]
[80] [assignment: *specification of revocation rules*]
[81] [assignment: *list of TSF data types*]
[82] [assignment: *list of interpretation rules to be applied by the TSF*]

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_TRP.1.1 The TSF shall provide a communication path between itself and *remote*[83] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from *modification, disclosure, or only modification*[84].

FTP_TRP.1.2 The TSF shall permit *remote users*[85] to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for *authentication requests with AES or ECC, confidentiality and/or integrity verification for data transfers protected with AES based on a setting in the file attributes*[86].

6.1.3.18 FCO_NRO.1

The TOE shall meet the requirement "Selective proof of origin" as specified below.

FCO_NRO.1 **Selective proof of origin**

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FCO_NRO.1.1 The TSF shall be able to generate evidence of origin for transmitted *transactions*[87] at the request of the *AppMgr*[88].

FCO_NRO.1.2 The TSF shall be able to relate the *identity*[89] of the originator of the information, and the *commands executed during the transaction*[90] of the information to which the evidence applies.

FCO_NRO.1.3 The TSF shall provide a capability to verify the evidence of origin of information to *the AppMgr or other 3rd parties*[91] given *that the AppMgr has the AppECCPrivateKey configured for Transaction Signature generated by the TOE, or ensures confidentiality of this key*[92].

[83] [selection: *remote, local*]
[84] [selection: *modification, disclosure, [assignment: other types of integrity or confidentiality violation]*]
[85] [selection: *the TSF, local users, remote users*]
[86] [selection: *initial user authentication, [assignment: other services for which trusted path is required]*]
[87] [assignment: *list of information types*]
[88] [selection: *originator, recipient, [assignment: list of third parties]*]
[89] [assignment: *list of attributes*]
[90] [assignment: *list of information fields*]
[91] [selection: *originator, recipient, [assignment: list of third parties]*]
[92] [assignment: *limitations on the evidence of origin*]

6.1.4 Security Functional Requirements regarding Robustness

6.1.4.1 FDP_ROL.1

The TOE shall meet the requirement "Basic rollback" as specified below.

FDP_ROL.1	**Basic rollback**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FDP_ROL.1.1	The TSF shall enforce *TOE Access Control Policy*[93] to permit the rollback of the *operations that modify the value or data file objects*[94] on the *backup files*[95].
FDP_ROL.1.2	The TSF shall permit operations to be rolled back within the *scope of the current transaction, which is defined by the following limitative events: chip reset, select command, deselect command, explicit commit, explicit abort, command failure*[96].

6.1.4.2 FPR_UNL.1

The TOE shall meet the requirement "Unlinkability" as specified below.

FPR_UNL.1	**Unlinkability**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPR_UNL.1.1	The TSF shall ensure that *unauthorised subjects other than the card holder*[97] are unable to determine whether *any operation of the TOE*[98] *were caused by the same user*[99].

6.1.4.3 FPT_RPL.1

The TOE shall meet the requirement "Replay detection" as specified below.

FPT_RPL.1	**Replay detection**
Hierarchical to:	No other components.

[93] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[94] [assignment: *list of operations*]
[95] [assignment: *information and/or list of objects*]
[96] [assignment: *boundary limit to which rollback may be performed*]
[97] [assignment: *set of users and/or subjects*]
[98] [assignment: *list of operations*]
[99] [selection: *were caused by the same user, are related as follows[assignment: list of relations]*]

Dependencies:	No dependencies.
FPT_RPL.1.1	The TSF shall detect replay for the following entities: *authentication requests with AES or ECC, confidentiality and/or data integrity verification for data transfers protected with AES and based on a setting in the file attributes*[100].
FPT_RPL.1.2	The TSF shall perform *rejection of the request*[101] when replay is detected.

6.1.5 Security Functional Requirements regarding Secure Dynamic Messaging

6.1.5.1 FDP_ETC.3

The TOE shall meet the requirement "Export of user data in unauthenticated state" as specified below.

FDP_ETC.3	**Export of user data in unauthenticated state**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_ETC.3.1	The TSF shall export the following pieces of user data: *a configurable subset of file data*[102] with the following user data's associated security attributes: *confidentiality, authenticity and replay protection for the configurable subset of the file data*[103].
FDP_ETC.3.2	The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data.
FDP_ETC.3.3	The TSF shall enforce the following rules when user data is exported from the TOE: *plain export of file data in case that Secure Dynamic Messaging is not activated for the file*[104].

6.2 Security Assurance Requirements

The following table lists all security assurance components that are valid for this Security Target.

Table 17. Security Assurance Requirements

Name	Title
ADV_ARC.1	Security architecture description
ADV_FSP.5	Complete semi-formal functional specification with additional error information
ADV_IMP.2	Complete mapping of the implementation representation of the TSF
ADV_INT.3	Minimally complex internals

[100] [assignment: *list of identified entities*]
[101] [assignment: *list of specific actions*]
[102] [assignment: *pieces of user data*]
[103] [assignment: *list of security attributes*]
[104] [assignment: *additional exportation control rules*]

Table 17. Security Assurance Requirements*...continued*

Name	Title
ADV_SPM.1	Formal TOE security policy model
ADV_TDS.5	Complete semiformal modular design
AGD_OPE.1	Operational user guidance
AGD_PRE.1	Preparative procedures
ALC_CMC.5	Advanced support
ALC_CMS.5	Development tools CM coverage
ALC_DEL.1	Delivery procedures
ALC_DVS.2	Sufficiency of security measures
ALC_LCD.1	Developer defined life-cycle model
ALC_TAT.3	Compliance with implementation standards - all parts
ASE_INT.1	ST introduction
ASE_CCL.1	Conformance claims
ASE_SPD.1	Security problem definition
ASE_OBJ.2	Security objectives
ASE_ECD.1	Extended components definition
ASE_REQ.2	Derived security requirements
ASE_TSS.2	TOE summary specification with architectural design summary
ATE_COV.3	Rigorous analysis of coverage
ATE_DPT.3	Testing: modular design
ATE_FUN.2	Ordered functional testing
ATE_IND.2	Independent testing - sample
AVA_VAN.5	Advanced methodical vulnerability analysis

In the set of assurance components chosen for EAL6, only ADV_SPM.1 requires an assignment. This assignment is given below.

ADV_SPM.1 **Formal TOE security policy model**

ADV_SPM.1.1D The developer shall provide a formal security policy model for the *following SFRs:*[105]

- *TOE Access Control Policy: FDP_ACC.1, FDP_ACF.1, FDP_ITC.2, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_SMF.1, FMT_SMR.1*

ADV_SPM.1.2D For each policy covered by the formal security policy model, the model shall identify the relevant portions of the statement of SFRs that make up that policy.

ADV_SPM.1.3D The developer shall provide a formal proof of correspondence between the model and any formal functional specification.

[105] [assignment: *list of policies that are formally modelled*]

ADV_SPM.1.4D	The developer shall provide a demonstration of correspondence between the model and the functional specification.

6.2.1 Refinements of the TOE Security Assurance Requirements

In compliance to Application Note 23 in the Protection Profile [6], this Security Target has to conform to all refinements of the security assurance requirements in the Protection Profile. Because the refinements in the Protection Profile are defined for the security assurance components of EAL4 (augmented by ALC_DVS.2 and AVA_VAN.5), some refinements have to be applied to assurance components of the higher level EAL6 stated in the Security Target.

Most of the security assurance components mentioned in the Protection Profile and in this Security Target have the same component level and therefore for these components the refinements from the Protection Profile are valid for this Security Target without change. The following subsections apply the refinements for the Security Assurance Requirements that are different between the Protection Profile and this Security Target.

6.2.1.1 Refinements regarding ADV_FSP

The refinement in Section 6.2.1.6 of the Protection Profile [6] regarding ADV_FSP.4 addresses the complete representation of the TSF, the purpose and method of use of all TSFIs, and the accuracy and completeness of the SFR instantiations. The refinement is not a change in the wording of the action elements, but a more detailed definition of the items above.

Compared to ADV_FSP.4 component ADV_FSP.5 requires a Functional Specification in a semi-formal style (ADV_ FSP.5.2C). In addition, component ADV_FSP.5 extends the scope of the error messages to be described from those resulting from an invocation of a TSFI (ADV_FSP.5.6C) to also those not resulting from an invocation of a TSFI (ADV_FSP.5.7C). For the latter a rationale shall be provided (ADV_FSP.5.8C).

Since the higher level ADV_FSP.5 only affects the style of description and the scope of and rationale for error messages, the refinement in the Protection Profile regarding ADV_FSP.4 can be applied without changes and is valid for ADV_FSP.5.

6.2.1.2 Refinements regarding ADV_IMP

The refinement in Section 6.2.1.7 of the Protection Profile [6] regarding ADV_IMP.1 states that it must be checked that the provided implementation representation is complete and sufficient to ensure that analysis activities are not curtailed due to lack of information.

This Security Target targets assurance level EAL6 augmented, which requires access to all source code of the TOE so that the above refinement is implicitly fulfilled.

6.2.1.3 Refinements Regarding ALC_CMC

The refinement in Section 6.2.1.4 of the Protection Profile [6] regarding ALC_CMC.4 is a clarification of the 'TOE' and the term 'configuration items'.

Since the higher level ALC_CMC.5 requires a higher assurance regarding the defined TOE and the configuration items, the refinement in the Protection Profile regarding ADV_CMC.4 can be applied without changes and is valid for ADV_CMC.5.

6.2.1.4 Refinements Regarding ALC_CMS

The refinement in Section 6.2.1.3 of the Protection Profile [6] regarding ALC_CMS.4 is a clarification of the configuration item 'TOE implementation representation'.

Compared to ALC_CMS.4 component ALC_CMS.5 only adds the requirement for a new configuration item to be included in the configuration list (ALC_CMS.51C) so that the refinement in the Protection Profile regarding ADV_CMS.4 can be applied without changes and is valid for ADV_CMS.5.

6.2.1.5 Refinements Regarding ATE_COV

The refinement in Section 6.2.1.8 of the Protection Profile [6] regarding ATE_COV.2 defines that test coverage must include different operating conditions and 'ageing' and that existence and effectiveness of countermeasures against physical attacks cannot be tested but must be given by evidence.

The refinement regarding test coverage is not a change in the wording of the action elements, but a more detailed definition of the items to be applied, so that it can be applied without changes and is valid for ATE_COV.3. The refinement regarding existence and effectiveness of countermeasures against physical attacks is implicitly fulfilled since this Security Target targets assurance level EAL6 augmented, which requires access to all source code and layout data.

6.3 Security Requirements Rationale

6.3.1 Rationale for the Security Functional Requirements

Section 6.3.1 in the Protection Profile provides a rationale for the mapping between security functional requirements and security objectives defined in the Protection Profile. This rationale is not repeated here.

This Security Target defines additional SFRs for the TOE. In addition security requirements for the environment are defined. The following table gives an overview, how the requirements are combined to meet the security objectives.

Table 18. Security Functional Requirements mapping to Security Objectives

Name	Title
O.Access-Control	FCS_CKM.4 FDP_ACC.1 FDP_ACF.1 FDP_ITC.2 FMT_MSA.1 FMT_MSA.3 FMT_MTD.1 FMT_SMF.1 FMT_SMR.1
O.Authentication	FCS_COP.1/AES FCS_COP.1/ECDSA FCS_COP.1/ECDH FCS_COP.1/SHA FCS_CKM.1/Session_AES FCS_CKM.1/Session_ECC FIA_API.1/ECDSA FIA_API.1/InternAuth FIA_UID.2 FIA_UAU.2 FIA_UAU.3 FIA_UAU.5 FMT_REV.1 FMT_SMF.1

Table 18. Security Functional Requirements mapping to Security Objectives*...continued*

Name	Title
	FMT_SMR.1 FPT_RPL.1 FTP_TRP.1
O.Encryption	FCS_CKM.1/Session_AES FCS_CKM.1/Session_ECC FCS_CKM.4 FCS_COP.1/AES FTP_TRP.1 FDP_ETC.3
O.Integrity	FCO_NRO.1 FCS_CKM.1/ECC FCS_CKM.1/Session_AES FCS_CKM.1/Session_ECC FCS_CKM.4 FCS_COP.1/AES FCS_COP.1/ECDSA FCS_COP.1/SHA FPT_RPL.1 FTP_TRP.1 FDP_ETC.3
O.Type-Consistency	FPT_TDC.1
O.Transaction	FDP_ROL.1
O.No-Trace	FPR_UNL.1

Justification related to Access Control (O.Access-Control)

The SFR FMT_SMR.1 defines the roles of the Access Control Policy. The SFR FDP_ACC.1 and FDP_ACF.1 define the rules and FMT_MSA.3 and FMT_MSA.1 the attributes that the access control is based on. FMT_MTD.1 provides the rules for the management of the authentication data. The management functions are defined by FMT_SMF.1.

Since the TOE stores data on behalf of the authorised subjects import of user data with security attributes is defined by FDP_ITC.2.

Since cryptographic keys are used for authentication (refer to O.Authentication), these keys have to be removed if they are no longer needed for the access control (e.g. an application is deleted). This is required by FCS_CKM.4.

These SFRs together provide an access control mechanism as required by the objective O.Access-Control.

Justification related to Authentication (O.Authentication)

For symmetric authentication, FCS_COP.1/AES requires that the TOE provides the basic cryptographic algorithm that can be used to perform the authentication. The SFR FCS_CKM.1/Session_AES generates the session keys used after the authentication.

For asymmetric authentication, the basic cryptographic algorithms are provided by FCS_COP.1/ECDSA, FCS_COP.1/ECDH, FCS_COP.1/SHA, FCS_COP.1/AES and the session keys to be used during and after the authentication are generated by FCS_CKM.1/Session_ECC.

The SFR FIA_UID.2, FIA_UAU.2 and FIA_UAU.5 together define that users must be identified and authenticated before any action. This authentication also associates users with the roles as defined in

FMT_SMR.1. The SFR FIA_UAU.3 prevents that forged authentication data can be used. The "none" authentication of FIA_UAU.5 also ensures that a specific subject is identified and authenticated before an explicit authentication request is sent to the TOE. For the asymmetric authentication, FMT_REV.1 allows to revoke the capability to authentication for individual subjects.FMT_SMF.1 defines security management functions the TSF shall be capable to perform. FTP_TRP.1 requires a trusted communication path between the TOE and remote users, FTP_TRP.1.3 especially requires "authentication requests". Together with FPT_RPL.1 which requires a replay detection for these authentication requests, these SFRs fulfill the objective O.Authentication.

Justification related to Confidential Communication (O.Encryption)

The SFR FCS_COP.1/AES requires that the TOE provides the basic cryptographic algorithm AES that can be used to protect the communication by encryption. FTP_TRP.1 requires a trusted communication path between the TOE and remote users, FTP_TRP.1.3 especially requires "confidentiality and/or data integrity verification for data transfers protected with AES and based on a setting in the file attributes".

The SFRs FCS_CKM.1/Session_AES and FCS_CKM.1/Session_ECC generates the session key used for encryption. FCS_CKM.4 requires that cryptographic keys used for encryption have to be removed after usage.

The TOE also provides Secure Dynamic Messaging service which allows encrypted data to be read without being in the authenticated state. FDP_ETC.3 requires confidential user data export in unauthenticated state, and hence models the requirements to reach O.Encryption.

Justification related to Integrity-protected Communication (O.Integrity)

The SFR FCS_COP.1/AES requires that the TOE provides the basic cryptographic algorithms that can be used to compute a MAC which can protect the integrity of the communication. FCS_COP.1/SHA and FCS_COP.1/ECDSA provide the algorithms for signature calculation and validation. FTP_TRP.1 requires a trusted communication path between the TOE and remote users, FTP_TRP.1.3 especially requires "confidentiality and/or data integrity verification for data transfers on request of the file owner". The SFRs FCS_CKM.1/Session_AES and FCS_CKM.1/Session_ECC generate the session keys used for the MAC calculation. FCS_CKM.1/ECC generates the static key used for the calculation of signatures. FCS_CKM.4 requires that cryptographic keys used for MAC or signature operations can be removed after usage. FPT_RPL.1 requires a replay detection for these data transfers.

The TOE provides a mechanism to prove the authenticity and integrity of complete transactions executed with the TOE to a third party (e.g. a back-end system). This can be done via a Transaction MAC or signature, where the latter is fulfilling FCO_NRO.1.

The TOE also provides Secure Dynamic Messaging service which allows MACed or signed data to be read without being in the authenticated state. FDP_ETC.3 requires user data export in unauthenticated state, and hence models the requirements to reach O.Integrity.

Justification related to Data type consistency (O.Type-Consistency)

The SFR FPT_TDC.1 requires the TOE to consistently interpret data files and values. The TOE will honor the respective file formats and boundaries (i.e. upper and lower limits, size limitations). This meets the objective O.Type-Consistency.

Justification related to Transaction mechanism (O.Transaction)

The SFR FDP_ROL.1 requires the possibility to rollback a set of modifying operations on backup files in total. The set of operations is defined by the scope of the transaction, which is itself limited by some boundary events. This fulfils the objective O.Transaction.

Justification related to Preventing Traceability (O.No-Trace)

The SFR FPR_UNL.1 requires that unauthorised subjects other than the card holder are unable to determine whether any operation of the TOE were caused by the same user. This meets the objective O.No-Trace.

6.3.2 Dependencies of Security Functional Requirements

The dependencies listed in the Protection Profile are independent of the additional dependencies listed in the table below. The dependencies of the Protection Profile are fulfilled within the Protection Profile and at least one dependency is considered to be satisfied. The following discussion demonstrates how the SFR dependencies (defined by Part 2 of the Common Criteria [3]) satisfy the requirements specified in Section 6.1.

The dependencies and their fulfillment are listed in the tables below:

Table 19. Dependencies of Security Functional Requirements (PP-0084)

SFR	Dependency	Fulfilled in ST
FAU_SAS.1	No dependencies.	No dependency
FCS_RNG.1/PTG2	No dependencies.	No dependency
FCS_RNG.1/DRG4	No dependencies.	No dependency
FDP_ITT.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Yes
FDP_IFC.1	FDP_IFF.1 Simple security attributes	See discussion in the PP
FDP_SDC.1	No dependencies.	No dependency
FDP_SDI.2	No dependencies.	No dependency
FMT_LIM.1	FMT_LIM.2 Limited availability.	Yes
FMT_LIM.2	FMT_LIM.1 Limited capabilities.	Yes
FPT_FLS.1	No dependencies.	No dependency
FPT_ITT.1	No dependencies.	No dependency
FPT_PHP.3	No dependencies.	No dependency
FRU_FLT.2	FPT_FLS.1 Failure with preservation of secure state.	Yes

Table 20. Dependencies of Security Functional Requirements (Security Target)

SFR	Dependency	Fulfilled in ST
FCO_NRO.1	FIA_UID.1 Timing of identification	Yes, by FIA_UID.2
FCS_CKM.1/Session_AES	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_ CKM.4 Cryptographic key destruction	Yes, by FCS_COP.1/AES, FCS_CKM.4
FCS_CKM.1/Session_ECC	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_ CKM.4 Cryptographic key destruction	Yes, by FCS_COP.1/ECDH, FCS_CKM.4
FCS_CKM.1/ECC	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_ CKM.4 Cryptographic key destruction	Yes, by FCS_COP.1/ECDH, FCS_COP.1/ECDSA, FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	Yes, by FDP_ITC.2, FCS_CKM.1/Session_AES, FCS_CKM.1/Session_ECC , FCS_CKM.1/ECC
FCS_COP.1/AES	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction	Yes, by FDP_ITC.2, FCS_CKM.1/Session_AES, FCS_CKM.4.

Table 20. Dependencies of Security Functional Requirements (Security Target)*...continued*

SFR	Dependency	Fulfilled in ST
FCS_COP.1/ECDH	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction	Yes, by FDP_ITC.2, FCS_CKM.1/ECC, FCS_CKM.4.
FCS_COP.1/ECDSA	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction	Yes, by FDP_ITC.2, FCS_CKM.1/ECC, FCS_CKM.4.
FCS_COP.1/SHA	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction	As no key is used, there is no need for key-related dependencies.
FDP_ACC.1	FDP_ACF.1 Security attribute based access control	Yes, by FDP_ACF.1.
FDP_ACF.1	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation	Yes, by FDP_ACC.1, FMT_MSA.3
FDP_ITC.2	[FDP_ACC.1 Subset access control, or FDP_ IFC.1 Subset information flow control], [FTP_ ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path], FPT_TDC.1 Inter-TSF basic TSF data consistency	Yes, by FDP_ACC.1, FTP_TRP.1, FPT_TDC.1.
FDP_ROL.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Yes, by FDP_ACC.1.
FDP_ETC.3	No dependencies.	No dependency.
FIA_API.1/ECDSA	No dependencies.	No dependency.
FIA_API.1/InternAuth	No dependencies.	No dependency.
FIA_UAU.2	FIA_UID.1 Timing of identification	Yes, by FIA_UID.2.
FIA_UAU.3	No dependencies.	No dependency.
FIA_UAU.5	No dependencies.	No dependency.
FIA_UID.2	No dependencies.	No dependency.
FMT_MSA.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions	Yes, by FDP_ACC.1, FMT_SMR.1, FMT_SMF.1.
FMT_MSA.3	FMT_MSA.1 Management of security attributes, FMT_SMR.1 Security roles	Yes, by FMT_MSA.1, FMT_SMR.1.
FMT_MTD.1	FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions	Yes, by FMT_SMR.1, FMT_SMF.1.
FMT_REV.1	FMT_SMR.1 Security roles	Yes, by FMT_SMR.1.
FMT_SMF.1	No dependencies.	No dependency.
FMT_SMR.1	FIA_UID.1 Timing of identification	Yes, by FIA_UID.2.
FPR_UNL.1	No dependencies.	No dependency.
FPT_RPL.1	No dependencies.	No dependency.
FPT_TDC.1	No dependencies.	No dependency.
FTP_TRP.1	No dependencies.	No dependency.

6.3.3 Rationale for the Assurance Requirements

The selection of assurance components is based on the underlying Protection Profile [6]. The Security Target uses the same augmentations as the PP (and the addition of augmentation ASE_TSS.2), but chooses a higher assurance level. The level EAL6 is chosen in order to meet assurance expectations of access control applications and automatic fare collection systems. Additionally, the requirement of the PP to choose at least EAL4 is fulfilled.

The rationale for the PP augmentations is the same as in the PP. The assurance level EAL6 is an elaborated pre-defined level of the CC Part 3 [4]. The assurance components in an EAL level are chosen in a way that they build a mutually supportive and complete set of components. The requirements chosen for augmentation do not add any dependencies, which are not already fulfilled for the corresponding requirements contained in EAL6. Therefore, these components add additional assurance to EAL6, but the mutual support of the requirements is still guaranteed.

As stated in the Section 6.3.3 of the Protection Profile [6], the TOE is intended to defend against sophisticated attacks. Therefore specifically AVA_VAN.5 was chosen by the PP in order to assure that even attackers with high attack potential cannot successfully attack the TOE.

In addition to the SARs introduced by EAL6, ASE_TSS.2 was chosen as augmentation to include architectural information on the security functionality of the TOE in the ST.

6.3.4 Security Requirements are Internally Consistent

The discussion of security functional requirements and assurance components in the preceding sections has shown that mutual support and consistency are given for both groups of requirements. The arguments given for the fact that the assurance components are adequate for the functionality of the TOE also show that the security functional and assurance requirements support each other and that there are no inconsistencies between these groups.

The security functional requirements required to meet the security objectives O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation and O.Leak-Forced also protect the cryptographic algorithms and the access control function used to implement the Access Control Policy. The security objectives defined in the Protection Profile can be seen as "low-level protection" objectives, while the additional security objectives defined in this Security Target are "high-level protection" objectives. For example, O.Encryption states that the communication can be protected by encryption. While this ensures the rather high-level goal that the communication can not be eavesdropped, the overall goal that the communication is confidential is ensured with the help of the Protection Profile objective that prevent attacks on the key and the cryptographic implementation like probing or fault injection attacks.

7 TOE Summary Specification

7.1 Portions of the TOE Security Functionality

The TOE Security Functionality (TSF) directly corresponds to the TOE security functional requirements defined in Section 6. The table below lists the TSF of the TOE.

Table 21. Portions of the TSF

TSF portion	Title	Description
TSF.Service	Service functionality supporting other TSF	This portion of the TSF comprises services like random number generation and provides mechanisms to store initialization, pre-personalization, and/or other data on the TOE.
TSF.Protection	General security measures to protect the TSF	This portion of the TSF comprises physical and logical protection to avoid information leakage and detect fault injection. It defines resets in case an error or attack was detected.
TSF.Control	Operating conditions, memory and hardware access control	This portion of the TSF controls the operating conditions.
TSF.Authentication	Mutual Authentication	This portion of the TSF provides a mutual authentication mechanism to separate authorized subjects from unauthorized subjects.
TSF.Access-Control	Access Control	This portion of the TSF provides an access control mechanism to the subjects, objects, operations and attributes defined by the TOE Access Control Policy.
TSF.Encryption	Encryption	This portion of the TSF provides cryptographic operations to protect communication against eavesdropping.
TSF.Integrity	Integrity-Protected Communication	This portion of the TSF allows both the TOE and the terminal to detect integrity violations, replay or man-in-the-middle attacks.
TSF.Transaction	Transaction	This portion of the TSF ensures that either all or none of the commands within an transaction are performed.
TSF.No-Trace	Preventing Traceability	This portion of the TSF prevents tracing of the TOE by e.g. simply retrieving its UID.

The TSF are described in more detail in the following sections and the relation to the security functional requirements is shown.

7.2 TOE Summary Specification Rationale

7.2.1 Mapping of Security Functional Requirements and TOE Security Functionality

SFR	TSF.Service	TSF.Protection	TSF.Control	TSF.Access-Control	TSF.Authentication	TSF.Encryption	TSF.Integrity	TSF.Transaction	TSF.No-Trace	Description
Security Functional Requirements from the Protection Profile										
FRU_FLT.2			X							Limited fault tolerance
FPT_FLS.1			X							Failure with preservation of secure state
FMT_LIM.1			X							Limited capabilities
FMT_LIM.2			X							Limited availability
FAU_SAS.1	X									Audit storage
FDP_SDC.1		X								Stored data confidentiality
FDP_SDI.2		X								Stored data integrity monitoring and action
FPT_PHP.3		X								Resistance to physical attack
FDP_ITT.1		X								Basic internal transfer protection
FPT_ITT.1		X								Basic internal TSF data transfer protection
FDP_IFC.1		X								Subset information flow control
FCS_RNG.1/PTG2	X									Random number generation (Class PTG.2)
FCS_RNG.1/DRG4	X									Random number generation (Class DRG.4)
Security Functional Requirements regarding Access Control										
FDP_ACC.1				X						Subset access control
FDP_ACF.1				X						Security attribute based access control
FDP_ITC.2				X						Import of user data with security attributes
FMT_MSA.1				X						Management of security attributes
FMT_MSA.3				X						Static attribute initialization
FMT_MTD.1				X						Management of TSF data
FMT_SMF.1				X	X					Specification of Management Functions
FMT_SMR.1				X	X					Security roles
Security Functional Requirements regarding Confidentiality, Authentication and Integrity										
FCS_COP.1/AES					X	X	X			Cryptographic Operation (AES)
FCS_COP.1/ECDSA					X		X			Cryptographic Operation (ECDSA)
FCS_COP.1/ECDH					X					Cryptographic Operation (ECDH)
FCS_COP.1/SHA					X		X			Cryptographic Operation (SHA)
FCS_CKM.1/Session_AES					X	X	X			Cryptographic key generation (Session AES)

SFR	TSF.Service	TSF.Protection	TSF.Control	TSF.Access-Control	TSF.Authentication	TSF.Encryption	TSF.Integrity	TSF.Transaction	TSF.No-Trace	Description
FCS_CKM.1/Session_ECC					X	X	X			Cryptographic key generation (Session ECC)
FCS_CKM.1/ECC							X			Cryptographic key generation (ECC)
FCS_CKM.4				X		X	X			Cryptographic key destruction
FIA_UAU.2					X					User authentication before any action
FIA_UAU.3					X					Unforgeable authentication
FIA_UAU.5					X					Multiple authentication mechanisms
FIA_UID.2					X					User identification before any action
FIA_API.1/ECDSA					X					Authentication Proof of Identity (ECDSA)
FIA_API.1/InternAuth					X					Authentication Proof of Identity (ISOInternalAuthenticate)
FMT_REV.1					X					Revocation
FPT_TDC.1				X						Inter-TSF basic TSF data consistency
FTP_TRP.1					X	X	X			Trusted path
FCO_NRO.1							X			Selective proof of origin
Security Functional Requirements regarding Robustness										
FDP_ROL.1								X		Basic rollback
FPR_UNL.1									X	Unlinkability
FPT_RPL.1					X		X			Replay detection
Security Functional Requirements regarding Secure Dynamic Messaging										
FDP_ETC.3						X	X			Export of user data in unauthenticated state

7.2.2 TSF.Service

TSF.Service provides the following functionality:

TOE identification

FAU_SAS.1 is implemented by a test function that allows to store identification and/or pre-personalization data (including a unique ID for each die) for the TOE in the non-volatile memory (NVM) at the end of the tests in Phase 3.

Random Number Generation

The TOE provides a hardware (physical) random number generator (RNG) according to PTG.2 as described in [1]. The physical RNG comprises a hardware and software test functionality to detect faults in the circuitry of the RNG (total failure test). Therefore this functionality meets FCS_RNG.1/PTG2.

The TOE also provides a hybrid deterministic RNG according to DRG.4 as described in [1]. This functionality therefore meets FCS_RNG.1/DRG4. This hybrid deterministic RNG is seeded by the hardware (physical) PTG.2 RNG and is responsible for providing random numbers for the cryptographic protocols.

7.2.3 TSF.Protection

TSF.Protection addresses functionalities of the TOE which are used to protect the TSF, TSF data and user data from any kind of attack. Its functionality mainly addresses self-protection of the TSF. However, TSF.Protection also addresses non-bypassability as it implements logical protection to avoid information leakage. TSF.Protection provides the following functionality:

Integrity protection of memories

As required by FDP_SDI.2, TSF.Protection supports the integrity of the ROM, RAM and NVM. The NVM is able to perform error correction. The ROM, RAM and NVM provide parity protection.

Furthermore, TSF.Protection also implements integrity protection during start-up. TSF.Protection supports all other SFRs because prevention of successful manipulation of security functionality is a pre-condition for the reliable work of all other functions.

Protection against physical manipulations

TSF.Protection protects the TOE against physical manipulation. In case a manipulation is detected, a reset is triggered to return to a secure state. Therefore, TSF.Protection implements FPT_PHP.3.

The aspect of TSF.Protection is further supported by FPT_FLS.1 which controls the environmental conditions and triggers a reset in case these are out of bounds.

Logical protection

TSF.Protection prevents the reconstruction of TOE internal information that can be found by analysis of external measured signals like power or clock. Within the different components of the TOE dedicated functions are implemented to sufficiently limit or eliminate the information that might be contained in the shape and amplitude of signals or in the time between events.

Logical protections implemented by TSF.Protection covers the SFRs FDP_ITT.1, FPT_ITT.1 and FDP_IFC.1. They cannot be influenced from outside the TOE.

In addition, TSF.Protection encrypts contents stored in RAM and NVM memory and applies memory address scrambling. This ensures the confidentiality of user data stored in RAM and NVM memory as required by FDP_SDC.1.

Cryptographic co-processors and cryptographic library

The cryptographic co-processors (AES, ECC) as well as the cryptographic library implement countermeasures against fault injection and information leakage. Another implemented mechanism to protect User Data from unwanted disclosure is an automatic clean-up of relevant registers (key and data registers of the used coprocessor) after usage and before changing the TOE mode. Therefore, all FCS_COP.1 and FCS_CKM.4 iterations indirectly support TSF.Protection.

7.2.4 TSF.Control

TSF.Control addresses those aspects the TSF controls, e.g., the secure operating conditions or access to specific memory addresses. Its functionality mainly addresses non-bypassability of the TSF. TSF.Control provides the following functionality.

Control of operating conditions

TSF.Control ensures the correct operation of the TOE hardware (functions offered by the micro-controller including the standard CPU, the cryptographic coprocessors, the memories, registers, I/O interfaces and the other system peripherals) during the execution of the IC Dedicated Support Software and Security IC Embedded Software. For this the TOE comprises filters for power supply and clock input. In addition, TSF.Control controls the allowed secure range of temperature, clock frequency, voltage and light.

The filters support the correct function of the TOE within the limits of the secure operating conditions. This robustness implements FRU_FLT.2 and ensures that the processing is performed without failure that may be caused by interference of any external communication interface or other external influences.

FPT_FLS.1 is implemented by sensors that limit the temperature, clock frequency, and voltage to a secure upper and lower threshold. These sensors detect whether the TOE is operating outside its specified secure range. Light sensors distributed over the chip surface detect abnormal light intensities. The secure state required by FPT_FLS.1 is realized by an internal reset of the TOE.

Mode control

TSF.Control realizes the control within the TOE testing phases (phase 3 of the life-cycle) and afterwards. The life-cycle 'Wafer Test' is available for testing purposes in the phases before TOE delivery and disabled before the TOE is delivered from NXP to the customer.

The test concept with specific hardware operations initiated by the test software cannot be used to read out directly any data stored in one of the memories of the TOE. Therefore the capabilities to abuse the test functions for compromising User Data or TSF data is very limited as required by FMT_LIM.1.

At the end of the wafer test the access to the IC Dedicated Test Software is disabled. TSF.Control ensures that it is not possible to switch back and reuse the test functions again. In addition, the test functions of the IC Dedicated Test Software require a special sequence to execute a dedicated test routine. Therefore, TSF.Control limits the availability of the test functions as stated by FMT_LIM.2.

7.2.5 TSF.Authentication

TSF.Authentication provides an authentication mechanism to separate authorised subjects from unauthorised subjects. The authentication of subjects is performed by either a challenge-response-based mutual authentication protocol using symmetric cryptography, or an asymmetric authenticated Diffie-Hellman key agreement protocol, supporting both mutual and reader-unilateral authentication. The TOE supports the cryptographic algorithms AES (128 and 256 bits) for the symmetric authentication and ECC (256 bits) for the asymmetric authentication. By this TSF.Authentication meets FCS_COP.1/AES, FCS_COP.1/ECDSA, FCS_COP.1/ECDH, FCS_COP.1/SHA, FCS_CKM.1/Session_AES and FCS_CKM.1/Session_ECC.

TSF.Authentication also identifies the user to be authenticated by the currently selected context (card or specific application) and the key number. This meets FIA_UID.2. The cryptographic authentication is used for the *Admin*, *PICCUser*, *AppMgr*, *DelAppMgr*, *AppUser*, *AppChangeUser*, or *AppRollUser*. Since the TOE can be used without authentication the "none" authentication is used to "authenticate" *Anybody*. Therefore it implements FIA_UAU.2, FIA_UAU.5 and FMT_SMR.1.

The symmetric authentication protocol requires the user to proof knowledge of a secret key by applying it on a freshly generated random challenge, generated to the TOE. The asymmetric authentication protocol requires the user to proof knowledge of a private key by applying it on the public key of a freshly generated ephemeral key pair used for the key agreement. This ensures that the authentication requests itself cannot be forged or circumvented by attacks like replay or man-in-the-middle. Therefore these protocols meet FIA_UAU.3 and the relevant parts of FTP_TRP.1 and FPT_RPL.1 with respect to the authentication requests.

In the context of the asymmetric authentication protocol, the TOE allows to deny-list individual subjects based on their certificate via a certificate revocation mechanism. This implements FMT_REV.1.

Authentication of a user is initiated by an authentication request and the authentication state is reset if one of the following events occurs: selecting an application or the card, changing the key corresponding to the current authentication, occurrence of any error during the execution of a command, starting a new authentication, rolling a key set, failed proximity check, deleting an *Application* as *AppMgr*, and reset. By this FMT_SMF.1 is also implemented.

The authentication functionality also provides an authentication mechanism to authenticate the TOE. While this is also provided by the mutual authentication mechanisms discussed in the previous section, the TOE also supports TOE-unilateral authentication mechanisms. These mechanisms are based on asymmetric cryptography and do not require any secret key material in the terminal. This method can also be used for originality checking, verifying the authenticity of the TOE immediately after manufacturing, i.e. before further personalization. In this case, one relies on a key pair and certificate injected in the TOE during manufacturing.

For TOE-unilateral authentication, the TOE implements a dedicated unilateral authentication protocol, but also provides generic ECDSA signature support. With this, FIA_API.1/InternAuth and FIA_API.1/ECDSA are implemented.

7.2.6 TSF.Access-Control

TSF.Access-Control provides an access control mechanism to the objects and Security Attributes that are part of the TOE Access Control Policy. The access control mechanism assigns subjects - (possibly multiple) *AppUser* or *PICCUser* - to 4 different groups of operations on *Files*. The operations on *Files* are File.Read, File.Write, File.ReadWrite and File.Change. One subject can be assigned to each group of *File* operations. The special subjects *Anybody* and *Nobody* can also be assigned. Therefore this functionality maintains the roles as required by FMT_SMR.1.

Since TSF.Access-Control also maintains the objects and Security Attributes as stated in the TOE Access Control Policy, it also implements FDP_ACC.1, FDP_ACF.1 and FMT_MSA.1. Management of authentication data is necessary to separate the roles, therefore it also implements FMT_MTD.1.

The primary use of the TOE is storage of data on behalf of the authorised users. The rules for data storage are defined by the TOE Access Control Policy. The storage of data is an import of data with security attributes, therefore FDP_ITC.2 is also implemented. This applies to the operations *File.Create* and *File.Delete* on the object *File* within *Applications*.

For the card the operations are *Application.Create* and *Application.Delete* on the object *Application*. If an *Application* is created default Security Attributes are assigned to the *Application*, thereby implementing FMT_MSA.3. If an *Application* is deleted the keys used to authenticate the respective *AppMgr* and *AppUser* are destroyed. This implements FCS_CKM.4.

Additionally, the TOE supports operations to change keys. If keys used to authenticate roles like the AppMgr or AppUser are changed, the existing role instances are replaced by new instances. This implements FCS_CKM.4.

The TOE Access Control Policy also defines the rules for delegated-application support, which allows third party service providers to create their applications onto the issued TOE. After creation, a *DelApplication* has the same attributes as an *Application*.

TSF.Access-Control also controls access to the security attributes. Because it also controls create and delete operations, it implements part of FMT_SMF.1.

Finally the type consistency of the file types stored by the TOE is ensured. It ensures that values can not over- or underflow. Furthermore size limitations of files are obeyed. By this FPT_TDC.1 is implemented.

7.2.7 TSF.Encryption

TSF.Encryption provides a mechanism to protect the communication against eavesdropping by encryption. The encryption is requested by the file owner (i.e. the subject *AppUser* that has the right to perform *File.Change* on a *File*) by setting an option in the attributes of that *File*.

The encryption is using the AES algorithm and by this the functionality implements FCS_COP.1/AES. The SFRs FCS_CKM.1/Session_AES and FCS_CKM.1/Session_ECC generates the session keys used during the encryption, after the symmetric AES-based authentication or asymmetric ECC-based authentication respectively. The SFR FCS_CKM.4 removes the used cryptographic keys after encryption. Note that the encryption functionality is active after an authentication is performed. If an authorised user sets the access control permissions in a way that an object is accessible to *Anybody* (refer to Access Control) this object can be accessed without authentication and therefore also without protection by this functionality.

TSF.Encryption also adds data to the communication stream that enables the terminal to detect integrity violations, replay attacks or man-in-the-middle attacks. If an encrypted communication is requested, it also verifies the data sent by the terminal and returns an error code if such an attack is detected. The detection mechanism covers all frames exchanged between the terminal and the card up to the current encrypted frame. Therefore it can detect any injected/modified frame in the communication before the transfer of the encrypted frame.

The encryption for communication and the information to detect integrity violations implement FTP_TRP.1 with respect to the confidentiality and/or data integrity verification for data transfers on request of the File owner.

When using the Secure Dynamic Messaging functionality, the TOE encrypts a configurable part of the File to be read when required by the File security attributes, therefore implementing FDP_ETC.3.

7.2.8 TSF.Integrity

TSF.Integrity adds data to the communication stream that enables the terminal to detect integrity violations, replay attacks or man-in-the-middle attacks. Vice-versa it verifies the data sent by the terminal and returns an error code if such an attack is detected. When applied on data exchanged after an authentication, it uses the cryptographic algorithm 128-bit AES CMAC. TSF.Integrity therefore implements FCS_COP.1/AES. The SFRs FCS_CKM.1/Session_AES and FCS_CKM.1/Session_ECC generate the session keys used during the calculation, after the symmetric AES-based authentication or asymmetric ECC-based authentication respectively. The SFR FCS_CKM.4 removes the used cryptographic keys after calculation.

The detection mechanism covers all frames exchanged between the terminal and the card up to last frame with a MAC. Depending on the selected mode it can also detect what frame was injected/modified. By this FPT_RPL.1 is implemented.

The information to detect integrity violations implement FTP_TRP.1 with respect to the confidentiality and/or data integrity verification for data transfers on request of the File owner.

The Transaction MAC or signature functionality provides an option to the *AppMgr* to enable an *AppUser* to prove the authenticity of committed transactions on the TOE. In order to do this a MAC or signature is calculated over a committed transaction. FCS_CKM.1/ECC generates the static key used for the calculation of signatures. For the Transaction MAC, AES CMAC is supported, implementing FCS_COP.1/AES, while for the Transaction Signature ECDSA is supported, again implementing FCS_COP.1/ECDSA and FCS_COP.1/SHA and fulfilling FCO_NRO.1.

When using the Secure Dynamic Messaging functionality, the TOE provides a mechanism for integrity protection for the File to be read when required by the File security attributes, therefore implementing FDP_ETC.3. This can be based on an AES CMAC, implementing FCS_COP.1/AES as above, or an ECDSA signature using SHA-256 for hashing, therefore implementing FCS_COP.1/ECDSA and FCS_COP.1/SHA. FCS_CKM.1/ECC generates the key used for this calculation.

7.2.9 TSF.Transaction

TSF.Transaction provides a transaction mechanism that ensures that either all or none of the (modifying) commands within a transaction are performed. The transaction mechanism is active for backup data files, values, linear record files and cyclic record files, it is not active for standard data files. All file types with the exception of "standard data files" are called "backup files" in the following. Note that it is possible to update files in up to 2 applications within one transaction.

This functionality is always active for the respective file types. This means that for every modifying operation with a backup file an explicit commit request must be issued in order to let the modifications take effect.

The following reasons will abort a transaction: Selecting an application or the card, Changing a key, Occurrence of any error during the execution of a command, starting a new authentication, deselection of the virtual card, Rolling a key set, Failed Proximity Check, Deleting an Application and Reset. FDP_ROL.1 is therefore also implemented by this functionality.

7.2.10 TSF.No-Trace

TSF.No-Trace provides an option to the Admin to use a random UID during ISO14443 anti-collision sequence. By this the device cannot be traced any more by simply retrieving its UID. Device specific information can be read out only by the *Admin*, *AppMgr*, *AppChangeUser*, *AppRollUser* and *AppUser* if this option is set.

The card specific information is protected and therefore FPR_UNL.1 is implemented. This functionality does not cover the data in the TOE file system. This data is protected by the TOE Access Control Policy and the tracing protection depends on the access control configuration created by the authorised subjects.

In the default configuration, the TOE is injected with a key pair and certificate for originality checking, i.e. allowing to verify that the TOE was manufactured by the certified manufacturer. This key pair and certificate are shared per production batch, therefore preventing the traceability of individual users. If preferred, this functionality can also be disabled after further personalization, i.e. before distributing the TOE to the device owner in the field.

8 References

8.1 Evaluation documents

[1] A proposal for: Functionality classes for random number generators, Wolfgang Killmann, T-Systems GEI GmbH, Werner Schindler, Bundesamt für Sicherheit in der Informationstechnik (BSI), Version 2.0, 18 September 2011.

[2] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 3.1, Revision 5, CCMB-2017-04-001, April 2017.

[3] Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Version 3.1, Revision 5, CCMB-2017-04-002, April 2017.

[4] Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, Version 3.1, Revision 5, CCMB-2017-04-003, April 2017.

[5] Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Version 3.1, Revision 5, CCMB-2017-04-004, April 2017.

[6] Security IC Platform Protection Profile with Augmentation Packages, Registered and Certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-CC-PP-0084-2014, Version 1.0, 13 January 2014.

[7] Common Criteria Protection Profile, Machine Readable Travel Document with "ICAO Application", Extended Access Control with PACE (EAC PP), BSI-CC-PP-0056-V2-2012, Version 1.3.2, 5 December 2012.

8.2 Developer documents

[8] MF3E(H)x3, MIFARE DUOX contactless smartcard IC, Product data sheet, DocStore number 974410, NXP Semiconductors, Revision 1.0, 31 July 2024.

[9] MF3E(H)x3 PDC, MIFARE DUOX Post Delivery Configuration, Preliminary data sheet addendum, DocStore number 975401, NXP Semiconductors, Revision 0.1, 19 July 2024.

[10] MF3E(c)x3, Information on Guidance and Operation, Guidance and Operation Manual, DocStore number 974812, NXP Semiconductors, Revision 1.2, 1 July 2024.

[11] MF3E(c)x3, Wafer and Delivery Specification, Product Data sheet addendum, DocStore number 974712, NXP Semiconductors, Revision 1.2, 26 July 2024.

8.3 Standards

[12] FIPS PUB 180-4: Secure Hash Standard (SHS), Federal Information Processing Standards Publication, US Department of Commerce/National Institute of Standards and Technology, August 2015.

[13] FIPS PUB 186-5: Digital Signature Standard (DSS), Federal Information Processing Standards Publication, US Department of Commerce/National Institute of Standards and Technology, 3 February 2023.

[14] FIPS PUB 197: Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, US Department of Commerce/National Institute of Standards and Technology, 26 November 2001.

[15] NIST SP 800-38A: Recommendation for Block Cipher Modes of Operation: Methods and Techniques, Morris Dworkin, National Institute of Standards and Technology, December 2001.

[16] NIST SP 800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, Morris Dworkin, National Institute of Standards and Technology, May 2005.

[17] NIST SP 800-56A Revision 3: Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography, National Institute of Standards and Technology, April 2018.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Tables

Figures

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.