SLM10TLD002Y A12 CALYPSO™ move Security Target

Revision: v3.0

Table of Contents

Contents

1 Security Target Introduction (ASE_INT)

1.1 ST reference

The ST has the revision v3.0 and is dated 2022-11-25. The title of this document SLM10TLD002Y A12 CALYPSO™ move Security Target.

1.2 TOE Reference

The ST comprises an Infineon Technologies Security Controller named SLM10TLD002Y design step A12 with CALYPSO™ move software.

The targeted assurance level is EAL2+.

Table 1 Identification

Hardware	Version	Method of identification	Form of delivery
SLM10TLD002Y	Platform ID = 51h (Product name) DSI= 0Ch (corresponds to Design step A12)	"Get Data" command	Wafer (sawn), modules
Software			
CALYPSO™ move software	Version: 02h Revision: 01h	Return after "Select Application"	Part of the devices
User Guidance			
CALYPSO™ move Extended Datasheet	1.1, 2022-07-15	document	Pdf document via secured download[1]
CALYPSO™ move Personalization Guide	1.1, 2022-07-15	document	
Calypso Specification - Calypso Basic	Version 1.1 (Ref. 191011)	document	Pdf upon request from CNA

A customer shall identify the TOE hardware using the "Get Data" command. The CALYPSO™ move software version is returned by the TOE after "Select Application". Guidance about a secure delivery are provided in the user guidance "CALYPSO™ move Extended Datasheet" section 3.3.

1.3 TOE Overview

1.3.1 TOE Definition and Usage

This TOE is a Calypso Basic product, a contactless smartcard, i.e. a portable object (PO) with an ISO/IEC 14443 interface, running a single Calypso Basic application. Such products focus on providing access to public transportation and possibly other associated services that can be combined into a transport title or ticket.

A transport title or ticket is used during a ticket validation or control process. The rights written in the Calypso Basic product's files are checked in order to establish whether the user's entrance to a transport network or access to a delivery of services or goods is allowed.

The TOE consists of smart card ICs (Security Controllers), firmware and user guidance meeting high requirements in terms of performance and security designed by Infineon Technologies AG.

Selected terms used in this document are defined in [PP] section 1.3.2 "Definitions".

For further information see [PP] section 2.1 "TOE Type" and [PP] section 2.2 "TOE Description".

1.3.2 TOE major security features

See [PP] section 2.3 "TOE Major Security Features". Note, that key diversification is not performed by the TOE. The sentence in [PP] "All the cryptographic keys stored in a Calypso Basic product are fully defined and diversified:" has to be considered as an environmental assumption. This is also reflected in the user guidance "CALYPSO™ move Personalization Guide".

[1] Secured download is a way of delivery of documentation and TOE related software using a secure ishare connected to Infineon customer portal. The TOE user needs a DMZ Account to login (authenticate) via the Internet.

1.4 TOE description

This section describes the physical and logical scope of the TOE.

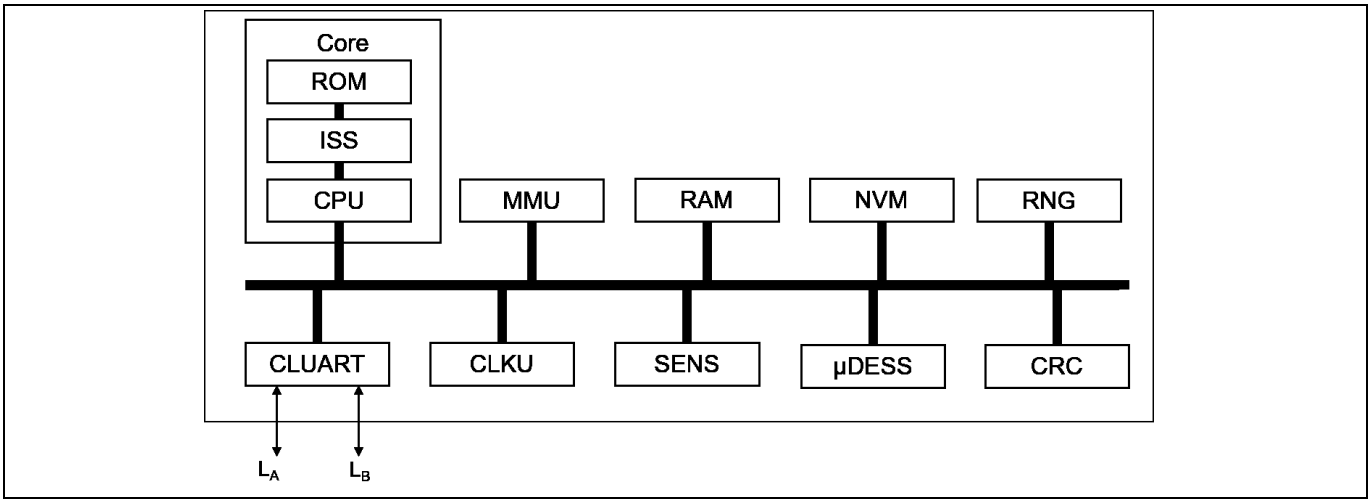
1.4.1 TOE components

The TOE consists of a microcontroller, Test software and CALYPSO™ move application software.

1.4.1.1 Hardware components

Figure 1 shows a simplified block diagram of the TOE hardware:

Figure 1 Block diagram of TOE hardware



The TOE hardware is an IC and consists of a Core and peripherals connected via an internal bus system. The Core consists of ROM, ISS and CPU. The ROM (28 kB) contains the TOE software. The ISS is used for code flow protection over security relevant software parts. The CPU executes the instructions.

The memory consists of RAM and NVM.

The MMU provides efficient access control to all peripherals from a central location. Furthermore it maps the available address spaces of the available physical memory to access conditions in order to prevent unauthorized access.

For generation of random numbers a physical RNG is available, which can be classified as PTG.1 according to [BSI_RNGs].

The CLUART provides an RFID communication interface to/from the terminal. It encodes/decodes ISO14443-2/3 standard signals into/from stream of bytes. The CLUART is capable to work in two different modes: reception mode and transmission mode.

The Clock Unit (CLKU) supplies the clocks for all components of the hardware on either the clock, which is extracted from the electromagnetic field, or based on an internal oscillator.

The purpose of the sensors and filters (SENS) is to monitor certain chip operating conditions. In case the conditions are not tolerable a Security Reset is issued. One of these mechanisms is an active shield as a measure against probing and forcing attacks.

The µDESS is a hardware accelerator dedicated to the TDES calculation. It supports the implementation of software and hardware security measures against power analysis and fault attacks that may be used to reveal a secret key.

For error checking during transmission the CRC follows ISO/IEC 13239. The CRC is further used to protect the integrity of NVM data.

1.4.1.2 Software components

There is only one software component, which combines startup, test and application code. Test code is deactivated before entering pre-personalisation phase.

1.4.1.3 User Guidance components

The user guidance consist of:

- CALYPSO™ move Extended Datasheet: This document describes the SLM10TLD002Y in detail, all its commands and interfaces in Personalisation stage and operation phase.
- CALYPSO™ move Personalization Guide: This user manual provides guidance, how to maintain the targeted security level during personalisation and operation phase.
- [C_BASIC]:Standard Calypso specification for the Calypso Basic product.

1.4.2 Physical scope of the TOE

The physical scope of the TOE is defined by the TOE components described in chapter 1.4.1

1.4.3 Logical scope of the TOE

The logical scope of the TOE consists of the logical security features provided by the TOE as follows:

- Cryptographic support: RNG (True Random Number Generator PTG.1) according to [BSI_RNGs], MAC authentication mechanism based on TDES with dual key (112 Bit)
- Key hierarchy of cryptographic static secret keys: debit key, load key and issuer key are the static secret keys. The debit key's access rights are a subset of the load key's access rights. The load key's access rights are a subset of the issuer key's access rights.
- Session key derivation and monotonic transaction counter to limit the usage of the static application keys (Issuer key, Load key and Debit key)
- Session authentication mechanism: each transaction is MACed. The MAC is based on a session key derived from one of the static keys, random challenge and transaction counter
- File access control mechanism based on the static application keys: access conditions are mapped to commands and file types. Access conditions can be "always" (command can be performed outside a secure session), "never" (command cannot be performed for the mapped file type) or session (command can only be executed within a secure session) indicating the hierarchically lowest required static key for authentication.
- Session atomicity, i.e. rollback of file modifications in case of secure session failure except transaction counter
- Integrity protected ratification status in order to prevent abuse of ratification status to gain unallowed access.
- Secured management of static application keys in NVM
- Robust set of sensors and detectors for the purpose of monitoring proper chip operating conditions.
- Hardware enabled program flow integrity protection mechanism
- Test entry protection
- Peripheral locking via a Memory Management Unit to prevent unallowed peripheral accesses
- NVM integrity protection

1.4.4 Interfaces of the TOE

The TOE has an RFI interface and is compliant to the ISO/IEC 14443 standard, including parts 1, 2, 3 and 4, and to CEN/TS 16794.

1.4.5 TOE life cycle

See [PP] section 2.6 “TOE Life Cycle”

The TOE delivery point establishes the limits of the evaluation and is located after the end of Phase 2, which means phase 1 and phase 2 are covered by the TOE evaluation. The delivery follows the writing of the chip’s ID and the setting of the CSN.

2 Conformance Claims (ASE_CCL)

2.1 Conformance Claims (ASE CCL)

This ST is conformant to the Common Criteria version 3.1 Revision 5, i.e. CC Part 1 [CC1], CC Part 2 [CC2] and CC Part 3 [CC3].

It claims CC Part 2-extended conformance and CC Part 3-extended conformance.

CC Part 2 is extended with FCS_RNG “Generation of random numbers”.

CC Part 3 is extended with the security assurance component AVA_SPECIFIC.1 “Vulnerability analysis of Calypso Basic products at Enhanced Basic”attack potential.

The evaluation is driven by the following documents:

- Common Methodology for Information Technology Security Evaluation, Evaluation methodology [CEM];
- Application of Attack Potential to Smartcards [JIL-AAPS].

2.1.1 PP Claim

This ST is strictly conformant to [PP].

2.1.2 Package Claim

This ST is conformant to the EAL2+ assurance package which consists of the predefined EAL2 package augmented with ALC_DVS.1 and AVA_SPECIFIC.1, as defined in [PP] section 6.2 “AVA_SPECIFIC – Vulnerability analysis of Calypso Basic products”.

3 Security Problem Definition (ASE_SPD)

See [PP] section 4 “Security Problem Definition”.

Application Note regarding OSP.DIVERSIFICATION: The [PP] requests the keys to be diversified and written during the initialization phase. This however contrasts with the statement in [PP] from section 2.6 “TOE Life Cycle”:

“Phase 3 is performed by the card embedder and consists in writing the start-up information (including the AID and the three Application keys) using ISO/IEC 7816-4 commands and pre-defined commands.”

Phase 3 corresponds to the pre-personalization phase. Therefore the author of this ST assumes the statement from [PP] for OSP.DIVERSIFICATION to be incorrect, whereby the term initialization should be replaced by pre-personalization.

4 Security Objectives

See [PP] section 5 “Objectives”.

5 Extended Component Definition (ASE_ECD)

See [PP] section 6 “Extended Requirements”.

Statement regarding the evaluation methodology for AVA_SPECIFIC:

The evaluation methodology is as stated in [CEM] section 16.2.3 “Evaluation of sub-activity (AVA_VAN.3)”.

6 Security Requirements (ASE_REQ)

6.1 Security Functional Requirements

See [PP] section 7.1 “Security Functional Requirements”.

6.1.1 FCS_CKM.1 Cryptographic key generation

FCS_CKM.1	Cryptographic key generation
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	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [CalypsoCrypto][1] and specified cryptographic key sizes 112 Bit[2] that meet the following: [CalypsoCrypto][3]

Application Note: This SFR applies to the Calypso Basic session keys. The Calypso Basic specification [C-BASIC] ensures the uniqueness and unpredictability of the session keys by using a challenge that includes the TC and a random number.

6.1.2 FCS_CKM.4 Cryptographic key destruction

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 with random values during next Open Secure Session command[4] that meets the following: None[5]

Application Note: The static application keys cannot be destroyed or replaced. The destruction of Calypso Basic session keys does not require to comply with a specific standard.

6.1.3 FCS_COP.1 Cryptographic operation

FCS_COP.1	Cryptographic operation
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data of the Composite TOE without security attributes, or FDP_ITC.2 Import of user data of the Composite TOE with security attributes, or FCS_CKM.1 Cryptographic key management] FCS_CKM.4 Cryptographic key destruction.
FCS_COP.1.1	The TSF shall perform Session MAC authentication[1] in accordance with a

[1] [assignment: cryptographic key generation algorithm]
[2] [assignment: cryptographic key sizes]
[3] [assignment: list of standards]
[4] [assignment: cryptographic key destruction method]
[5] [assignment: list of standards]

	specified cryptographic algorithm MAC Algorithm 1 with Padding method 2[2] and cryptographic key sizes 112-bits (TDES keys option 2)[3] that meet the following: ISO/IEC 9797-1 and ISO/IEC 18033-3[4]

Application Note: MAC Algorithm 1 with Padding method 2 is covered in ISO/IEC 9797-1, and TDES keying option 2 is covered in ISO/IEC 18033-3.

Application Note: The Calypso Basic specification [C-BASIC] ensures the uniqueness and the unpredictability of the MAC by including the TC and the random number. The cryptographic keys used for the Session MAC authentication are those generated with FCS_CKM.1.

6.1.4 FCS_RNG.1 Generation of Random Numbers

FCS_RNG.1	**Random Number Generation**
Hierarchical to	No other components.
Dependencies	No dependencies
FCS_RNG.1	Random number generation **Class PTG.1** according to [BSI_RNGs]
FCS_RNG.1.1	The TSF shall provide a physical[5] random number generator that implements:
PTG.1.1	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.1.2	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.
PTG.1.3	The online test detects non-tolerable statistical defects of the internal random numbers. The online test is applied continuously. When a defect is detected, the output of further random numbers is prevented.
PTG.1.4	Within one year of typical use, the probability that an online alarm occurs is in the order of 10^{-6} or larger if the RNG works properly.
FCS_RNG.1.2	The TSF shall provide numbers in the format 8-bit[6] that meet
PTG.1.5	Test procedure A, as defined in [BSI_RNGs] does not distinguish the internal random numbers from output sequences of an ideal RNG.

Application Note: The definition of the quality metric should meet recognized standards, see for instance [SOGIS-ACM].

Refinement: Online alarm in the context of the TOE RNG is defined as the delay necessary to wait for further entropy in case at least 10 bits of raw random data occur without transition (I.e. all 0's or all 1's).

[1] [assignment: list of cryptographic operations]
[2] [assignment: cryptographic algorithm]
[3] [assignment: cryptographic key sizes]
[4] [assignment: list of standards]
[5] [selection: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic]
[6] [assignment: format of the numbers]

6.1.5 FDP_ACC.1 Subset access control

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 Calypso Basic access control SFP [1] on: • Subjects: Calypso Basic Application • Objects: Files • Operations: read(), modify(), invalidate()[2].

Application Note: The correspondence with Calypso Basic commands is the following:

- read() stands for read_record(): reads a single record from the indicated file;
- modify() stands for
    - write_record(): writes over the data of the indicated record of a file;
    - append_record(): adds a record to a Cyclic file; this record becomes the first record of the file; the last record is removed;
    - update_record(): replaces the data of the indicated record with the new data provided;
    - increment_value(): increments the value of a Counter file;
    - decrement_value(): decrements the value of a Counter file.
- invalidate() stands for the command with the same name, which invalidates the DF; when the DF is invalidated, any command modifying the file system data is rejected.

Application Note: There are other commands in the Calypso Basic product that are not concerned or affected by the access control to the file's contents:

- select_application(): selects an application in the Calypso Basic product making the DF of the application the current DF and the current file and returns information about this application; aborts any secure session currently opened; if the command fails, the current file and current DF remain unchanged;
- select_file(): gets the current DF or sets the current file pointer to a specific EF; returns the type and parameters of the file;
- open_secure_session(key.id, file_id): opens a secure session with the indicated key key.id; when a file identifier file_id is indicated, it becomes the current file; otherwise, the currently selected file is used;
- close_secure_session(): closes the currently open secure session; if the Session MAC high part is correct, the modifications are committed, otherwise, the modifications are rollbacked;
- get_data(): returns the requested data.

Application Note: The usage scenarios can be characterized by a sequence of operations denoted by ";" in the following examples:

- Ticket-loading: select_application(); open_secure_session(); if (transaction_counter > 0) then read_record() and (update_record() and/or write_record() and/or append_record() and/or increment_value()); close_secure_session()
- Ticket-control: select_application(); open_secure_session(); if (transaction_counter > 0) then read_record(); close_secure_session().

Remark from the ST author: in the Ticket-control example from the [PP] the transaction counter check is missing, which is always performed during an open_secure_session() attempt. As this is an example only and focus is actually on access control, strict conformancy is not affected.

[1] [assignment: access control SFP]

[2] [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]

- Ticket-debiting: select_application(); open_secure_session();if (transaction_counter > 0) then (read_record(); decrement_value(); append_record()); close_secure_session().

6.1.6 FDP_ACF.1 Security attribute based access control

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 initialization
FDP_ACF.1.1	The TSF shall enforce the Calypso Basic access control SFP [1] to objects based on the following: Security attributes of the Calypso Basic Application: • session_status (closed/1:Issuer/2:Load/3:Debit) • DF_status (valid/invalid) • access_rights: file x operation → {Always, Session_1, Session_2, Session_3, Never, undefined}[2] .
FDP_ACF.1.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: • DF_status = invalid and (op = invalidate or op=read), or • DF_status = valid and access_rights(f,op) = Always, or • DF_status = valid and (access_rights(f,op) = Session_i and session_status = j and j <= i) where op is the operation requested over the file f[3].
FDP_ACF.1.3	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none[4].
FDP_ACF.1.4	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: at least one of the conditions stated in FDP_ACF.1.2 is not satisfied, that is: • DF_status = invalid and op = modify, or • access_rights(f,op) = Never or undefined, or • access_rights(f,op) = Session_i and session_status = j and j > i where op is the operation requested over the file f[5].

Application Note: The value « undefined » is introduced to fully cover the domain (file_type x operation).

6.1.7 FDP_RIP.1 Subset residual information protection

FDP_RIP.1	Subset residual information protection
Hierarchical to:	No other components.
Dependencies:	No dependencies
FDP_RIP.1.1	The TSF shall ensure that any previous information content of a resource is made unavailable upon the deallocation of the resource from[1] the following objects: session keys, Session MAC, random numbers[2]

[1] [assignment: access control SFP]

[2] [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]

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

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

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

6.1.8 FDP_ROL.1 Basic rollback

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 Calypso Basic access control SFP[3] to permit the rollback of the modify() and invalidate() operations[4] on the Calypso Basic files and DF_status, respectively, without modifying the Transaction Counter[5]
FDP_ROL.1.2	The TSF shall permit operations to be rolled back within the limits of the set of operations performed in the failed or interrupted secure session[6].

Application Note: Recall that modify() stands for write_record(), append_record(), update_record(), increment_value() and decrement_value().

Application Note: When the Calypso Basic Application cannot authenticate the Terminal, i.e. when the Session MAC high is incorrect, the secure session fails. It can also be interrupted/cancelled by the Calypso Basic Application for other reasons.

6.1.9 FMT_MSA.1 Management of security attributes

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_SMF.1 Specification of management functions FMT_SMR.1 Security roles
FMT_MSA.1.1	The TSF shall enforce the Calypso Basic access control SFP [7] to restrict the ability to modify[8] the security attributes session_status, DF_status and access_rights [9] to the following roles: • Calypso Basic Application in a secure session role for session_status and DF_status • No role for access_rights[10].

6.1.10 FMT_MSA.3 Static attribute initialisation

FMT_MSA.3	Static attribute initialization
Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes

[1] [selection: allocation of the resource to, deallocation of the resource from]
[2] [assignment: list of objects]
[3] [assignment: access control SFP(s) and/or information flow control SFP(s)]
[4] [assignment: list of operations]
[5] [assignment: information and/or list of objects]
[6] [assignment: boundary limit to which rollback may be performed]
[7] [assignment: access control SFP(s), information flow control SFP(s)]
[8] [selection: change_default, query, modify, delete, [assignment: other operations]]
[9] [assignment: list of security attributes]
[10] [assignment: the authorised identified roles]

	FMT_SMR.1 security roles
FMT_MSA.3.1	The TSF shall enforce the Calypso Basic access control SFP [1] to provide restrictive[2] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow the no role[3] to specify alternative initial values to override the default values when an object or information is created.

Application Note: Default values are given upon selection of Calypso Basic application:

- access_rights = fixed in the file structure
- DF_status = last status
- session_status=closed.

6.1.11 FMT_MTD.1 Management of TSF data

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 modify, export[4] the list of the following TSF data [5] to the following authorized identified roles: • modify the CSN to no role after Phase 2 • modify the Calypso Basic file structure to no role after Phase 2 • modify the file access conditions to no role after Phase 2 • modify the static application keys or key index to no role after Phase 3 • modify the Ratification Status to "not-ratified" to the Calypso Basic Application in a secure session • modify (decrement) the Transaction Counter to the Calypso Basic Application in a secure session • modify (once) the DF status to the Calypso Basic Application in a secure session • export static or session application keys to no role[6].

Application Note: A secure session request consists in the Calypso Basic Application successfully receiving an open_secure_session() command.

Refinement: The author of this ST added "in a secure session" to the list item "modify (once) the DF status to the Calypso Basic Application to stress the fact that an authenticated secure session is necessary for this modification.

6.1.12 FMT_MTD.2 Management of limits on TSF data

FMT_MTD.2	**Management of limits on TSF data**
Hierarchical to:	No other components.
Dependencies:	FMT_MTD.1 Management of TSF data

[1] [assignment: access control SFP, information flow control SFP]
[2] [selection, choose one of: restrictive, permissive, [assignment: other property]]
[3] [assignment: the authorised identified roles]
[4] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]
[5] [assignment: list of TSF data]
[6] [assignment: the authorised identified roles]

	FMT_SMR.1 Security roles
FMT_MTD.2.1	The TSF shall restrict the specification of the limits for Transaction Counter and DF status[1] to no role[2]
FMT_MTD.2.2	The TSF shall take the following actions, if the TSF data are at, or exceed the indicated limits: • If Transaction Counter is '0' then the TSF shall not accept any request to open a secure session • If DF status is 'invalid' then the TSF shall not allow any file modification[3].

6.1.13 FMT_MTD.3 Secure TSF data

FMT_MTD.3	**Secure TSF data**
Hierarchical to:	No other components.
Dependencies:	FMT_MTD.1 Management of TSF data
FMT_MTD.3.1	The TSF shall ensure that only secure values are accepted for • Key index • Transaction Counter lower and upper bounds • Transaction Counter • DF status • Keys used for MAC calculation[4]

Application Note: In the context of the Calypso Basic Application, a secure value is a value that is managed internally and exclusively by the TSF, i.e. no imported data is accepted. The Calypso Basic specification [C-BASIC] provides the following values restrictions :

- Key Index '1', '2' or '3'
- Transaction Counter upper bound equal to '1000'
- Transaction Counter higher than '0'
- Transaction Counter strictly decreasing
- DF status 'valid' or 'invalid'
- DF status 'invalid' irreversible.
- Keys used for MAC calculation are generated as specified in FCS_CKM.1

6.1.14 FMT_SMR.1 Security roles

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: • Calypso Basic Application in a secure session • Calypso Basic Application out of a secure session. [5]

[1] [assignment: list of TSF data]
[2] [assignment: the authorised identified roles]
[3] [assignment: actions to be taken]
[4] [assignment: list of TSF data]
[5] [assignment: the authorised identified roles]

FMT_SMR.1.2	The TSF shall be able to associate users with roles.

Application Note: here "users" stands for the Terminal. The association is initiated upon reception of an open_secure_session command provided the Transaction Counter is higher than 0.

This application note, does not follow the application note from [PP]. The[PP] application note only covers the status "open_secure_session is received, while Transaction Counter is higher than 0 AND DF status is valid". However this ST also covers Security Roles in case DF status is invalid (secure sessions are still possible), thus providing a superset to the SFR claimed in [PP] and is therefore still strictly compliant.

6.1.15 FPT_FLS.1 Failure with preservation of secure state

FPT_FLS.1	Failure with preservation of secure state
Hierarchical to:	No other components.
Dependencies:	No dependencies
FPT_FLS.1.1	The TSF shall preserve a secure state when the following types of failures occur:

- Random number generation failure (FCS_RNG.1)
- Cryptographic key generation failure (FCS_CKM.1)
- Cryptographic operation failure (FCS_COP.1)
- File access control failure (FDP_ACC.1, FDP_ACF.1)
- Transaction Counter reached the lower limit (FMT_MTD.2)
- Secure session failure (FTP_ITC.1)
- Rollback failure (FDP_ROL.1)
- Deallocation failure (FDP_RIP.1)
- TSF data protection failure (FMT_MSA.1, FMT_MTD.1, FMT_MTD.3)
- Unexpected or unknown commands
- Unexpected termination[1]

Application Note: A secure state is a state in which the assets' security properties (confidentiality, integrity, etc.) are enforced.

Application Note: The following are examples of failures

- Related to file access control: Failed or aborted modification commands because:
  - DF was set to 'invalid'
  - Access rights were not correctly enforced
- Related to the secure session:
  - Incorrect Session MAC
  - Limit of the allowed number of modifications inside a secure session exceeded
  - Unsuccessful close_secure_session() command
- Related to termination:
  - Card tearing
  - Calypso Basic application selection
  - Calypso Basic application termination.

[1] [assignment: list of types of failures in the TSF]

6.1.16 FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1	Inter-TSF trusted channel
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_ITC.1.1	The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification ~~or disclosure~~[1].
FTP_ITC.1.2	The TSF shall permit another trusted IT product[2] to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for any modification, i.e. (re)loading and debit/validation transactions and application invalidation[3].

Application Note: The trusted IT product with which the TSF communicates using the provided communication channel is a Terminal. The Terminal requests the trusted channel by sending an open_secure_session() command. The trusted channel is used to communicate the commands of the secure session.

Application Note: The Calypso Basic specification [C-BASIC] enforces the protection of the integrity of the channel data but does not provide or allow any confidentiality protection. To avoid any misunderstanding, the protection from disclosure is not considered for Calypso Basic products.

6.2 TOE Security Assurance Requirements

See [PP] section 7.2 “Security Assurance Requirements”.

6.3 Security Requirements Rationale

See [PP] section 7.3 “Security Requirements Rationale”.

[1] Editorially refined

[2] [selection: the TSF, another trusted IT product]

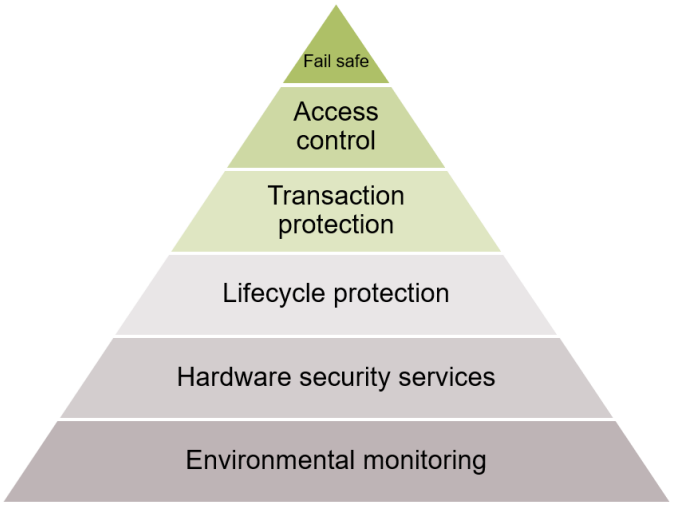
[3] [assignment: list of functions for which a trusted channel is required][assignment: rules]

7 TOE Summary Specification (ASE_TSS)

7.1 Introduction

The TSF is based on a hierarchical security functionality as shown below:

Figure 2 TOE hierarchical security functionality



Hardware security features to monitor environmental conditions are the basis upon which other TSF is built. The next level describes Hardware based security services. These services are utilized by the application software to protect security assets and also to prevent unallowed lifecycle stage transmissions. A protected Calypso Basic Transaction is the fourth layer providing logical and cryptographical measures to secure and rollback a transaction. A mutual authentication mechanism and provision of three static and hierarchical application keys are the basis for the fifth layer: role based access control. In case a security violation is detected in any layer, a secure state must be preserved. Therefore a proper failure handling is placed on top of the hierarchy.

7.2 TSF

7.2.1 SF_environment:

The environmental monitoring consists of a frequency sensor, a fuse and an active shield. The active shield provides protection against physical probing and forcing attacks. The fuse contributes to lifecycle protection and the frequency sensor prevents single stepping of instructions. These measures support the robustness of all SFRs.

7.2.2 SF_HardwareServices:

The MMU allows to lock and unlock peripherals and memory sections, which the software uses to e.g. lock CLUART while security relevant code is executed and confidential data accessed, which prevents leakage via the RFI interface.

The ISS offers a mechanism to control the program flow and to check the integrity and the confidentiality of the program flow.The code integrity is protected by hash calculation over each instruction (accumulated signature value). The ISS is used for execution protection of security relevant code of the Calypso Basic software.

The physical true RNG is used to provide a challenge for the MAC based authentication mechanism.

The µDESS is used by the Calypso Basic software for all TDES operations. It provides measures against leakage and fault attacks. The measures are based on scrambling and masking.

A CRC module is used to protect security relevant NVM data. Pages in the memory which contain the security relevant data contain several Bits of CRC data. These CRC data are calculated prior programming a page into the NVM using the CRC module.

The hardware services support the robustness of all SFRs.

7.2.3 SF_LifeCycle

During the pre-personalisation phase the instruction “put data” has to be used to write initialization values, such as the three keys (Issuer, Load and Debit). After transition to issuance stage this command is blocked. In order to personalise the card an authentication is required with one of the static keys, typically the “Issuer key”. The TOE also supports reloading. During a reloading new rights are written into Calypso Basic files. Information may also be added or updated. Personalisation and Reloading strictly follow the rules of Calypso Basic secure transactions and role based access control as described in the next chapters.

The lifecycle protection supports all SFRs.

7.2.4 SF_SecureTransaction

Any file modifications have to be performed within a secure channel. A trusted channel with a terminal is established by a session MAC authentication based on **SF_HardwareServices** from the µDESS. To derive session keys with low probability of recurrence the physical true RNG is used for the challenge data. This prevents replay attacks. As soon a transaction is finished or aborted the session keys are erased by random numbers. Session MAC and random challenges are zeroised. A transaction has to be closed with a session MAC. In case the session is not closed with a correct session MAC, all file modifications during that transaction are cancelled, which is implemented by a rollback buffer. This part of the TSF contributes to the SFRs as follows:

FCS_CKM.1, FCS_CKM.4, FCS_COP.1, FCS_RNG.1, FDP_RIP.1, FDP_ROL.1, FMT_SMR.1, FTP_ITC.1

7.2.5 SF_RoleBasedAccessControl

The TOE differentiates between the roles “in a secure session” and “out of secure session”. Before any successful authentication is established only a limited set of funtionality is available, e.g. application and file selection.

Access follows the strict definition of [C-BASIC] section 5.2 “File Structure”. For definition of “group”, “access modes” and “Key Identifiers” refer to [C-BASIC] section 6.4 “Access Conditions”.

The invalidation of a Calypso Basic application is definitive, it cannot be made valid again. If the application is invalid, any file modification is prevented by the TOE (modification commands are rejected). If the DF status is neither valid nor invalid a security reset is issued by the TOE.

CSN, Calypso Basic file structure, file access conditions and Static application keys are written during phase 3. Afterwards (starting from phase 4) modification of these TSF data is blocked by the TOE.

Ratification status is set by the TOE to “not ratified” just before the acknowledgment of the session closure is sent. Immediately it receives a next Calypso Basic command ratification status is set to “ratified”

The transaction counter is decremented after a successful open session command. If the transaction counter is 0, the TOE blocks any secure transaction attempts. Any modification of the transaction counter is guarded by flow-control measures such as ISS. The transaction counter is never rolled back.

The static application keys cannot be directly accessed (neither read nor modified). Any key index, which is not 1, 2 or 3 is rejected (TOE returns error code).

This part of the TSF contributes to the SFRs as follows:

FDP_ACC.1, FDP_ACF.1, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_MTD.2, FMT_MTD.3

7.2.6 SF_FailureHandling

There are several failure types, which require the TOE to handle appropriately to preserve a secure state. Here, thre ones are depicted from SFR FPT_FLS.1:

- Incorrect Session MAC: in case of a MAC verification failure all file modifications are rolled back.
- Limit of Session MAC exceeded: The TOE sends an error message and aborts the session, whereby the session is closed and all file modifications are cancelled.
- Unsuccessful close session command: in case the TOE can't close the session, same handling as for "Incorrect Session MAC". In case the terminal does not receive TOE response, ratification procedure is used.
- Failed or aborted commands:
  - Invalid DF Access: TOE returns error on "Select Application" command
  - Transaction counter: TOE returns error on Open Secure Session
  - Access rights violated: TOE returns error on command requesting access
- (Expected/Unexpected) Shut down: only the case of a shutdown during a transaction is relevant. In this case all file modifications are cancelled.
- Cryptographic operation failure: transaction is cancelled or not started
- Unexpected commands: TOE returns error
- Unexpected command order: TOE returns error
- Transaction replay attempt: transaction is cancelled or not started

This part of the TSF contributes to the SFRs as follows:

FPT_FLS.1

7.3 Assignment of Security Functional Requirements to TOE's Security Functionality

The justification and overview of the mapping between security functional requirements (SFR) and the TOE's security functionality (TSF) is given in the sections above. The results are shown in Table 2. The security functional requirements are addressed by at least one related security feature.

Table 2 Mapping of SFR and SF

SFR	SF_Environment	SF_HardwareServices	SF_LifeCycle	SF_SecureTransaction	SF_RoleBasedAccessControl	SF_FailureHandling
FCS_CKM.1	x	x	x	x		
FCS_CKM.4	x	x	x	x		
FCS_COP.1	x	x	x	x		
FCS_RNG.1	x	x	x	x		

FDP_ACC.1	x	x	x		x	
FDP_ACF.1	x	x	x		x	
FDP_RIP.1	x	x	x	x		
FDP_ROL.1	x	x	x	x		
FMT_MSA.1	x	x	x		x	
FMT_MSA.3	x	x	x		x	
FMT_MTD.1	x	x	x		x	
FMT_MTD.2	x	x	x		x	
FMT_MTD.3	x	x	x		x	
FMT_SMR.1	x	x	x	x		
FPT_FLS.1	x	x	x			x
FTP_ITC.1	x	x	x	x		

8 References

Reference Name	Standard Description
[BSI_RNGs]	A proposal for: Functionality classes for random number generators, Wolfgang Killmann, Werner Schindler, Version 2.0, 18 Sept 2011
[C-PRIME]	Calypso Revision 3 Specification – Portable Object Application – Version 3.3
[C-BASIC]	Calypso Specification - Calypso Basic – Version 1.1 (Ref. 191011)
[CalypsoCrypto]	Calypso Specification, Calypso Basic Cryptographic Algorithms, Version 3, 01/02/2022
[CC1]	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model; CCMB-2017-04-001, Version 3.1, Revision 5, April 2017
[CC2]	Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components; CCMB-2017-04-002, Version3.1, Revision5, April 2017
[CC3]	Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components; CCMB-2017-04-003, Version 3.1, Revision 5, April 2017
[CEM]	Common Methodology for Information Technology Security Evaluation, Evaluation methodology; CCMB- 2017-04-004, Version 3.1, Revision 5, April 2017
[ISO18033_3]	ISO/IEC 18033-3: 2005 - Information Technology - Security techniques - Encryptionalgorithms - Part 3: Block ciphers
[ISO/IEC 9797-1]	ISO/IEC 9797-1: 2011 - Information Technology - Security techniques - Message Authentication Codes - Part 1: Mechanisms using block cipher
[JIL-AAPS]	Application of Attack Potential to Smartcards, Version 3.1, June 2020, edited by JIL
[PP]	Calypso Networks Association, Calypso Specification, Calypso Basic – Protection Profile under the reference ANSSI-CC-PP-2021/01, Version 1.0, 2021-10-26

9 List of Abbreviations

C_ID	Chip type ID, Product name
CNA	Calypso Networks Association
DSI	Design step information
ISS	Instruction Stream Signature

10 Revision History

Major changes since the last revision

Version	Description of change
V0.1	Initial draft version
V3.0	Final version

Edition 2022-11-25

Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Email: erratum@infineon.com

<DOC_Number>
Document reference