GLOBALPLATFORM®
THE STANDARD FOR SECURE DIGITAL SERVICES AND DEVICES

GlobalPlatform Technology
Trusted User Interface PP-Module
Version 0.0.0.3

Public Review

September 2020

Document Reference: GPD_SPD_142

Contents

Figures

Tables

1 Introduction

Title:	Trusted User Interface PP-Module (TUI PP-Module), ref. GPD_SPE_142
Base PP:	TEE PP, ref. GPD_SPE_021
Identification:	GPD_SPE_142
Sponsor:	GlobalPlatform
Editor:	GlobalPlatform
Date:	October 2019
Version:	0.0.0.3
CC Version:	3.1 Revision 5

This document defines the TUI PP-Module, which extends the TEE Protection Profile. The scope of this PP-Module is the Trusted User Interface (TUI) on which applications rely for displaying and/or retrieving sensitive information to/from an end user. It targets a TEE running with at least one input and/or output peripheral permanently or temporarily controlled by the TEE. Examples of output peripherals are screens, audio speakers, light indicators, etc. Examples of input peripherals are keyboards, cameras, microphones, buttons, fingerprint sensors, etc. A touch screen is an example of an input-output peripheral.

The TUI PP-Module aims to be applicable to any (set of) user interface(s), independently of the form of output or input peripherals involved for presenting and/or retrieving the sensitive user information. It is therefore written in a generic way.

The peripherals needed for capturing or retrieving the sensitive information from/to the end-user shall be wired and integral to the device. Their drivers shall be among the TEE components (part of the Trusted OS). The software completing any required peripheral activity may be implemented as part of the TEE and all data will be processed and stored protected by the TEE. Otherwise, its implementation will ensure an equivalent level of data protection.

The evaluation assurance level applicable to the TUI PP-Module is EAL2+, as defined in the TEE Protection Profile.

The TUI PP-Module is aimed at being used together with the core [TEE PP]. It can be freely used as well in combination with the Debug PP-Module, the Time & Rollback PP-Module and the Biometric System PP-Module.

1.1 Audience

This document is intended to support all actors in the TEE value chain: peripheral developers, implementers of the TUI in the TEE itself, integrators (in particular handset makers), service providers implementing Trusted Applications running inside the TEE that need to display or / and to retrieve sensitive information to / from the user, as well as ITSEFs, certification bodies and certificate consumers.

1.2 IPR Disclaimer

Attention is drawn to the possibility that some of the elements of this GlobalPlatform specification or other work product may be the subject of intellectual property rights (IPR) held by GlobalPlatform members or others. For additional information regarding any such IPR that have been brought to the attention of GlobalPlatform, please visit https://www.globalplatform.org/specificationsipdisclaimers.asp. GlobalPlatform shall not be held responsible for identifying any or all such IPR, and takes no position concerning the possible existence or the evidence, validity, or scope of any such IPR.

1.3 References

Table 1-1: Normative References

Standard / Specification	Description	Ref
CC Part 1	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model. Version 3.1, revision 5, April 2017. CCMB-2017-04-001.	[CC1]
CC Part 2	Common Criteria for Information Technology Security Evaluation, Part 2: Security functional requirements. Version 3.1, revision 5, April 2017. CCMB-2017-04-002.	[CC2]
CC Part 3	Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements. Version 3.1, revision 5, April 2017. CCMB-2017-04-003.	[CC3]
CEM	Common Methodology for Information Technology Security Evaluation, Evaluation Methodology. Version 3.1, revision 5, April 2017. CCMB-2017-04-004.	[CEM]
GPD_SPE_009	TEE System Architecture, GlobalPlatform (Latest applicable version)	[TEE Arch]
GPD_SPE_010	TEE Internal Core API Specification, GlobalPlatform (Latest applicable version)	[TEE Core API]
GPD_SPE_007	TEE Client API Specification, GlobalPlatform (Latest applicable version)	[TEE Client API]
GPD_SPE_021	TEE Protection Profile v1.2.1 (or Latest applicable version)	[TEE PP]
GPD_SPE_055	TEE Trusted User Interface Low-level API (Latest applicable version)	[TEE TUI LL API]
RFC 2119	Key words for use in RFCs to Indicate Requirement Levels	[RFC 2119]

1.4 Terminology and Definitions

The following meanings apply to SHALL, SHALL NOT, MUST, MUST NOT, SHOULD, SHOULD NOT, and MAY in this document (refer to [RFC 2119]):

- **SHALL** indicates an absolute requirement, as does **MUST**.
- **SHALL NOT** indicates an absolute prohibition, as does **MUST NOT**.
- **SHOULD** and **SHOULD NOT** indicate recommendations.
- **MAY** indicates an option.

Selected terms used in this document are included in Table 1-2. Additional terms are defined in [TEE PP] and TEE functional specifications.

Table 1-2: Terminology and Definitions

Term	Definition
Application Programming Interface (API)	A set of rules that software programs can follow to communicate with each other.
Client Application (CA)	An application running outside of the Trusted Execution Environment (TEE) making use of the TEE Client API that accesses facilities provided by the Trusted Applications inside the TEE. Contrast *Trusted Application*.
Consistency	A property of the TEE persistent storage that stands at the same time for runtime and startup consistency. Runtime consistency stands for the guarantee that the following clauses hold: • Read/Read: Two successful readings from the same storage location give the same value if the TEE did not write to this location and the TEE was not reset in between • Write/Read: A successful reading from a given storage location gives the value that the TEE last wrote to this location if the TEE was not reset in between. Startup consistency stands for the guarantee that the following clause holds: • During a given power cycle, the stored data used at startup is the data for which runtime consistency was enforced on the same TEE on a previous power cycle. Consistency implies runtime integrity of what is successfully written and read back – values or code. However, the stored data used at startup may be restored from an old power cycle, not the latest one. It is still consistent at start-up because it corresponds to a memory snapshot at a given time, but it represents an integrity loss compared with the latest power cycle. This notion is weaker than integrity that must be preserved between power cycles.
Device binding	Device binding is the property of data being only usable on a unique given system instance, here a TEE.

Term	Definition
Execution Environment (EE)	A set of hardware and software components that provide facilities (computing, memory management, input/output, etc.) necessary to support applications.
Monotonicity	Monotonicity is the property of a variable whose value is either always increasing or always decreasing over time.
Power cycle	A power cycle is the lapse between the moment a device is turned on and the moment the device is turned off afterwards.
Production TEE	A TEE residing in a device that is in the end user phase of its life cycle
Regular Execution Environment (REE)	An Execution Environment comprising at least one Regular OS and all other components of the device (SoCs, other discrete components, firmware, and software) which execute, host, and support the Regular OS (excluding any Secure Components included in the device). From the viewpoint of a Secure Component, everything in the REE is considered untrusted, though from the Regular OS point of view there may be internal trust structures. (Formerly referred to as a *Rich Execution Environment (REE)*.) Contrast *Trusted Execution Environment*.
Regular OS	An OS executing in a Regular Execution Environment. May be anything from a large OS such as Linux down to a minimal set of statically linked libraries providing services such as a TCP/IP stack. (Formerly referred to as a *Rich OS* or *Device OS*.) Contrast *Trusted OS*.
Root of Trust (RoT)	Generally, the smallest distinguishable set of hardware, firmware, and/or software that must be inherently trusted and which is closely tied to the logic and environment on which it performs its trusted actions.
System-on-Chip (SoC)	An electronic system all of whose components are included in a single integrated circuit.
TA instance time / TA persistent time	Time value available to a Trusted Application through the TEE Internal Core API. The API offers two types of time values: System Time, which exists only during runtime, and Persistent time, which persists over resets. System Time must be monotonic for a given TA instance, and the returned value is called “TA instance time”. Persistent time depends only on the TA but not on a particular instance, it must be monotonic even across power cycles. Its monotonicity across power cycles is related to the Time and Rollback optional PP-Module.
TEE Client API	The software interface used by clients running in the REE to communicate with the TEE and with the Trusted Applications executed by the TEE.
TEE Internal Core API	The software interface exposing TEE functionality to Trusted Applications.
TEE Service Library	A software library that includes all security related drivers.
Trusted Application (TA)	An application running inside the Trusted Execution Environment that provides security related functionality to Client Applications outside of the TEE or to other Trusted Applications inside the TEE. Contrast *Client Application*.

Term	Definition
Trusted Execution Environment (TEE)	An Execution Environment that runs alongside but isolated from an REE. A TEE has security capabilities and meets certain security-related requirements: It protects TEE assets against a set of defined threats which include general software attacks as well as some hardware attacks, and defines rigid safeguards as to data and functions that a program can access. There are multiple technologies that can be used to implement a TEE, and the level of security achieved varies accordingly. Contrast *Regular Execution Environment.*
Trusted OS	An OS executing in a Secure Component. Contrast *Regular OS*.
Trusted Storage	In GlobalPlatform TEE documents, *trusted storage* indicates storage that is protected to at least the robustness level defined for OMTP Secure Storage (in section 5 of [OMTP-TR1]). It is protected either by the hardware of the TEE, or cryptographically by keys held in the TEE. If keys are used they are at least of the strength used to instantiate the TEE. A GlobalPlatform TEE Trusted Storage is not considered hardware tamper resistant to the levels achieved by Secure Elements.
Trusted User Interface	A user interface that ensures that the displays and input components are controlled by the TEE and isolated from the REE and even the TAs.

1.5 Abbreviations and Notations

Table 1-3: Abbreviations and Notations

Abbreviation / Notation	Meaning
API	Application Programming Interface
CA	Client Application
CC	Common Criteria (defined in [CC1], [CC2], [CC3])
CEM	Common Evaluation Methodology (defined in [CEM])
CM	Configuration Management (defined in [CC1])
EAL	Evaluation Assurance Level (defined in [CC1])
EE	Execution Environment
ID	Identifier
NA	Not Applicable
OS	Operating System
OSP	Organisational Security Policy (defined in [CC1])
PCB	Printed Circuit Board
PP	Protection Profile (defined in [CC1])
RAM	Random Access Memory

Abbreviation / Notation	Meaning
REE	Regular Execution Environment
RFC	Request For Comments; may denote a memorandum published by the IETF
ROM	Read Only Memory
SAR	Security Assurance Requirement (defined in [CC1])
SFP	Security Function Policy (defined in [CC1])
SFR	Security Functional Requirement (defined in [CC1])
SoC	System-on-Chip
SPD	Security Problem Definition (defined in [CC1])
ST	Security Target (defined in [CC1])
TA	Trusted Application
TEE	Trusted Execution Environment
TOE	Target of Evaluation (defined in [CC1])
TSF	TOE Security Functionality (defined in [CC1])
TUI	Trusted User Interface
TSFI	TSF Interface (defined in [CC1])

1.6 Revision History

GlobalPlatform technical documents numbered *n*.0 are major releases. Those numbered *n*.1, *n*.2, etc., are minor releases where changes typically introduce supplementary items that do not impact backward compatibility or interoperability of the specifications. Those numbered *n.n*.1, *n.n*.2, etc., are maintenance releases that incorporate errata and precisions; all non-trivial revisions are indicated, often with revision marks.

Table 1-4: Revision History

Date	Version	Description
May 2019	0.0.0.*1*	Initial document with first draft of the TOE description, assets, threats and objectives
June 2019	0.0.0.*2*	Update following TEE Spec and TEE Security joint meeting (May 20th)
October 2019	0.0.0.3	Approved terminology, alignment with TEE PP and BIO PP-Module Complete document for TEE Security WG review

2 TOE Overview

This chapter defines the type of the Target of Evaluation (TOE), describes the functional behaviour of the TOE, presents typical TOE architectures, and describes the TOE's main security features and intended usages as well as the TOE's life cycle.

2.1 TOE Type

The TOE type is the GlobalPlatform TEE extended with TUI capabilities.

Applications rely on the TUI for interacting with an end user, i.e. for capturing and/or retrieving sensitive information from/to the user. The TOE includes some form of ancillary input (e.g. a keyboard, touch screen, camera, microphone, buttons, fingerprint sensor, eye scanner, etc.) and/or output (e.g. a screen, a vibrator, an audio speaker, etc.) peripherals.

A typical device may provide many peripherals but the TEE is not expected to have software drivers for interacting with every peripheral connected to the device.

Different classes of peripherals are considered:

- TEE-exclusive peripherals: peripherals that are permanently isolated from non-TEE entities, managed by the TEE and fully usable by a calling TA;
- REE-exclusive peripherals: peripherals that are outside the control of the TEE and that are not usable by the TEE or its TAs;
- Shared peripherals: peripherals which can be managed by either the TEE or the REE, and require a transfer from the REE to be used by the TEE or its TAs.

A calling TA invoking the system's peripheral discovery functionality receives a list of all peripherals that are visible to it. The list may include peripherals that are under the control of the TEE as well as peripherals that are under the control of the REE.

Peripherals which are under the full or partial control of the TEE may support exclusive access by no more than one TA at a time. The TUI is a user interface that ensures that the displays and input components are controlled by the TEE and are isolated from the REE and other TAs.

The exact list of input and output peripherals must be specified in the ST along with their class and their capabilities. For each peripheral it must be specified whether exclusive access is supported or not.

The TUI also contains software components, i.e. the peripherals' drivers, that run inside the TEE.

Optionally, a TUI includes one or more Security Indicators for providing a specific indication that the input or output peripheral can be considered trusted and secured by the user (the TAs).

The Security Indicator may be hardware or software-based, e.g. an output peripheral such as a LED, or a designated display area. In both cases, the Security Indicator is under permanent TEE control. It is managed directly and exclusively by the TEE itself and it is never accessible by the REE or a TA

The TOE comprises:

- All TEE-controlled hardware, firmware and software used to provide the TUI functionality ;
- Input and output peripherals including firmware and hardware. The exact list of these peripherals must be specified in the ST along with their class and their capabilities.

- All the required device drivers that may be used by the TUI functionality to access the input and output peripherals;
- (optional) Security Indicator(s);
- The guidance for the secure usage of the TUI functionality after delivery.

The TOE does not comprise:

- The capture functionality of the input peripherals;
- The presentation or display functionality of the output peripherals;
- The TAs running on top of the TEE that do not implement TUI functionality;
- The Regular Execution Environment (REE);
- The Client Applications running on top of the REE.

Application Note:

The TUI PP-Module does not mandate functional compliance with GlobalPlatform APIs specifications [TEE TUI LL API]. Note that GlobalPlatform specification classifies the peripherals in the following four classes:

- TEE ownable: peripherals that are temporarily or permanently isolated from non-TEE entities, managed by the TEE and fully usable by a calling TA;
- TEE controllable: peripherals for which the TEE cannot interpret events (because it does not have the required driver) but for which it can control the flow of events;
- TEE parseable: REE-controlled peripherals for which the TEE can parse and forward events;
- Not usable by the TEE: peripherals that are outside the control of the TEE and that are not usable by the TEE or its TAs.

TEE ownable, TEE parseable, and TEE controllable peripherals are either under the exclusive control of the TEE or shared between the REE and the TEE.

2.2 TOE Description

2.2.1 Functional Description

The TOE provides the core TUI functionality and may support different types of protection for each peripheral: data protection in confidentiality, data protection in integrity, or data protection in both integrity and confidentiality. The ST author needs to specify the exact protection that can be provided on each peripheral. By initialization, the TUI always ensures the authenticity of the peripherals it controls. The TUI protects the information that is transmitted through a *locked* peripheral to the application for which it was locked.

When a peripheral is locked for an application, no other application can gain access to it or to its resources. The peripheral must be released or unlocked by the application that locked it. It may also be released by the end user or under other specific circumstances (e.g. after a threshold of inactivity has been reached).

For an input peripheral protected in integrity, the data captured by the TUI is protected by the TOE in integrity until it has been transmitted to the application that locked the input peripheral.

For an output peripheral protected in integrity, the data transmitted to the TUI by the application that locked the peripheral is protected in integrity until it has been displayed by the peripheral.

All data captured by a peripheral secured in confidentiality and all data transmitted by an application to the TUI to display on a peripheral secured in confidentiality is protected in confidentiality.

Sensitive data exchanged between a peripheral and the TUI, i.e. raw data captured from the user by an input peripheral or sensitive data meant to be presented to the user via an output peripheral, should never be exported.

The capture function itself lies outside the TOE boundary. However, the TA that required it has exclusive access to the input peripheral, and therefore to the captured input data. The exclusive access applies only to the time of usage and only to captured input data.

The display function itself lies outside the TOE boundary. However, the TA that required it has exclusive access to the output peripheral, and therefore to the displayed data. The exclusive access applies only to the time of usage and only to displayed data.

A calling TA can concurrently open a TUI session on multiple peripherals (e.g. keyboard/display opening). When a TUI session is terminated, any state specific to the calling TA's activity or to the TA/user interaction is removed and the control of the peripheral is given back to the peripheral's owner (the TEE or the REE).

In Figures 2-1 and 2-2 the case of an open TUI session on a single peripheral supporting exclusive access is considered. The diagrams depict overviews of the data and control flow between the involved components during a TA's interaction with an input and output peripheral, respectively.

The arrows shown in orange depict the optional interaction and data flow involving the Security Indicator.

The communication channel between the TOE and the peripherals shown in green is protected in integrity and/or confidentiality.

The area shown in light gray lies inside the TOE.

Figure 2-1 TUI Functionality – Input Peripherals - Overview

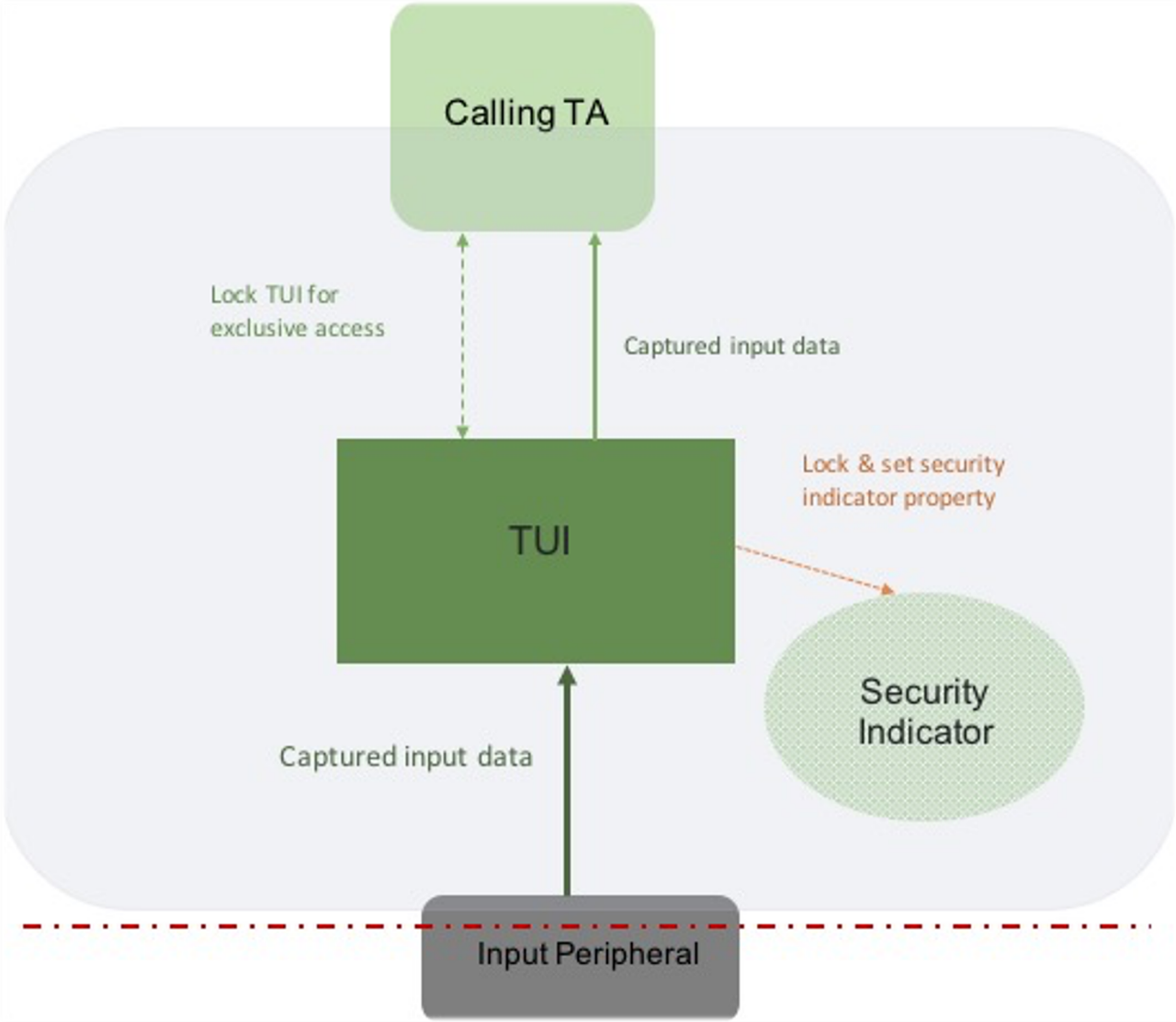
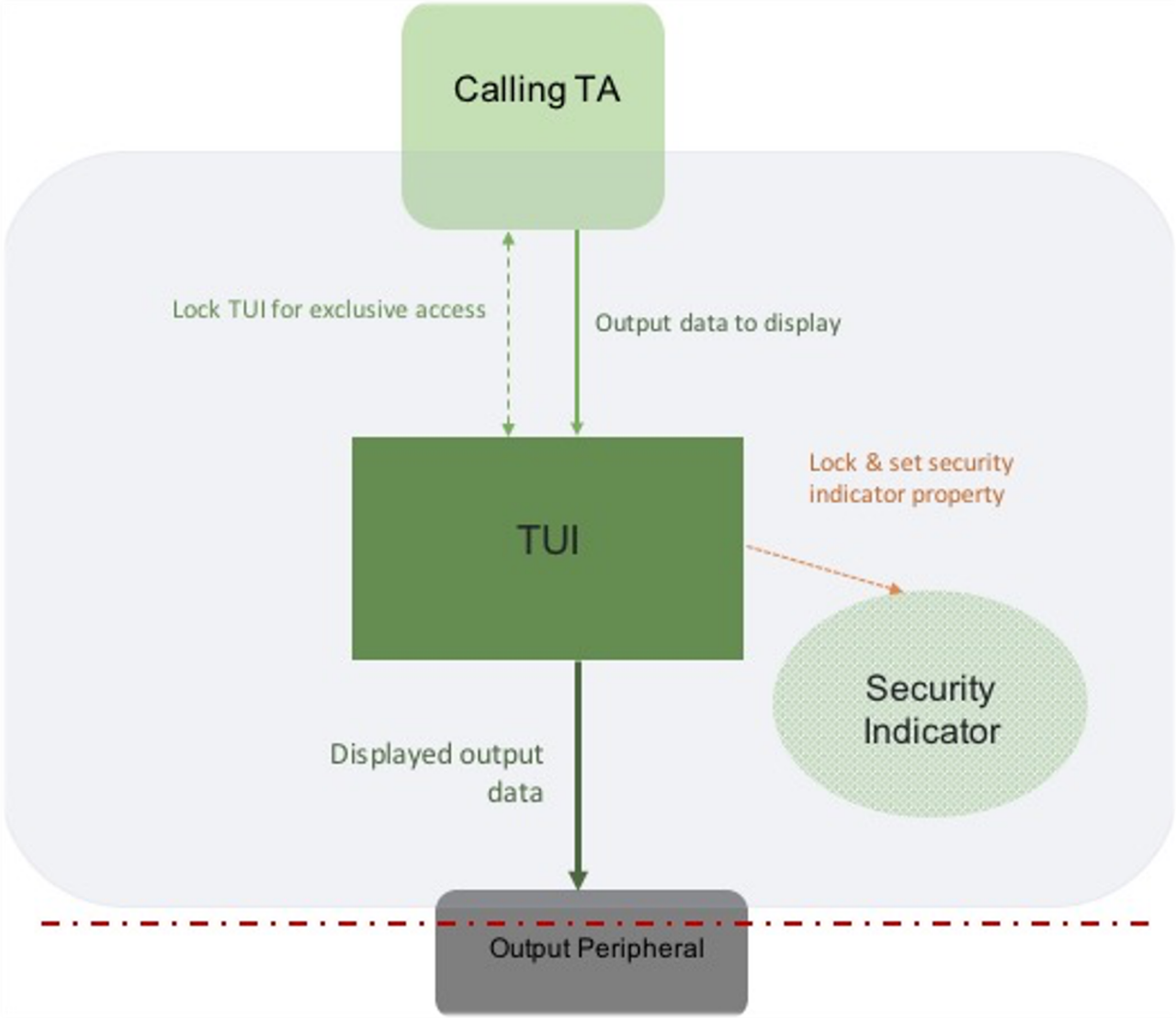
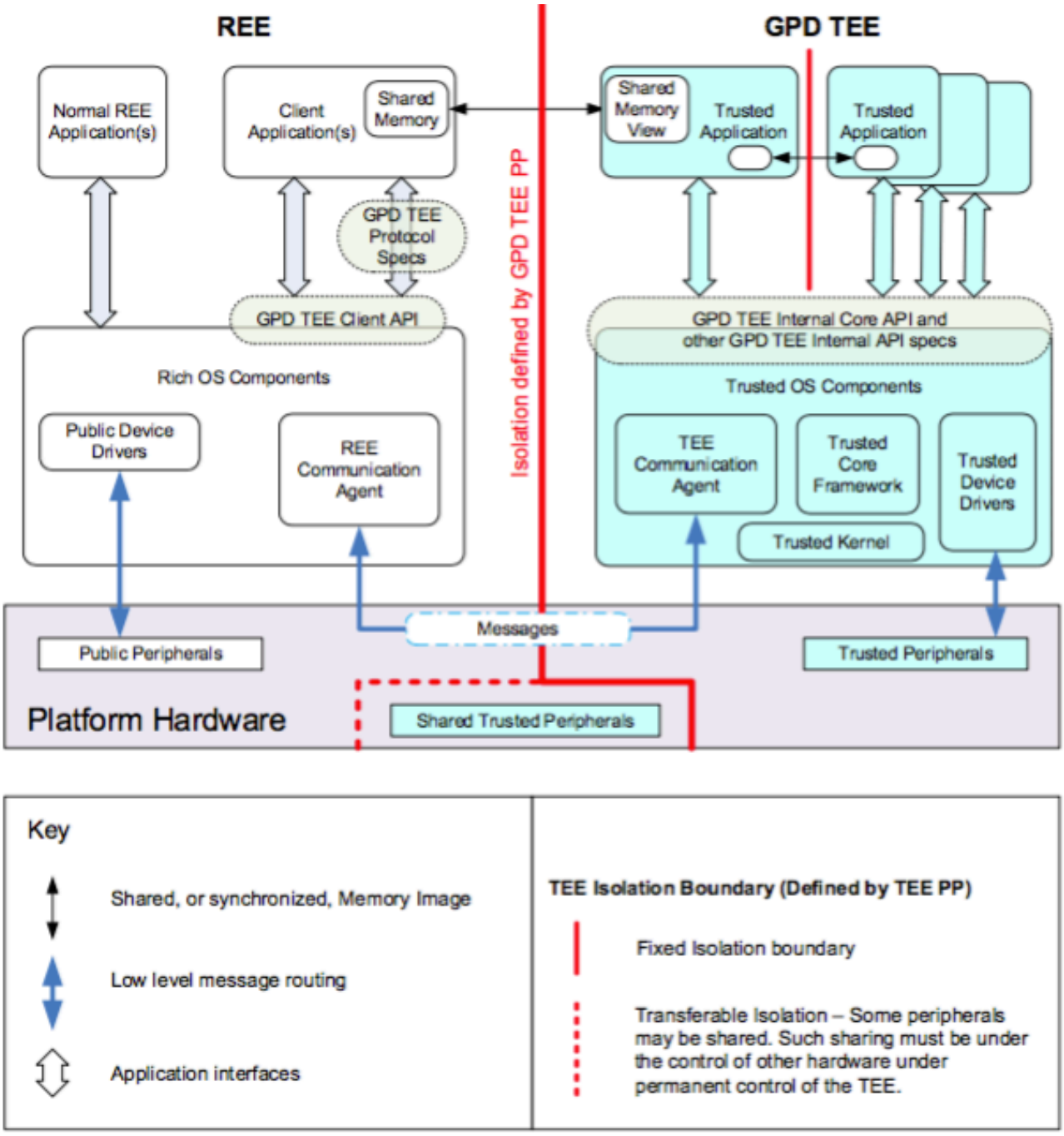


Figure 2-2 TUI Functionality - Output Peripherals - Overview



2.2.2 Architecture

The architecture of the TOE is depicted below.



The hardware components required for providing TUI functionality on a TEE-enabled device are:

- Input peripherals: the devices that are in charge of capturing the input information from the users. Any input peripheral shall be wired and integral to the TEE-enabled device. The capture function itself lies outside the TOE boundary, but at the time of usage, the system has exclusive access to the captured data. The input peripheral(s) controlled by the TEE shall be started by the TEE. In all cases, the authenticity and integrity of the input peripheral code shall be ensured. The input peripheral code may be initialized by the TEE or stored in ROM memory (not loaded by the TEE). The drivers for such input peripherals shall be among the Trusted OS Components. No access to the captured data is given and any usage of the captured data is made through the API within the TOE.

- Output peripherals: the devices that are in charge of displaying the output information to the users. Any output peripheral shall be wired and integral to the TEE-enabled device. The output data creation function used to generate the output data lies outside the TOE boundary, but at the time of usage, the system has exclusive access to the output data to be displayed. The output peripheral controlled by the TEE shall be started by the TEE. In all cases, the authenticity and integrity of the output peripheral code shall be ensured. The output peripheral code may be initialized by the TEE or stored in ROM memory (not loaded by the TEE). The drivers for such output peripherals shall be among the Trusted OS Components.
- (optional) Security Indicator(s): dedicated output peripheral(s) such as a LED controlled exclusively by the TEE and used to inform the user that a locked peripheral is secured.

The software components required for providing TUI functionality on a TEE-enabled device are:
- the input and output peripheral drivers;
- the API for using the input and output data;
- (optional) software-based Security Indicator(s).

2.3 Usage and Major Security Features of the TOE

2.3.1 TOE Security Functionality

The major security features offered by the TOE are the following:

For all available peripherals:

- Peripheral discovery;
- Peripheral lock (acquisition) and unlock (release);
- Peripheral ownership transfer, i.e. the ownership for some peripherals may be transferred during their lifetime (e.g. access to the TS controller may be transferred to the TEE by the REE during a TUI session, then transferred back to the REE when the TUI session closes).

For peripherals under the total or partial control of the TEE:

- Peripheral isolation;
- Peripheral authenticity;
- Integrity protection;
- Confidentiality protection;
- (optional) Security Indicator(s).

Application Note:

The ST author shall complete the descriptions of the security functionality with the characteristics of the actual TOE and shall provide the complete set of input/output peripherals considered within the evaluation.

2.3.2 TOE Usage

The device will be equipped with at least one input and/or output peripheral, and optionally with a Security Indicator. This enables a wide range of services, for instance, device unblock, mobile financial services and authentication services.

2.4 Available Non-TOE Hardware/Software/Firmware

The TOE may require some non-TOE hardware, software or firmware. However, the TOE must be realized in such a way that TOE security functionalities do not rely on proper behavior of non-TOE hardware, software or firmware.

The capture function of input peripherals is a necessary non-TOE component.

The presentation function of output peripherals is a necessary non-TOE component.

Application Note:

The ST author shall complete the descriptions of the available non-TOE hardware/software/firmware with the list of non-TOE resources used by the TOE.

2.5 Reference Device Life Cycle

The generic life cycle defined in the [TEE PP] applies to the TOE as defined in this PP-Module.

Application Note: The ST author shall describe the actual TOE life cycle, in particular with regard to the development and integration of input and output peripherals hardware and software components.

Conformance Claims

2.6 Conformance Claim to CC

This PP-Module is CC Part 2 [CC2] conformant.

2.7 Conformance Claim to a Package

This PP-Module inherits the assurance level EAL2+ from [TEE PP], which consists of predefined EAL 2 augmented with AVA_TEE.2.

2.8 Conformance Claim to the PP-Module

This PP-Module inherits from [TEE PP] the strict conformance as defined in [CC1] for all Security Targets claiming conformance to a PP-Configuration that includes this PP-Module.

2.9 Consistency Rationale wrt TEE PP

The consistency rationale against the base PP is given in the following chapters and sections.

3 Security Problem Definition

This chapter introduces the security problem addressed by the TUI and its operational environment. It applies to a generic input and/or output peripheral.

3.1 Assets

Depending on the implementation, the assets can be either TEE or TA assets (data or code). Independently of the implementation, the TEE/TA assets are in the scope of the evaluation and their security properties should be ensured.

As a general remark, we would like to remind that for **runtime** data, the integrity and consistency properties are equivalent. For **TEE runtime data**, these imply: consistency and confidentiality. For **TEE persistent data**, these imply: authenticity, consistency, confidentiality, and device-binding. For **TA code**, these imply authenticity and consistency. For **TA data**, these imply authenticity, consistency, confidentiality, device-binding, and atomicity.

However, some of the runtime data must additionally be *replay-protected*. Integrity excludes the possibility of data injection at the level of the TEE memory, which is what replay protection is understood to be.

The assets are the following:

EXCHANGED_DATA

This covers data transmitted between a locked peripheral and the calling TA. It represents runtime data.

For an input peripheral, the EXCHANGED_DATA represents INPUT_DATA captured by the input peripheral through its capture functionality on behalf of the calling TA.

For an output peripheral, the EXCHANGED_DATA represents OUTPUT_DATA displayed or presented via the output peripheral on behalf of the calling TA.

Properties (in the TOE): consistency (integrity), confidentiality (e.g. for privacy).

Application Note:

The communication from the peripheral to the TEE may be in PCB but not physically protected. The TEE will protect the captured/displayed against SW attacks.

The EXCHANGED_DATA asset is TA data.

PERIPHERAL_FIRMWARE

The peripheral's firmware. To a minimum, this includes the peripheral's driver. Such code is persistent and lies inside the TOE boundary.

Properties: integrity/consistency, authenticity, rollback protection.

Application Note:

This asset applies to peripherals that are under TEE control.

Depending on the implementation, this asset extends **TEE firmware** and / or **TA code** assets as defined in [TEE PP]. For a TEE level implementation, the rollback protection is ensured. An implementation at the TA level however would require the use of Time and Rollback PP-Module to ensure rollback protection.

TUI_RUNTIME_DATA

TUI associated runtime data. Such data may include handles to events or to input runtime data.

Properties: consistency, confidentiality.

PERIPHERAL_SETUP

The peripheral's configuration data and settings. Such data is persistent.

Properties: integrity, authenticity.

Application Note:

This asset applies to all peripherals.

If the system includes one or more Security Indicator(s), the following asset must be additionally considered:

(optional) SECURITY_INDICATOR

The code and data of security indicator(s), which is a hardware output peripheral or a software mechanism (e.g. a dedicated display area) that is always under the control of the TEE and is not accessible to the REE or the TAs. The security indicator's state shows whether or not the associated peripheral is secured.

Security properties: Integrity.

Application Note:

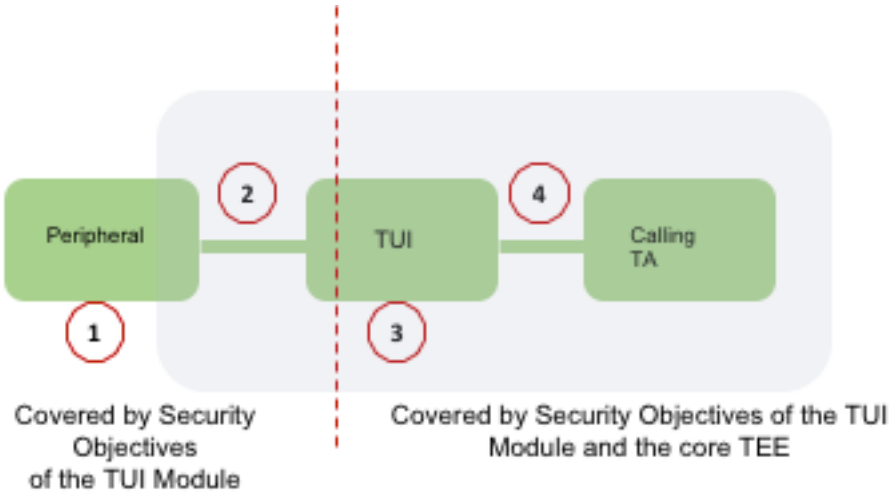
- A simple 2 state LED could be used even in a scenario when peripherals can be made secured independently. For example, if the light indicator indicates that the screen is secured, the secured screen can then indicate that the audio speaker and microphone are also secured.
- The exact meaning/interpretation of the information indicated by the security indicator has to be specified in the ST.

3.2 Threats

The same threat model and attackers' profiles as included in the [TEE PP] apply.

In Figure 3-1 below, the different main components, as well as the different main points at which attacks can be conducted are depicted.

Figure 3-1 Attack Points - Overview



Attacks can be grouped into direct and indirect attacks. Attacks on the peripheral and its functions, i.e. attacks at point 1 in Figure 3-1, are the only direct attacks. They require subverting/replacing the peripheral hardware or functions, but otherwise they require no specific knowledge about the peripheral or its inner working. These types of attacks are external to the TOE.

The only attack at the level of the peripheral, i.e. at point 1 in Figure 3-1, that is internal to the TOE is an attack on the access to the peripheral's function handling input/output data. For input peripherals this is the capture function. For output peripherals this is the display/presentation function.

T.SHARED_FUNCTION_ACCESS

An attacker intercepts and/or modifies exchanged data, taking advantage of a shared access to the peripheral's data-handling functionality and thus to the runtime data exchanged between the peripheral and the TOE.

Assets threatened indirectly: EXCHANGED_DATA (sent through the communication channel between the peripheral and the TOE)

This attack will be covered by the following objectives:

- O.FUNCTION_ACCESS,
- O.PREVENT_RESIDUAL_DATA,
- O.PROTECTED_ COMMUNICATION_CHANNEL.

Application Note:

For input peripherals, the goal of such an attack can be manifold, for instance:

- To extract genuine captured input data for subsequent replay;
- To alter genuine captured data before they are transmitted to the TOE.

Attacks at Point 2

Attacks at this point target the communication channel between the peripheral and the TUI. At this level, an attacker can potentially intercept data sent by/to the peripheral to/by the TUI.

T.EXTRACT_EXCHANGED_DATA

An attacker intercepts and extracts the data sent by/to the peripheral to/by the TUI through a communication channel.

Assets threatened indirectly: EXCHANGED_DATA (sent through the communication channel between the peripheral and the TUI)

This attack will be covered by the following objectives:

- O.PROTECTED_ COMMUNICATION_CHANNEL.

T.MODIFY_EXCHANGED_DATA

An attacker intercepts and modifies the data sent by/to the peripheral to/by the TUI through a communication channel.

Assets threatened indirectly: EXCHANGED_DATA (sent through the communication channel between the peripheral and the TUI)

This attack will be covered by the following objectives:

- O.PROTECTED_ COMMUNICATION_CHANNEL.

T.INJECT_EXCHANGED_DATA

An attacker injects data through the communication channel between the peripheral and the TUI.

Assets threatened indirectly: EXCHANGED_DATA (sent through the communication channel between the peripheral and the TUI)

This attack will be covered by the following objectives:

O.PROTECTED_ COMMUNICATION_CHANNEL.

Application Note:

For input peripherals, extraction and injection of the input data may be the two steps of a replay attack. First, an attacker intercepts and steals the captured input data. The interception is done when the input peripheral acquires input data from a genuine user and then sends it to the TOE through a communication channel. As a second step, the attacker injects or replays the stolen input data. The goal of this attack is to bypass the input peripheral.

Attacks at Point 3

T.MODIFY_FIRMWARE

An attacker modifies the firmware (drivers) pertaining to the locked peripheral to alter the behavior of the TUI.

Assets threatened directly: PERIPHERAL_FIRMWARE.

This attack will be covered by the following objectives:

O. PERIPHERAL_INITIALIZATION.

Attacks at Point 4

Attacks at this point are similar to those at point 2 but they target the data transmitted between the TUI and the TA that locked the peripheral.

T.INJECT_DATA

An attacker bypasses the TUI and the peripheral and injects (malicious or corrupt) runtime data.

Assets threatened directly: EXCHANGED_DATA

This attack will be covered by the following objectives:

- O.RUNTIME_INTEGRITY from [TEE PP].

T.EXTRACT_DATA

An attacker extracts data outside the TOE by intercepting the data transmitted between the TUI and the TA that locked the peripheral.

Assets threatened directly: EXCHANGED_DATA

This attack will be covered by the following objectives:

- O.RUNTIME_CONFIDENTIALITY from [TEE PP].

T.MODIFY_DATA

An attacker modifies the data transmitted between the TUI and the TA that locked the peripheral.

Assets threatened directly: EXCHANGED_DATA.

This attack will be covered by the following objectives:

- O.RUNTIME_INTEGRITY from [TEE PP].

Other Attacks

T.RESIDUAL

An attacker extracts unprotected residual security-relevant data during a TUI's session or from the cache.

This attack covers multiple scenarios:

- The attacker takes advantage of a flaw in the user interface of the TOE and gets access to the memory content, the cache or relevant temporary data;
- The attacker takes advantage of residual information such as residual runtime data at the level of the peripheral.

Assets threatened directly: EXCHANGED_DATA, TUI_RUNTIME_DATA, PERIPHERAL_SETUP.

This attack will be covered by the following objectives:

- O.RUNTIME_CONFIDENTIALITY from [TEE PP].

T.CORRUPT_RUNTIME_DATA

An attacker corrupts runtime data, such as a handle to runtime data, a handle to the locked peripheral or a reference provided to the TA. Such data can be manipulated to alter the system's expected behavior.

Assets threatened directly: TUI_RUNTIME_DATA.

Application Note: In a GlobalPlatform compliant implementation, this threat stands, for instance, for overwriting the *EventSourceHandle*.

This attack will be covered by the following objectives:

O.RUNTIME_INTEGRITY from [TEE PP].

T.CORRUPT_SETUP

An attacker modifies the setup of the locked peripheral.

Assets threatened directly: PERIPHERAL_SETUP.

T.PERIPHERAL_ACCESS

An attacker accesses (reads or writes) the peripheral while another TA has locked it.

Assets threatened directly: EXCHANGED_DATA.

Application Note: This may be, for example, by using an undocumented feature of the peripheral that allows dumping its memory to some location.

T.MODIFY_SEC_IND

An attacker directly or indirectly modifies the behaviour or the data of the security indicator either such that it may not indicate as secured a peripheral that is secured, or such that it may indicate as secured a peripheral that is not secured. This is a software attack.

Assets threatened directly: SECURITY_INDICATOR.

3.3 Organizational Security Policies

This module does not define any organizational security policy for implementation by the TOE and/or its operational environment.

3.4 Assumptions

This section defines the assumptions holding on the non-TOE components and the TOE's operational environment.

A.TA_DEVELOPMENT_TUI

TA developers shall follow the guidance documentation provided with the TOE and set by the TEE provider. This documentation shall include the constraints that need to be respected by the applications for each peripheral to ensure that data can be interpreted.

Application Note: This assumption is meant to ensure that data for which integrity shall be ensured by the TUI matches the range of the peripheral used by the application. For example, the microphone may not be capable of capturing all audible sounds. The application developer shall ensure that the application fits within these limits.

This assumption complements A.TA_DEVELOPMENT defined in the [TEE PP].

A.NO_RESIDUAL_DATA

The function of the peripheral that handles data exchanged between the peripheral and the TOE does not store residual exchanged data. The exchanged data is deleted between any two consecutive operations.

3.5 Correspondence to TEE PP Assets and SPD

3.5.1 Assets

For a full TEE implementation, the EXCHANGED_DATA is TA data. All other assets are TEE assets (data or code).

For a TEE/TA implementation, the persistent assets may be either TEE or TA code and data.

The table below shows the relationship between TEE assets as defined in the [TEE PP] and assets defined in this module. It is applicable to a full TEE implementation.

Peripheral Assets \ TEE Assets	TEE identification	TEE initialisation code & data	TEE storage root of trust	RNG	TA code	TA data and keys	TA instance time	TEE runtime data	TEE persistent data	TEE firmware
EXCHANGED_DATA						X				
PERIPHERAL_SETUP									X	
PERIPHERAL_FIRMWARE										X
TUI_RUNTIME_DATA								X		
SECURITY_INDICATOR										X[1]

3.5.2 SPD

The following table summarizes the correspondence between the threats, assumptions and OSPs identified for the TEE and those identified in this module. A full TEE implementation of the TOE is considered.

[1] TEE Firmware or hardware

Peripheral SPD / TEE SPD	T.ABUSE_FUNCT	T.CLONE	T.FLASH_DUMP	T.IMPERSONATION	T.ROGUE_CODE_EXECUTION	T.PERTURBATION	T.RAM	T.RNG	T.SPY	T.TEE_FIRMWARE_DOWNGRADE	T.STORAGE_CORRUPTION	OSP_INTEGRATION_CONFIGURATION	OSP.SECRETS	A.PROTECTION_AFTER_DELIVERY	A.TA_MANAGEMENT	A.TA_DEVELOPMENT
T.SHARED_FUNCTION_ACCESS																
T.MODIFY_EXCHANGED_DATA																
T.INJECT_EXCHANGED_DATA																
T.EXTRACT_EXCHANGED_DATA																
T.MODIFY_FIRMWARE	X			X	X	X				X						
T.MODIFY_DATA					X		X		X							
T.INJECT_DATA					X		X		X							
T.EXTRACT_DATA							X		X							
T.CORRUPT_SETUP					X	X					X					
T.CORRUPT_RUNTIME_DATA					X	X	X									
T.RESIDUAL							X		X							
T.PERIPHERAL_ACCESS				X												
T.MODIFY_SEC_IND																
A.TA_DEVELOPMENT_TUI																X
A.NO_RESIDUAL_DATA																

Attacks at point 1, i.e. T.SHARED_FUNCTION_ACCESS and at point 2, i.e. T.MODIFY_EXCHANGED_DATA, T.INJECT_EXCHANGED_DATA, and T.EXTRACT_EXCHANGED_DATA are specific to this module and disjoint from threats defined in the [TEE PP].

The attack T.MODIFY_SEC_IND on the Security Indicator(s) is specific to this module.

The other attacks are specific instances of modification, disclosure and perturbation attacks considered in [TEE PP]. The mapping can be organized in the following categories:

- Modify firmware: linked to T.ABUSE_FUNCT, T.IMPERSONATION, T.ROGUE_CODE_EXECUTION T.PERTURBATION, and T.TEE_FIRMWARE_DOWNGRADE;
- Modify / tamper / Override / overwrite / Corrupt: linked to T.ROGUE_CODE_EXECUTION, T.SPY and T.RAM;
- Inject: linked to T.ROGUE_CODE_EXECUTION, T.RAM and T.SPY;
- Extract and residual data: liked to T.RAM and T.SPY;
- Corrupt runtime data: linked to T.ROGUE_CODE_EXECUTION, T.RAM and T.PERTURBATION;
- Corrupt setup: linked to T.ROGUE_CODE_EXECUTION, T.PERTURBATION, and T.STORAGE_CORRUPTION;
- Unauthorized peripheral access: linked to T.IMPERSONATION;
- Force unsafe state: linked to T.ABUSE_FUNCT and T.PERTURBATION.

4 Objectives

4.1 Security Objectives for the TOE

O.PERIPHERAL_INITIALIZATION

The TOE shall ensure that the peripheral is started through a secure initialization process that ensures the integrity of the peripheral's initialization code and data, and the authenticity of the peripheral firmware.

The TOE shall ensure that all peripheral code and data are bound to the SoC of the device.

Application Note:

This objective is the extension of the objective O.INITIALIZATION defined in the [TEE PP]. It is included here to highlight the fact that the peripheral is indeed integral to the TEE.

Application Note:

The author of a compliant ST shall describe the initialisation process of the peripheral and the mechanisms used for enforcing the authenticity of the code. For input peripherals, this excludes the code of the capture function, which is out of the scope of the TOE. For output peripherals, this excludes the output presentation function which is out of the scope of the TOE.

O.PROTECTED_ COMMUNICATION_CHANNEL

The TOE shall provide the necessary means for protecting the communication channel between the peripheral and the TUI, i.e. it will isolate and protect it from unauthorized access by the REE or other TAs, which could lead to modification, injection or disclosure of the exchanged data.

Application Note:

This means that the TOE provides appropriate access control to the communication channel that carries the exchanged data.

O.PREVENT_RESIDUAL_DATA

The TOE shall ensure that when a locked peripheral is released, all residual data under TOE control captured or presented through this peripheral are erased at the level of this peripheral. Data must be deleted or invalidated between any two consecutive operations of the peripheral.

O.DATA_ACCESS

The TOE shall ensure that, when a peripheral is locked by a TA, only that TA can access (read or write) the data exchanged with the peripheral.

O. FUNCTION ACCESS

The TOE shall ensure that:

- the data exchanged between the TOE and the peripheral is handled by the expected wired peripheral and that
- at the time of the usage, the calling TA has exclusive access to the peripheral's function handling the exchanged (input/output) data, and therefore to the exchanged data.

O.SAFE_RELEASE

The TOE shall ensure that only the TA that locked a peripheral or the TOE itself or an external event (e.g. power event) can initiate a release of that peripheral.

O.INDICATOR_ACCESS

The TOE shall ensure that the REE and no TA can modify the state of the Security Indicator: it shall only be accessible (write) by itself.

4.2 Security Objectives for the Operational Environment

This section states the security objectives for the TOE's operational environment covering all the assumptions and the organizational security policies that apply to the environment.

OE.NO_RESIDUAL_DATA

The operational environment ensures that no residual exchanged (input/output) data is stored at the level of the peripheral. The exchanged data is deleted between any two consecutive capture or presentation operations.

OE.TA_DEVELOPMENT_TUI

TA developers shall comply with the guidance documentation provided with the TOE. In particular, for each peripheral, TA developers shall respect the constraints necessary for ensuring that data can be interpreted. For each peripheral the TA developers shall consider the characteristics specified in the guidance documentation. In particular, they consider:

- the peripheral's type**,** i.e. input, output, or I/O peripheral;
- whether the peripheral supports exclusive access;
- whether it relies on a Security Indicator and how the information indicated by it is to be interpreted.

4.3 Security Objectives Rationale

The following table indicates an overview of how the threats are addressed by the security objectives of the TOE. The last column of the table indicates whether a threat is covered (partly or completely) by security objectives defined in the [TEE PP]. An orange * indicates that a threat is covered by a conjunction of security objectives specific to this TOE and security objectives defined in the [TEE PP]. A blue * indicates that a threat is completely covered by security objectives defined in the [TEE PP].

Objectives / Threats	O.PERIPHERAL_INITIALIZATION	O.PROTECTED _COMMUNICATION_CHANNEL	O.PREVENT_RESIDUAL_DATA	O.FUNCTION_ACCESS	O.DATA_ACCESS	O.SAFE_RELEASE	O.INDICATOR_ACCESS	OE.TA_DEVELOPMENT_TUI	OE.NO_RESIDUAL_DATA	TEE Objectives
T.SHARED_FUNCTION_ACCESS		X	X	X						
T.MODIFY_EXCHANGED_DATA		X								
T.INJECT_EXCHANGED_DATA		X								
T.EXTRACT_EXCHANGED_DATA		X								
T.MODIFY_FIRMWARE	X									*
T.MODIFY_DATA										*
T.INJECT_DATA										*
T.EXTRACT_DATA										*
T.CORRUPT_RUNTIME_DATA										*
T.RESIDUAL										*
T.PERIPHERAL_ACCESS					X					
T.CORRUPT_SETUP										*
T.MODIFY_SEC_IND							X			
A.TA_DEVELOPMENT_TUI								X		
A.NO_RESIDUAL_DATA									X	

4.4 Correspondence to TEE PP Objectives

The objectives presented in Section 4.1.3 are specific to the TUI extension and they are disjoint from the TEE security objectives defined in the [TEE PP].

5 Security Requirements

This chapter provides the set of Security Functional Requirements (SFRs) the TOE has to enforce in order to fulfil the security objectives.

The following security functional components defined in CC Part 2 [CC2] **Error! Reference source not found.** are used:

- FDP_ACC.1 Subset access control
- FDP_ACF.1 Security attribute based access control
- FDP_RIP.1 Subset residual information protection
- FMT_MSA.1 Management of security attributes
- FMT_MSA.2 Secure security attributes
- FMT_MSA.3 Static attribute initialisationFPT_FLS.1 Failure with preservation of secure state
- FTP_TRP.1 Trusted path

5.1 Security Policy

This PP-Module requires a security access control policy to peripherals and TUI data, called **TEE Trusted User Interface Access Control SFP**, to enforce the correct behavior of the TUI functionality and the interactions between the peripherals and the TAs.

The subjects, objects, security attributes and operations of this policy are the following:

Subjects: The active entities are the TUI system itself, the TAs and the peripherals.

Objects: persistent state and a transient state of the system.

The minimum persistent state of the system consists of:

- peripheral_list: a list of unique identifiers of authorized input/output peripherals
- (optional) si_map: a map of unique identifiers of authorized input/output peripherals and their associated Security Indicator

Application Note:

- The Security Indicator is optional in the sense that it is not necessary for a TOE to include one. TOEs may include none, one or multiple Security Indicators, i.e. one per peripheral. However, when at least one Security Indicator is available, its usage is mandatory.
- If only one hardware-based Security Indicator (e.g. LED) is available, then all input/output peripherals will have the same associated Security Indicator in the si_map.

The minimum transient state of the system consists of:

- calling_TA: the current instance of TA that requests an interaction with peripherals
- out_data: data sent from the calling TA to a locked peripheral
- in_data: data received by the calling TA from a locked peripheral

- (optional) tui_data: data displayed during the interaction between the calling TA and a locked peripheral, i.e. notifications, updates, user messages

Security attributes:

- peripheral.status: the status “locked”, “unlocked” or “exclusively-locked” of the peripheral PR
- peripheral.type: the type “IN”, “OUT” or “I/O” of the peripheral PR
- peripheral.ownership: the owner “TEE” or “REE” of the peripheral PR
- peripheral.class : the class of the peripheral: TEE-only, Shareable, REE-only
- peripheral.exclusive_access: the flag “Yes” or “No” of the peripheral PR identifying whether PR supports exclusive access
- peripheral.ownership_changed: a flag “Yes” or “No” showing if the peripheral’s ownership has been temporarily transferred (from the REE to the TEE)
- (optional) security_indicator.state: the Security Indicator’s “ON” or “OFF” state

Operations:

- peripheral_discovery: return the list p_list of peripherals available for the calling TA; p_list should be a subset of peripheral_list
- lock_peripherals(p_list): lock all peripherals in p_list for exclusive access; it ensures that for all peripherals p in p_list, p.status == exclusively-locked and p.ownership == TEE
- unlock_peripherals(p_list): unlock or release the peripherals in p_list; it ensures that for all peripherals p in p_list, p.status == unlocked and p.ownership is set to the peripheral’s initial owner
- send_data(o, p): send output data o from the calling TA to the locked peripheral p
- receive_data(i, p): send input data i from the locked peripheral p to the calling TA

Internal operations:

- capture_input(p, d): capture input data d using the peripheral p
- present_output(p, d): present/display output data d using the peripheral p
- (optional) transfer_ownership(p): (temporarily) transfer the ownership of peripheral p, i.e. from “REE” to “TEE”;
- If p.ownership == REE before running transfer_ownership(p), then p.ownership == TEE and p.ownership_changed == Yes after running it;
- (optional) switch_SI_state(si): switch the security indicator si ON or OFF;
- If si.state == OFF before running switch_SI_state(si), then si.state == ON after running it;
- If si.state == ON before running switch_SI_state(si), then si.state == OFF after running it;
- (optional) notify_tui(info): notify the TUI and display the information info

Application Note:

- The operation transfer_ownership() is optional because not all implementations may support exclusive access and claiming ownership over peripherals.
- These operations must be executed in a specific order to provide the expected service. The symbol “;” is used to indicate a sequence of operations. The symbol “+” attached to an operation is used to indicate that the operation can be run one or more times. The symbol “*” attached to an operation is used to indicate that the operation is optional, i.e. it can be run zero or more times. E.g. “transfer peripheral ownership; switch SI state; lock peripherals; send/receive data; unlock peripherals.”
- Optionally, any operation may implicitly give rise to a notification, i.e. a return of an (intermediate) result to the calling TA.

5.1.1 FDP_ACC.1/TUI Subset access control

This SFR contributes to the following security objectives:

- O.PROTECTED _COMMUNICATION_CHANNEL
- O.DATA_ACCESS
- O.INDICATOR_ACCESS
- O.FUNCTION_ACCESS

FDP_ACC.1/TUI Subset access control[2]

Dependencies:

FDP_ACF.1 Security attribute based access control: **FDP_ACF.1/TUI**

FDP_ACC.1.1/TUI The TSF shall enforce the **TEE Trusted User Interface Access Control SFP** on

- **Subjects: TUI system, TAs, peripherals**
- **Objects:**
  - **Persistent objects: peripheral_list, (optional) si_map;**
  - **Transient objects: calling_TA, in_data, out_data, (optional) tui_data;**
  - ***[assignment: other persistent or transient objects of the system]***
- **Operations:**
  - **lock_peripherals, unlock_peripherals, peripheral_discovery, send_data, receive_data;**
  - **internal operations: (optional) transfer_ownership, capture_input, present_output, (optional) switch_SI_state;**
  - **(optional) notify_tui**
  - ***[assignment: other operations of the system]***

5.1.2 FDP_ACF.1/TUI Security attribute based access control

This SFR contributes to the following security objectives:

- O.PROTECTED _COMMUNICATION_CHANNEL
- O.DATA_ACCESS
- O.INDICATOR_ACCESS
- O.FUNCTION_ACCESS

[2] FDP_ACC.1.1 The TSF shall enforce the *[assignment: access control SFP]* on *[assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP].*

FDP_ACF.1/TUI Subset access control[3]

Dependencies:

FDP_ACC.1 Subset access control: **FDP_ACC.1/TUI**

FMT_MSA.3 Static attribute initialisation: **FMT_MSA.3/TUI**

FDP_ACF.1.1/TUI The TSF shall enforce the **TEE Trusted User Interface Access Control SFP** to objects based on the following:

- **peripheral.status ("locked", "unlocked" or "exclusively-locked")**
- **peripheral.ownership ("TEE" or "REE")**
- **peripheral.type ("IN", "OUT" or "I/O")**
- **peripheral.class ("TEE-only", "Shareable", "REE-only")**
- **peripheral.exclusive_access ("Yes" or "No")**
- **(optional) security_indicator.state ("ON" or "OFF**")
- ***[assignment: list of additional security attributes]***

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

- **rule-lock-peripherals:**
  lock_peripherals (pls) is allowed
  if *pls* is a subset of peripheral_list and for all peripheral in *pls* peripheral.status == unlocked and peripheral.ownership == TEE and peripheral.exclusive_access == YES
- **(optional) rule-transfer-owner-and-lock-peripherals**:
  transfer_ownership()[+]; lock_peripherals (pls) is allowed
  if *pls* is a subset of peripheral_list and for all peripheral in *pls* peripheral.status == unlocked and peripheral.exclusive_access == YES and either peripheral.ownership == TEE or (peripheral.ownership == REE and peripheral.class == Shareable)

[3] FDP_ACF.1.1 The TSF shall enforce the [assignment: access control SFP] to objects based on the following: *[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]*.

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *[assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]*.

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *[assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]*.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects].*

- **rule-unlock-peripherals:** unlock_peripherals(pls) is allowed
  if pls is a subset of peripheral_list and for all peripherals in pls either peripheral.status == locked or exclusively-locked, and peripheral.ownership_changed == No
- **(optional) rule-transfer-owner-and-unlock-peripherals:**
  transfer_ownership(peripheral)$^{+}$; unlock_peripherals (pls) is allowed
  if *pls* is a subset of peripheral_list and for all peripheral in *pls* either peripheral.status == locked or exclusively-locked, and peripheral.ownership_changed == Yes
- **(optional) rule-lock-peripherals-with-SI:**
  switch_SI_state(si)$^{+}$; lock_peripherals (pls) is allowed
  if *pls* is a subset of peripheral_list and for all peripheral in *pls* peripheral.status == unlocked and peripheral.ownership == TEE and peripheral.exclusive_access == YES and si_map(peripheral).state == OFF
- **(optional) rule-transfer-owner-and-lock-peripherals-with-SI:**
  transfer_ownership(p)$^{+}$; switch_SI_state(si)$^{+}$; lock_peripherals (pls) is allowed
  if *pls* is a subset of peripheral_list and for all peripheral in *pls* peripheral.status == unlocked and peripheral.exclusive_access == YES and (peripheral.ownership == TEE or (peripheral.ownership == REE and peripheral.class == Shareable) and si_map(peripheral).state == OFF
- **(optional) rule-unlock-peripherals-with-SI:**
  switch_SI_state(si)$^{+}$; unlock_peripherals(pls) is allowed
  if pls is a subset of peripheral_list and for all peripherals in pls either peripheral.status == locked or exclusively-locked, and peripheral.ownership_changed == No and si_map(peripheral).state == ON
- **(optional) rule-transfer-owner-and-unlock-peripherals:**
  transfer_ownership(peripheral)$^{+}$; switch_SI_state(si)$^{+}$; unlock_peripherals (pls) is allowed
  if *pls* is a subset of peripheral_list and for all peripheral in *pls* either peripheral.status == locked or exclusively-locked, and peripheral.ownership_changed == Yes and si_map(peripheral).state == ON
  rule-send-data:
  send_data(out_data, peripheral); present_data(od, peripheral); notify_tui(tui_info)* is allowed
  if peripheral.status == locked and peripheral.type == OUT or I/O
  Application Note:
  The output data out_data sent by a TA to be presented to users on an output peripheral may be transformed by the peripheral before presenting/displaying it.
- **rule-receive-data:**
  capture_data(peripheral, in_data); receive_data (in_data, peripheral); notify_tui(tui_info)* is allowed
  if peripheral.status == locked and peripheral.type == In or I/O
- ***[assignment: list of additional rules]***

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

- ***[assignment: list of additional rules]***

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

- **rule-deny-lock-peripherals:** if any of the conditions stipulated for lock_peripherals does not hold
- **rule-deny-unlock-peripherals:** if any of the conditions stipulated for unlock_peripherals does not hold
- **rule-deny-send-data:** if any of the conditions stipulated for send_data does not hold
- **rule-deny-receive-data:** if any of the conditions stipulated for receive_data does not hold
- ***[assignment: list of additional rules]***

Application Note:

The Security Target shall complete the above rules to address all the operations supported by the system and all their usage condition.

5.1.3 FDP_RIP.1/TUI Residual information protection

This SFR contributes to the following security objectives:

- O.PREVENT_RESIDUAL_DATA.

FDP_RIP.1/TUI Subset residual information protection[4]

Dependencies: No dependencies.

FDP_RIP.1.1/TUI The TSF shall ensure that any previous information content of a resource is made unavailable upon the **deallocation of the resource from** the following objects:

- **out_data, in_data, tui_data as per FDP_ACC.1/TUI upon *[selection: lock_peripherals, unlock_peripherals, send_data, receive_data, notify_tui, capture_input, present_output]***
- **trusted path as per FDP_TRP.1/TUI upon *[selection:* unlock_peripherals*]***

5.1.4 FMT_MSA

These SFRs contribute to the following security objectives:

- All the objectives linked to FDP_ACF.1/TUI.

[4] FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: *allocation of the resource to, deallocation of the resource from*] the following objects: [assignment: *list of objects*].

5.1.4.1 FMT_MSA.1/TUI Management of security attributes

FMT_MSA.1/TUI Management of security attributes[5]

Dependencies:

[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]: **FDP_ACC.1/TUI**

Discarded dependencies:

FMT_SMR.1 Security roles: **since the operations can only be performed by the TUI system itself**.

FMT_SMF.1 Specification of Management Functions: **since only standard operations are necessary**.

FMT_MSA.1.1/TUI The TSF shall enforce the **TEE Trusted User Interface Access Control SFP** to restrict the ability to **query and modify** the security attributes:

- **peripheral.status**
- **peripheral.ownership**
- **peripheral.type**
- **peripheral.class**
- **peripheral.exclusive_access**
- **(optional) peripheral.ownership_changed**
- **(optional) security_indicator.state**
- ***[assignment: list of security attributes]***

to **the TUI system**.

5.1.4.2 FMT_MSA.2/TUI Secure security attributes

FMT_MSA.2/TUI Secure security attributes[6]

Dependencies:

[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]: **FDP_ACC.1/TUI**

FMT_MSA.1 Management of security attributes: **FMT_MSA.1/TUI**

Discarded dependencies:

FMT_SMR.1 Security roles: **since the operations can only be performed by the TUI system itself.**

[5] FMT_MSA.1.1 The TSF shall enforce the *[assignment: access control SFP(s), information flow control SFP(s)]* to restrict the ability to [*selection: change_default, query, modify, delete, [assignment: other operations]]* the security attributes *[assignment: list of security attributes]* to *[assignment: the authorized identified roles].*

[6] FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for *[assignment: list of security attributes].*

FMT_MSA.2.1/TUI The TSF shall ensure that only secure values are accepted for **the following security attributes:**

- **peripheral.status == locked or unlocked or exclusively-locked**
- **peripheral.ownership == TEE or REE**
- **peripheral.exclusive_access == Yes or No**
- **peripheral.type == IN or OUT or I/O**
- **peripheral.class == TEE-only or Shareable or REE-only**
- **peripheral.ownership_changed == Yes or No**
- **security_indicator.state: == "ON" or "OFF"**
- ***[assignment: list of security attributes]***

5.1.4.3 FMT_MSA.3/TUI Static attribute initialisation

FMT_MSA.3/TUI Static attribute initialisation[7]

Dependencies:

FMT_MSA.1 Management of security attributes: **FMT_MSA.1/TUI**

Discarded dependencies:

FMT_SMR.1 Security roles: **since the operations can only be performed by the TUI system itself.**

FMT_MSA.3.1/TUI The TSF shall enforce the **TEE Trusted User Interface Access Control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/TUI The TSF shall allow no role to specify alternative initial values to override the default values when an object or information is created.

5.1.5 FPT_FLS.1/TUI Failure with preservation of secure state

This SFR contributes to the following security objectives:

- O.DATA_ACCESS
- O.SAFE_RELEASE

FPT_FLS.1/TUI Failure with preservation of secure state[8]

---

[7] FMT_MSA.3.1 The TSF shall enforce the [assignment: access control SFP, information flow control SFP] to provide [selection, choose one of: restrictive, permissive, [assignment: other property]] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [assignment: the authorised identified roles] to specify alternative initial values to override the default values when an object or information is created.

[8] FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [assignment: *list of types of failures in the TSF*].

Dependencies: No dependencies.

FPT_FLS.1.1/TUI The TSF shall preserve a secure state when the following types of failures occur:

- **a TUI operation is halted**
- **panic occurs**
- **an operation is denied as per FDP_ACF.1/TUI**
- **a trusted path tampering has been detected as per FTP_TPR.1/TUI**
- ***[assignment: list of types of failures in the TSF].***

Application Note:

As defined in [CC1], secure state stands for "*state in which the TSF data are consistent and the TSF continues correct enforcement of the SFRs*". The ST author shall describe the secure state(s) that can be reached upon failure.

5.1.6 FTP_TRP.1/TUI

This SFR contributes to the following security objectives:

- O.PROTECTED _COMMUNICATION_CHANNEL,
- O.DATA_ACCESS
- O.FUNCTION_ACCESS

FTP_TRP.1/ Trusted path[9]

Dependencies: No dependencies.

FTP_TRP.1.1/TUI The TSF shall provide a communication path between itself and **local** 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 and disclosure by the REE and/or TAs**.

Application Note:

In practice, the trusted path provides protected communication from and to the peripherals.

"Local users" stands for a user interface.

FTP_TRP.1.2/TUI The TSF shall permit **the TSF** to initiate communication via the trusted path.

[9] FTP_TRP.1.1 The TSF shall provide a communication path between itself and [selection: *remote, local*] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [selection: *modification, disclosure, [assignment: other types of integrity or confidentiality violation]*].

FTP_TRP.1.2 The TSF shall permit [selection: *the TSF, local users, remote users*] to initiate communication via the trusted path.

FTP_ITC.1.3 The TSF shall require the use of the trusted path for [selection: initial user authentication, *[assignment: other services for which trusted path is required]*].

FTP_TRP.1.3/TUI The TSF shall require the use of the trusted path for **exchanging, i.e. sending and receiving, data to and from controlled peripherals**.

5.2 Security Objectives Rationale

The following table shows an overview of how the security objectives for the systems are covered by the SFRs.

Table 5-1: Coverage of security objectives

Security Objectives	**SFRs**
O.PERIPHERAL_INITIALIZATION	This objective is fulfilled by the TEE through FPT_INI.1
O.PROTECTED_COMMUNICATION_CHANNEL	FDP_ACC.1/TUI FDP_ACF.1/TUI FMT_MSA.1/TUI FMT_MSA.2/TUI FMT_MSA.3/TUI FTP_TRP.1/TUI
O.PREVENT_RESIDUAL_DATA	FDP_RIP.1/TUI
O.DATA_ACCESS	FDP_ACC.1/TUI FDP_ACF.1/TUI FPT_FLS.1/TUI FTP_TRP.1/TUI
O.FUNCTION_ACCESS	FDP_ACC.1/TUI FDP_ACF.1/TUI FMT_MSA.1/TUI FMT_MSA.2/TUI FMT_MSA.3/TUI FTP_TRP.1/TUI
O.SAFE_RELEASE	FDP_ACF.1/TUI FPT_FLS.1/TUI
O.INDICATOR_ACCESS	FDP_ACC.1/TUI FDP_ACF.1/TUI FMT_MSA.1/TUI FMT_MSA.2/TUI FMT_MSA.3/TUI