Car Connectivity Consortium

CCC Digital Key®

Protection Profile of the Digital Key Applet

Version 2.0
(CCC-CP-023)

CARCONNECTIVITY consortium®

VERSION HISTORY

Version	Date	Comments
1.0	2023-10-16	Approved by CCC Board.
1.0	2024-08-02	Updated Legal Notice for Certification Documents
1.1.3	2025-03-07	Approved by CCC Board - Transposition to CC:2022 - Updated PP introduction - Updated conformance claims, including new optional functional package for UWB-based secure ranging - Removed extended components (FCS_RNG) - Updated SFRs (FCS components) - Added ALC_FLR.2 (mandatory requirement) - Added Functional Package for UWB Secure Ranging in Section 6. - Modifications to FCS_CK.5 and FCS_RNG.1. - Correction of typos and other editorial changes
2.0	2025-03-24	Version for BSI Certification

LEGAL NOTICE

The copyright in this certification document is owned by the Car Connectivity Consortium LLC ("CCC LLC"). Use of this certification document and any related intellectual property contained in this certification document (collectively, the "Certification Document"), is governed by these license terms and the CCC Intellectual Property Rights Policy (the "IPR Policy").

Use of the Certification Document by any party who is not a member of CCC LLC (each such party, a "Member") is prohibited unless such party has obtained the express written consent of CCC LLC or has duly executed a license agreement with CCC LLC. The IPR Policy governs the legal rights applicable to the creation and licensing of the Certification Document, as documentation created by CCC or one of its Committees, as such term is defined in the IPR Policy. This Certification Document, regardless of its title or content, is not a Final Specification as defined in the IPR Policy.

CCC LLC hereby grants each Member a right to use and to make verbatim copies of the Certification Document for the purposes of developing, performing, and administering interoperability testing (the "Purpose"). Members are not permitted to make available or distribute this Certification Document or any copies thereof to non-Members other than to their Affiliates (as defined in the IPR Policy) and subcontractors but only to the extent that such Affiliates and subcontractors have a need to know for carrying out the Purpose and provided that such Affiliates and subcontractors accept confidentiality obligations similar to those contained in the Agreement. Each Member shall be responsible for the observance and proper performance by such of its Affiliates and subcontractors of the terms and conditions of this Legal Notice and the IPR Policy. No other license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Any use of the Certification Document not in compliance with the terms of this Legal Notice, the IPR Policy and CCC Membership Agreement (the "Membership Agreement") is prohibited, and any such prohibited use may result in termination of the applicable Membership Agreement and other liability permitted by the applicable Agreement or by applicable law to CCC LLC or any of its members for patent, copyright and/or trademark infringement.

THE CERTIFICATION DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT OF ANY THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS, AND COMPLIANCE WITH APPLICABLE LAWS.

Each Member is solely responsible for the compliance by their products and services with any applicable laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their products and services related to such regulations within the applicable jurisdictions. Each Member acknowledges that nothing in the Certification Document provides any information or assistance in connection with securing such compliance, authorizations or licenses. Each Member is responsible for the correct use of the Certification Document.

NOTHING IN THE CERTIFICATION DOCUMENT CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS. ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE CERTIFICATION DOCUMENT IS EXPRESSLY DISCLAIMED. BY USE OF THE CERTIFICATION DOCUMENT, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST CCC LLC AND ITS MEMBERS RELATED TO USE OF THE CERTIFICATION DOCUMENT.

CCC LLC reserves the right to adopt any changes or alterations to the Certification Document as it deems necessary or appropriate.

EXECUTIVE SUMMARY

The Car Connectivity Consortium (CCC) represents a large portion of the global automotive and smartphone industries, with more than one hundred member companies.

The CCC is a cross-industry standards organization with a mission to create sustainable and flexible ecosystems that standardize interface technologies to provide consistently great user experiences across all vehicles and mobile devices.

The Car Connectivity Consortium Digital Key® is a standardized ecosystem that enables mobile devices to store, authenticate, and share Digital Keys for vehicles in a secure, privacy-preserving way that works everywhere, even when the smartphone's battery is low.

Digital Key allows consumers to easily and confidently use their mobile devices to access vehicles. Along with robust capability and convenience, it offers enhanced security and privacy protections. Digital Key aims to complement traditional methods, while being robust enough to fully replace them.

The CCC Digital Key Release 3 defines the standardized interface between the vehicle and the mobile device as an NFC-based wireless interface designed for direct communication between the vehicle and mobile device. This communication may also be performed through a BLE-based wireless interface.

The Digital Key architecture uses standards-based public key infrastructure to establish end-to-end trust. Mobile devices create and store Digital Keys in Secure Elements – embedded technology that provides a tamper-resistant secure implementation – to provide the highest-level of protection from the plethora of known hardware- and software-based attacks, including tampering, storage intrusion, cloning, and unauthorized access as well as side channel, fault injection, and many other forms of attack.

To ensure this is the case, proper standards are in place and implementations will be tested and provide assurance on their security level. Common Criteria is chosen to express the security requirements for security evaluations of these implementations. To enable certification of such implementations CCC requires a Protection Profile or similar document to be created. This is what is presented in this document, a Protection Profile (PP) that may be brought through Common Criteria certification body approval at later stage. This PP has been prepared following the rules and formats of CC:2022 Revision 1.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TERMS AND ABBREVIATIONS

AID	Application Identifier
API	Application Programming Interface
BLE	Bluetooth Low Energy
CA	Certificate Authority
CC	Common Criteria
DK	Digital Key
ECU	Electronic Control Unit
EE	Execution Environment
FP	Functional Package
GP	GlobalPlatform
JC	Java Card
KTS	Key Tracking Server
MITM	Man-in-the-middle attack
NFC	Near Field Communication
NVM	Non-Volatile Memory
OEM	Original Equipment Manufacturer
PP	Protection Profile
RE	Runtime Environment
RKE	Remote Keyless Entry
SCP	Secure Channel Protocol
SE	Secure Element
SR	Secure Ranging
ST	Security Target
TOE	Target of Evaluation
TSFI	TOE Security Functionality Interface
UI	User Interface
URSK	UWB Ranging Secret Key
UWB	Ultra Wide Band
VM	Virtual Machine

(CC terminology except PP, ST, TOE and TSFI, defined in [CC1] is not listed here.)

TERMINOLOGY AND DEFINITIONS

In this document keywords are capitalized when used to unambiguously specify an interpretation. When these words are not capitalized, they are meant in their natural-language sense.

The key words “SHALL”, “SHALL NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

- SHALL - This word, or the terms “REQUIRED” or “SHALL”, mean that the definition is an absolute requirement of the specification
- SHALL NOT - This phrase, or the phrase “SHALL NOT”, mean that the definition is an absolute prohibition of the specification
- SHOULD - This word, or the adjective “RECOMMENDED”, mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications SHALL be understood and carefully weighed before choosing a different course
- SHOULD NOT - This phrase, or the phrase “NOT RECOMMENDED” mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- MAY - This word, or the adjective “OPTIONAL”, mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option SHALL be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option SHALL be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

1 INTRODUCTION

1.1 PP Identification

Name:	**Protection Profile of the Digital Key Applet**
Editor:	Internet of Trust
Sponsor:	Car Connectivity Consortium (CCC), LLC
Supported/Certified by:	Federal Office for Information Security (BSI) Germany
CC Version:	CC:2022 Revision 1
Assurance Level:	EAL 4 augmented with ALC_DVS.2, ALC_FLR.2 and AVA_VAN.5
Version:	2.0
Date:	March 24, 2025
Registration:	BSI-CC-PP-0119-V2
Keywords:	Digital Key, Secure Element

1.2 PP Overview

The CCC Digital Key, Release 3 [CCC-DK-TS] the third in a series of releases, allows individual owners to use their mobile devices as keys to their vehicles. The specification enables:

- Security and privacy equivalent to physical vehicle keys.
- Interoperability and user experience consistency across mobile devices and vehicles.
- Vehicle access, start, mobilization, and other use cases.
- Owner pairing and key sharing with friends, with standard or custom entitlement profiles.
- Support for mobile devices with low batteries.

The Digital Key (DK) architecture uses standards-based public key infrastructure to establish end-to-end trust. Mobile devices create and store DKs in Secure Elements (SE) – embedded technology that provides a tamper-resistant secure implementation – to provide the highest-level of protection from the plethora of known hardware- and software-based attacks, including tampering, storage intrusion, cloning, and unauthorized access as well as side channel, fault injection, and many other forms of attack.

Mobile devices may act as either owner or friend devices, but the vehicle-to-device interface is the same in either role. Interoperability between mobile devices and vehicles is supported by standardizing the vehicle-to-device interface, including the NFC communication channel, the protocols, the DK structures, the BLE communication channel and the UWB-based secure ranging.

This Protection Profile written in Common Criteria (CC) language defines the security requirements for the DK Applet on an SE implementing Java Card and GlobalPlatform specifications.

1.2.1 *TOE Type*

The Target of Evaluation (TOE) is the composition of the DK Applet implementing the CCC Digital Key Release 3 specification [CCC-DK-TS] with a certified SE as illustrated in Figure 1. The TOE provides secure, tamper-proof storage of Digital Keys and processing of

authentication, encryption protocols and key generation used for owner pairing, key sharing, vehicle access and engine start. If the Digital Key Applet follows the SE-centric applet model, the DK Applet also verifies the Vehicle Public Key Certificate.

The TOE SHALL comprise at least the following components, depicted in Figure 1:

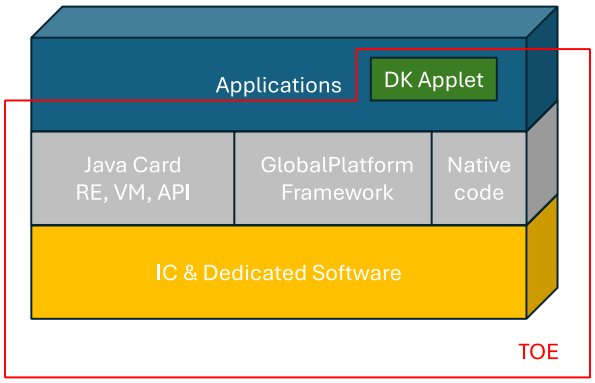
- The IC and Dedicated Software.
- The Java Card Runtime Environment (JCRE), Java Card Virtual Machine (JCVM), Java Card API, and, optionally, native code.
- The GlobalPlatform Framework for card and application management.
- The DK Applet.
- The user guide.

The TOE SHALL implement CCC Digital Key specifications [CCC-DK-TS].

Additionally, this PP requires that, at the time of the TOE evaluation, the following certificates for the SE exist and are valid:

- The certificate of the TOE's IC and IC Dedicated Software against [PP0084].
- The certificate of the TOE's Java Card Platform against [PP0099] or any [PP0099]-conformant PP such as [SEPP].

Figure 1: TOE architecture



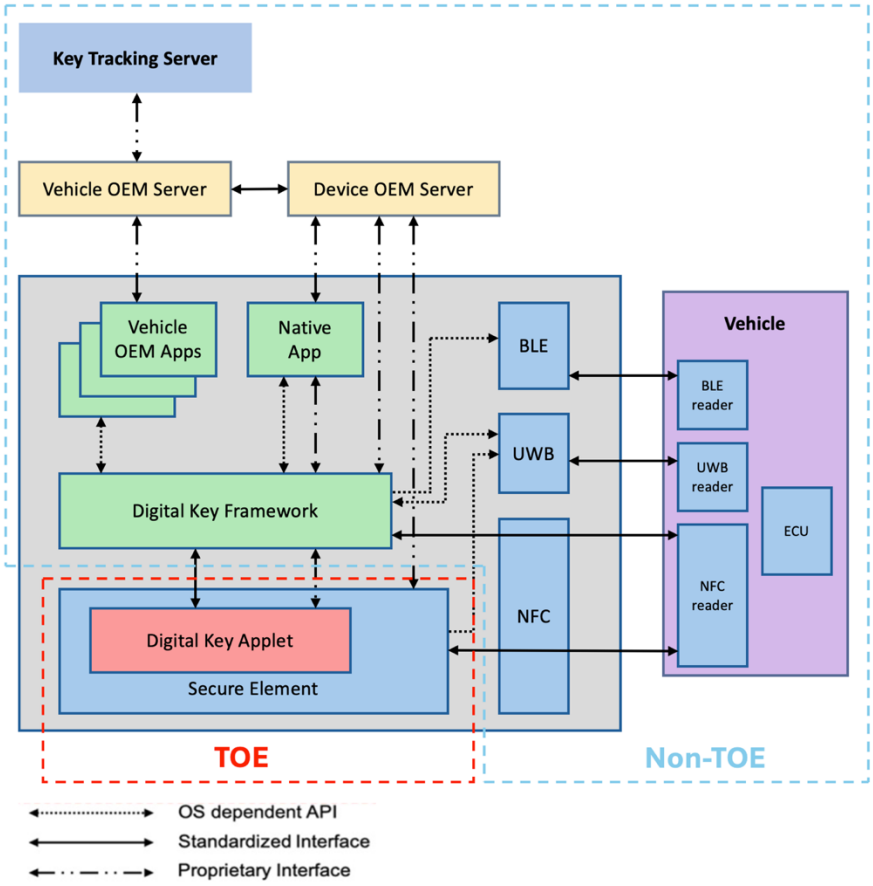
Application Note 1: The scope of the Java Card Platform may vary from product to product. At a minimum it includes the IC and Dedicated Software, the Java Card Runtime Environment (RE), the Java Card Virtual Machine (VM), the Java Card API, the Installer, and any native code. In the context of [PP0099], it may also include all or part of the GlobalPlatform Framework. In the context of [SEPP], it includes the GlobalPlatform Framework. The ST author should define the components of the actual TOE and the scopes of the two certificates.

Application Note 2: This PP does not address security aspects that are already covered in [PP0084] and [PP0099], except for the physical protection, which is explicitly required to cover the additional functionality.

1.2.2 *Available Non-TOE Hardware/Software/Firmware*

Figure 2 presents the DK system and depicts the TOE and the non-TOE components. The region inside red discrete lines (---) corresponds to the TOE, the region inside blue discrete lines (---) corresponds to the non-TOE components and the ash-colored box corresponds to the device.

Figure 2: TOE and non-TOE components in the DK system



Application Note 3:

- Vehicle OEMs and Device OEMs provide their own PK infrastructure and the root CA each in their context. The Vehicle OEM signs Device OEM root CA certificate.
- The proprietary interface depicted in Figure 2 between the Device OEM Server and the Secure Element is supplied by the Device OEMs. This proprietary interface is outside of the responsibility and authority of CCC. Nevertheless, this interface must be assessed during an evaluation of a TOE claiming conformance to this PP.

The TOE interacts with the following non-TOE components to fulfil its functionality: Digital Key Framework, Device NFC, Device OEM Server and optionally with the Device UWB module. The non-TOE components depicted in Figure 2 are described in the rest of the section.

1.2.2.1 Device

A device supports the following main functions:

- Contactless transactions to lock/unlock vehicle and start the engine
- Configurable user authentication (e.g., passcode).

Devices can take on the role of an ‘owner device’ or a ‘friend device’.

1.2.2.1.1 Owner Device

An owner device supports the following main features: transaction processing, owner pairing, Digital Key sharing (sender) and Digital Key termination. It stores the certificates that are necessary for owner pairing and Digital Key sharing and terminates the Shared Keys.

1.2.2.1.2 Friend Device

A friend device supports the following main features: transaction processing, Digital Key sharing (receiver) and key termination. It stores the certificates that are necessary for Digital Key sharing and sends the termination attestation to the Vehicle OEM Server.

A friend device can host a Digital Key shared by an owner, but it cannot share this key with any other device. A friend device may have restricted access rights to the vehicle compared to the owner.

1.2.2.2 Device OEM Server

The Device OEM Server loads and installs the instance of the Digital Key Applet (if necessary). It provides and updates the certificates in the Device.

1.2.2.3 Digital Key Framework

The Digital Key Framework implements the following main features: owner pairing, Digital Key sharing and management. It provides common Digital Key service functionality to Vehicle OEM Apps via a set of APIs specific to the Device Operating System (OS).

1.2.2.4 Native App

The Native App provides device-native UI such as Digital Key creation, Digital Key termination and deletion, and Digital Key enable/disable. It also allows to display the list of all issued owner/friend Digital Keys.

1.2.2.5 Vehicle OEM Apps

Vehicle OEM Apps are optional. The main features of these apps are supported natively by the device. They may support the same features as the Native App plus Vehicle OEM-specific features.

1.2.2.6 Device NFC, BLE and UWB modules

The NFC module and the optional BLE and UWB modules of the device allow to communicate with the Vehicle. The NFC interface is routed directly to the Digital Key Applet, providing a communications path that is protected from, and that operates independently of, the rest of the mobile device. On the other hand, the optional BLE interface is not routed directly to the Digital Key Applet. The communication is indirect through the Digital Key Framework.

The DK Applet may optionally support secure ranging functionality through the UWB module after establishing the ranging session through the BLE module.

It should be noted that while the NFC module is always required in the device, the BLE and UWB modules are optional. The possible configurations are:

- NFC module alone,
- NFC and BLE modules,
- NFC, BLE and UWB modules.

The UWB module cannot be used without the BLE module.

1.2.2.7 Vehicle

The Vehicle determines if the owner/friend device is eligible for the Digital Key service before allowing owner pairing or accepting a friend key shared by the owner device. It verifies the authenticity of the device.

1.2.2.8 Vehicle ECU

The ECU of the Vehicle performs the security functions for managing access and starting the Vehicle.

1.2.2.9 Vehicle NFC, BLE and UWB Readers

The Vehicle's NFC and BLE readers allow to communicate with the device, e.g. for owner pairing and Digital Key transactions (lock/unlock, engine start, etc.) with owner/friend devices. The Vehicle's UWB reader allows the secure ranging functionality.

1.2.2.10 Vehicle OEM Server

The Vehicle OEM Server provides the following main functionality:

- Backend for external management of the Vehicles.
- Hosting of owner account that links to the owner's vehicle(s).
- Management of Digital Key service subscriptions.
- Online attestation to the vehicle (when online) so that shared friend Digital Keys are accepted by the vehicle in the first friend transaction.
- Secure channel with the vehicle.

1.2.2.11 Key Tracking Server (KTS)

The Key Tracking Server records relevant data to be able to assign a tracked Digital Key for a vehicle to a device. The KTS is likely to be managed by the Vehicle OEM.

1.2.3 *TOE Security Features*

This section summarizes the security features of the TOE[1], defined in the Technical Specification documentation [CCC-DK-TS].

1.2.3.1 Secure Owner Pairing

The owner pairing flow is operated by the Digital Key Framework and DK Applet. The Digital Key Framework uses the APDU commands to manage the configuration of the Digital Key,

[1] The security features of the IC conformant to [PP0084] and of the Java Card Platform conformant to [PP0099]are not listed in this PP.

protected by the SE. The SE provides the root of trust, which is the starting point in the trust chain.

The vehicle can select the Digital Key Applet over NFC using its AID and to select the Digital Key Framework using the corresponding AID. The NFC controller may be reconfigured for changing the routing of the communication from the SE to the Digital Key Framework and vice versa, based on the selected AID.

The vehicle can also select the Digital Key Applet over BLE if the TOE supports secure ranging.

A new owner device pairing flow, or owner device change, does not imply an implicit unpairing, i.e., a new device owner pairing flow only changes the owner's key. Existing shared/friend keys that are already paired, and vehicle public keys, are not necessarily impacted. The DK Applet instance SHALL be available on the SE before the time of owner pairing execution. Note that the Device OEM CA root key is never stored in the SE - Either variant 1 (SE root of trust based on CASD) or variant 2 (SE root of trust based on the Security Domain associated with the DK Applet) must be implemented according to the specifications.

1.2.3.2 Secure Standard Transaction

A secure channel between vehicle and device is initiated by generating ephemeral key pairs on the vehicle and device sides. Using a key agreement method, a shared secret can be derived on both sides and used for generation of a shared symmetric key, using Diffie-Hellman and a key derivation function.

The ephemeral public key generated on the vehicle side is signed with the vehicle's private key (vehicle_SK). This results in an authentication of the vehicle by the device. From the device's perspective, this guarantees that no privacy-sensitive data can be leaked by a MITM attack. This principle also allows the device to transmit data to the vehicle without any possibility of leakage by a passive or active eavesdropper.

Finally, the device uses the established secure channel to encrypt its public key identifier along with the signature computed on a vehicle's data-derived challenge and some additional application-specific data. This verification of the device's signature by the vehicle allows the vehicle to authenticate the device.

1.2.3.3 Secure Fast Transaction

The device generates a cryptogram based on a secret previously shared during a standard transaction, and this allows the vehicle to authenticate the device. Optionally, a secure channel between vehicle and device is established by deriving session keys from a secret previously shared during a standard transaction and from the ephemeral keys. The ability of the vehicle to establish the secure channel authenticates the vehicle to the device.

The fast transaction protocol is intended to provide the following properties:

- Device authentication or mutual authentication.
- Integrity and confidentiality.
- Tracking resilience.

1.2.3.4 Secure Check Presence Transaction

The presence check transaction protocol is intended to provide the following properties:

- Vehicle authentication.

- Device identification.
- Integrity and confidentiality.
- Tracking resilience.

The mechanism is like the standard transaction mechanism described in section 1.2.3.2, except that the device signature is not sent to the vehicle, and user authentication is disabled. The goal is to allow verification of device presence near the vehicle without requiring user authentication, while preventing tracking.

1.2.3.5 Secure DK Sharing

The DK sharing flow is operated by the Digital Key Framework and DK Applet running on the owner and the friend device. During the sharing flow the owner device creates a sharing invitation that is sent to the friend device. Based on this invitation, the friend device creates a DK in the DK Applet and generates a key signing request, which is signed by the friend private key. The owner device then creates an attestation package over the friend public key from the key signing request and optionally exports and includes an immobilizer token from the owner DK confidential mailbox. At the end of this flow, the friend's private mailbox stores the attestation package, in which the friend's public key is signed by the owner private key along with a set of entitlements. The optional immobilizer token is stored in the friend`s confidential mailbox.

The attestation package is verified by the vehicle OEM KTS server during key tracking and at the vehicle's first transaction with the friend device.

1.2.3.6 Key Termination & Suspension

Unlike physical keys and key fobs, Digital Keys may be easily terminated or suspended by friend devices, owner devices, vehicles, and/or OEM Servers. Termination is permanent and requires the sharing of a new Digital Key to restore access, while suspension is temporary and simply disables a Digital Key until it is resumed.

1.2.3.7 Secure Application Management

The TOE offers additional security services for application management, relying on the GlobalPlatform Framework:

- The SE Issuer is the main entity authorized to manage applications (loading, instantiation, deletion) through a secure communication channel with the SE.
- DK Applet Provider personalizes the DK Applet and the associated Security Domain (SD) in a confidential manner. The DK Applet Provider is usually the SE Issuer. The Security Domain keysets are used to establish a Secure Channel between the TOE and external entities (e.g. Device OEM server). If the SE Issuer is not the DK Applet Provider, the Security Domain keysets are unknown to the SE Issuer.
- The services provided by the Controlling Authority Security Domain (CASD) allows the implementation of the SE root of trust.

1.2.3.8 (Optional) UWB-based Secure Ranging Support

Optionally, the DK Applet supports the secure ranging. The device establishes a secure ranging session, communicating through the UWB module. This session allows the vehicle to determine the distance between itself and the device. Should the device be within the configured range of the

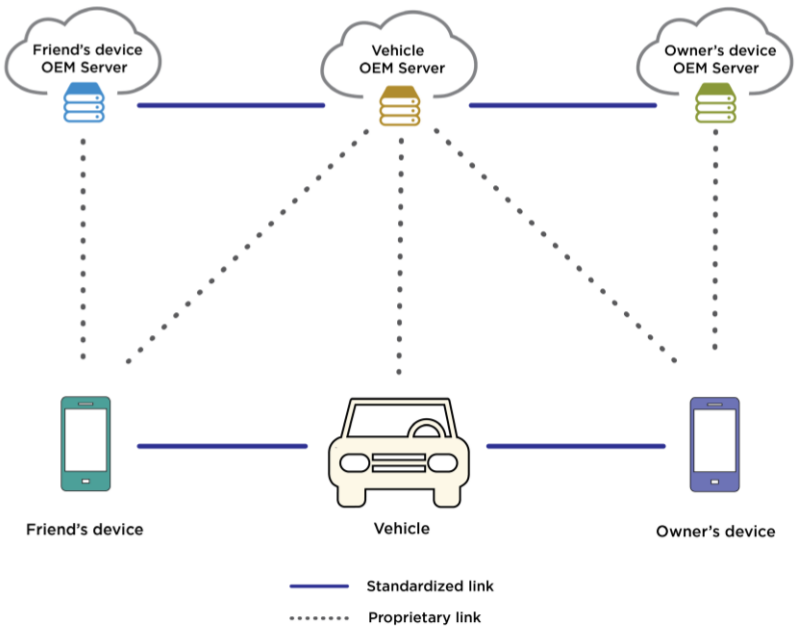
vehicle, the latter may allow the passive entry and the passive start. This requires the TOE to ensure its binding with the UWB module. The ranging session is based on a SE-based secure channel with the URSK (UWB Ranging Secret Key) that must be protected in integrity, authenticity and confidentiality.

1.2.4 *TOE Usage*

The Digital Key system enables mobile devices to store, authenticate, and share Digital Keys for vehicles in a secure, privacy preserving way that works everywhere, even when the mobile device's battery is low. It is an important service for the automotive industry, enabling a significantly improved vehicle access experience that builds consumer confidence through its ease of use, convenience, security and privacy protection, and extensive capability. The CCC, representing most of the global automotive and mobile device industries, enables the Digital Key's broad cross-industry support and facilitates the coordination of mobile Device OEMs and Vehicle OEMs to provide a consistent user experience by increasing interoperability and reducing market fragmentation.

Figure 3 shows a high-level view of the DK system architecture.

Figure 3: DK system architecture



The mobile device's Native App allows consumers to use and manage Digital Keys without any extra apps, and its Digital Key Framework enables developers to build custom apps that provide enhanced services and vehicle-specific features. Mobile devices and vehicles interact with their respective OEM Servers to share and manage Digital Keys across mobile device and vehicle platforms. The DK system ensures that the owner or an allowed friend can access their vehicle even when neither the mobile device nor the vehicle have Internet connectivity, while still allowing OEMs to add features that require Internet connectivity for certain operations.

The different use cases of the DK Applet include:

- **Owner Pairing:** The owner device's role consists in identifying the device hosting the owner key for a given vehicle. This enables any mobile device that meets the technology and security requirements of Digital Key to be paired as an owner device with a vehicle. Each vehicle can be associated with only one owner device; an owner device has full authority over the paired vehicle and all associated Digital Keys. A given device can host several owner keys (in case someone owns multiple cars) but for a given car there is a single owner device.
- **Vehicle Access/Engine start**: Digital Keys may be used to access a vehicle, start the engine, mobilize the vehicle, or authorize any other operation e.g. by placing the mobile device near an NFC reader, without requiring any interaction with a user interface of the mobile device (e.g., an app). For this operation to take place, the vehicle and the device SHALL mutually authenticate first, and the vehicle verifies that the mobile device's Digital Key authorizes the requested operation. Mobile devices may also perform user authentication by requesting the user to insert a passcode or use a biometric authentication mechanism. These operations may also be performed through BLE communication if the TOE supports the UWB-based secure ranging functionality. Examples of operations include unlocking of the doors, starting the engine, etc.
- **Sharing Digital Keys**: Devices that can use Digital Keys can be owner devices as well as friend devices. There is no limit to the number of friend devices with Digital Keys for a given vehicle, but friend devices may not share access with other friend devices. An owner device shares a Digital Key with a friend device by sending a sharing link to the friend device (e.g., via SMS). When the Digital Key is accepted (e.g., by tapping the sharing link), the friend device creates a Digital Key with the appropriate parameters (vehicle, entitlements, etc.), the Digital Key Framework establishes a secure communications channel between the two devices, through which the owner device signs (approves) the friend device's Digital Key (public key), and necessary signatures (approvals) are obtained from cloud services (e.g., Vehicle OEM Servers). To ensure that the shared Digital Key is usable only by the intended recipient, the owner may optionally provide them with one or more sharing passwords and/or PINs communicated on a different channel than the sharing link.
- **Termination/Suspension of Digital Keys**: This feature enables the user (owner or friend) to terminate their Digital Key or to suspend it during various situations such as selling of the vehicle, the mobile device being stolen/lost, a security breach on the mobile device, or even when the owner decides not to share the keys anymore with a friend. Digital Keys may be terminated or suspended at any time. Termination is permanent and requires the sharing of a new Digital Key to restore access, while suspension is temporary and simply disables a Digital Key until it is resumed.
- **Remote on-demand features called RKE function**: Digital Keys may be used for on-demand features (e.g. lock, unlock and alarm) from within Bluetooth LE range. Before triggering a desired action, the device shall perform user authentication. If the user is successfully authenticated, the device shall request a challenge from the vehicle. Then, the device shall include the received challenge to the arbitrary data containing the requested RKE function and create a device signature using its private key of the Digital Key corresponding to the targeted vehicle. These operations are available on TOE

supporting BLE. Examples of operations include doors or trunk opening/closing, panic alarm muting/triggering, windows opening/closing, etc.

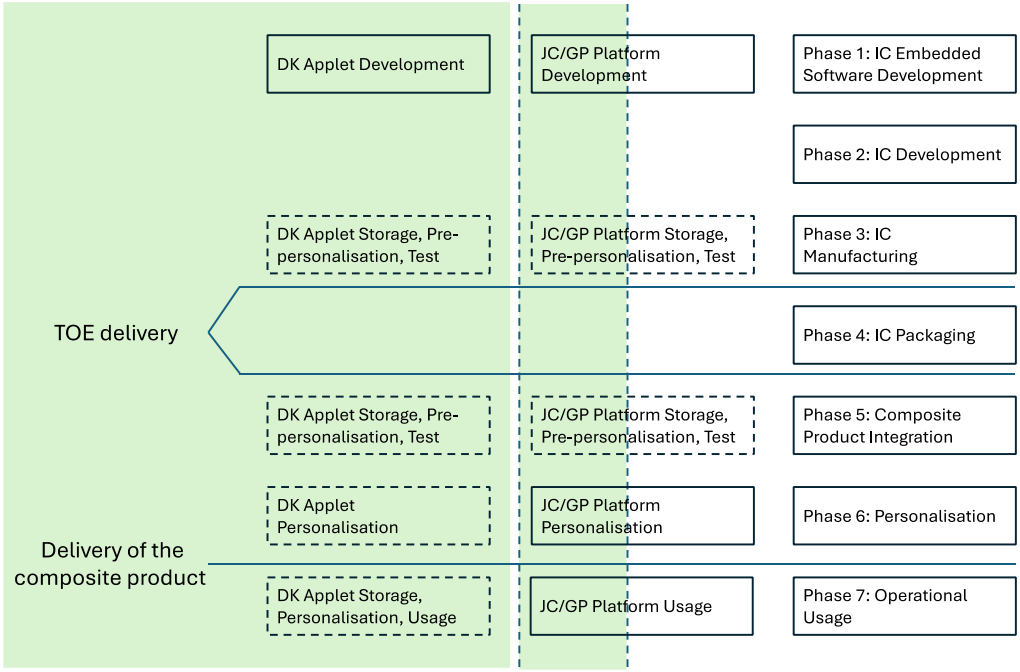
1.3 TOE Lifecycle

The TOE lifecycle follows [PP0084] and is part of the product lifecycle, which goes from development to its usage by the final user.

The typical lifecycle phases are those detailed in Figure 4:

- The rightmost column represents a generic product composed of an IC and Embedded Software, as defined in [PP0084].
- The middle and leftmost columns map the generic Embedded Software to the Java Card and GlobalPlatform software and the DK Applet, respectively.
- The rightmost and middle columns comply with the TOE and product lifecycle defined in [PP0099].
- The white boxes depict components and activities that are assumed already certified.
- The green boxes depict components and activities that are covered in this PP.
- The dotted boxes indicate the possible phases where the activities take place. For instance, the loading and personalization of the DK Applet can occur in Phase 7.
- The exact content of the components belonging to the green-dotted region is implementation-dependent and may be empty if the entire Java Card/GlobalPlatform software has already been certified.
- The delivery of the TOE can occur after Phase 4, in which case the environments associated with these phases are also part of the evaluation.

Figure 4: TOE Lifecycle



We refer to [PP0084] for the complete description of Phases 1 to 7 shown in Figure 4:

- Phases 1 and 2 compose the development of the product, i.e. the Embedded Software including the Java Card System, the GlobalPlatform Framework and the DK Applet, and the IC and IC Dedicated Software.
- Phases 3 and 4 correspond to IC manufacturing and packaging, respectively. Some IC pre-personalisation steps may occur in Phase 3.
- Phase 5 concerns the (partial) embedding of the software components within the IC, which can be completed in Phases 6 and 7.
- Phase 6 is dedicated to the product personalisation.
- Phase 7 is the product operational phase (usage phase).

The lifecycle of the Embedded Software is composed of different activities, some of which can happen in different phases, as shown by the doted boxes in Figure 4:

- Development.
- Storage, pre-personalisation, and test.
- Personalization and test.
- Final usage.

Let us assume, for example, that the entire Java Card / GlobalPlatform software has been already certified and therefore the green-dotted region is empty:

- Phase 1 consists of the DK Applet Development. This could happen in parallel to the Java Card and GlobalPlatform platform development and covers the DK Applet conception phase, design, implementation, testing and its documentation. The development of the DK Applet SHALL be carried out in such a way that it is conformant with the Java Card and GlobalPlatform platform guidelines. The development SHALL take place in a controlled environment. This is important to guarantee the integrity of the developing elements and to ensure non-disclosure of sensitive/confidential data. The evaluation of a product against this PP SHALL include the DK Applet development environment. As an exception, the delivery of the DK Applet for integration could take place in complete form or in parts during either of the two phases: Phase 3 (IC Manufacturing) or Phase 5 (Composite Product Integration). Otherwise, the typical delivery of the DK Applet happens post-issuance during Phase 7 (Operational Usage). Delivery and acceptance procedures SHALL guarantee the authenticity, the confidentiality and integrity of the exchanged pieces. The evaluation of a product against this PP SHALL include the delivery process.
- Phases 3 and 4 consist of the IC Manufacturing and Packaging. This IC must be certified (e.g. against [PP0084]).
- During Phase 5, the product integrator may perform the storage and pre-personalisation of the DK Applet and may also conduct tests. The product integration environment SHALL protect the confidentiality and integrity of the DK Applet and of any related material being used including testing material.

- The personalisation of the DK Applet may take place during Phase 6. Thus, the personalisation environment SHALL ensure the confidentiality and integrity of any associated data or material being used.
- During Phase 7, the operational usage of the DK Applet takes place. The DK Applet may be loaded into the product. The DK Applet final usage environment is that of the SE where the DK Applet is embedded in.

Application Note 4:

- The ST author shall describe the lifecycles of the TOE and of the composite product, and indicate the phases where the development, storage, (pre-)personalization and test are performed for all the components of the Embedded Software that are not covered by an existing certificate. The ST author shall indicate in which phase of the life cycle the TOE is delivered.

1.4 TOE evaluation

This PP applies to TOEs that include certified IC and Java Card Platform (cf. section 1.2.1), which permits the reuse of the pre-existing evaluation results for the composite evaluation of the TOE.

Composite evaluation SHALL meet the _COMP assurance requirements defined in [CC3] and [CEM]. The following composite evaluation rules apply:

- The TOE SHALL be evaluated at a minimum at the assurance level EAL4 augmented with ADV_DVS.2, ALC_FLR.2 and AVA_VAN.5 claimed by this PP, which corresponds to the minimum assurance level of the underlying Java Card Platform. A higher assurance level may be claimed by the Security Target provided this is supported by the certificate of the Java Card Platform.
- The composite TOE integrating the certified underlying Java Card Platform SHALL satisfy the objectives on the integration environment which are enforced through the platform user guidance (AGD_OPE) as defined in [PP0099].

Application Note 5: In the composite evaluation, the Java Card Platform, including the IC, constitutes the base component, and the remaining parts of the TOE Embedded Software, including the DK Applet, constitute the dependent component. The ST author shall describe the base and dependent structure of the TOE and shall provide the statement of compatibility with the ST of the base component.

2 CONFORMANCE CLAIMS

2.1 CC Conformance Claim

This PP claims conformance to the CC:2022 Revision 1 [CC]. This PP is CC Part 2-conformant and CC Part 3-conformant.

2.2 PP Conformance Claim

This PP does not claim conformance to any other PP.

2.3 Package Claim

2.3.1 *Functional Package Claim*

This PP claims conformance to the Functional Package for UWB-based Secure Ranging (FP UWB-SR).

FP UWB-SR is optional in the sense that it applies if and only if the TOE implements the UWB-based secure ranging support functionality. Additionally, the UWB-SR package contains optional cryptographic SFRs of conditional type, which shall be claimed in any PP-conformant Security Target for a TOE that implements the cryptographic functionality specified in those conditional SFRs.

2.3.2 *Assurance Package Claim*

This PP claims conformance to the following assurance packages defined in [CC5]:

- EAL4 augmented with ALC_DVS.2 “Sufficiency of security measures”, ALC_FLR.2 “Flaw reporting procedures” and AVA_VAN.5 “Advanced methodical vulnerability analysis”.
- Composite product package COMP.

2.4 Conformance Statement

This PP requires strict conformance of STs and PPs claiming conformance to it.

3 SECURITY PROBLEM DEFINITION

3.1 ASSETS

Assets are entities that the owner of the TOE presumably places value upon. Assets are expected to be directly protected by the TOE.

The assets are divided in two groups. The first one, defined in Table 3-1, contains the data created by and for the user (User data) and the second one, defined in Table 3-2, includes the data created by and for the TOE (TSF data).

Table 3-1: User Data Assets, Description and Sensitivity

Assets (User Data)	Description	Sensitivity (C, I, A)2
D.OWNER_DATA	Information related to the device owner or owner's friend like phone number, location, device usage or other (Personally Identifiable Information) PII on the device side.	C, A
D.KEY_OPTIONS	Options to the keys like access rights, key validity and other fields which are not specified in more detail.	C, I, A
D.MESSAGES_EXCHANGES	Data and commands exchanged between the components of the systems in plain form.	C, I, A

Table 3-2: TSF Data Assets, Description and Sensitivity

Assets (TSF Data)	Description	Sensitivity (C, I, A)
D.KCMAC_KEY	The D.KCMAC_KEY is a derived symmetric key used to calculate cryptograms. It is part of owner DK secret / Friend DK secret. This key is used for secure channel opening during the fast transactions.	C, I, A
D.IMMOTOKEN	Vehicle cryptographic material that is provisioned by some vehicles (confidential mailbox) at the DK creation and that might be requested back during the fast or standard transaction to allow engine start.	C, I, A
D.SECRET_SHARED_KEY	A shared symmetric key generated on both the vehicle and the device sides during owner pairing (standard transaction) using key agreement method (Kdh). Kdh is a shared key computed using Diffie-Hellman according to [BSI TR-03111] Section 4.3 indications.	C, I, A
D.GP_CODE	The code of the GlobalPlatform framework on the secure element.	I, A
D.SE_MNGT_DATA	The data of the secure element management environment, like for instance, the identifiers, the privileges, life cycle states, the memory resource quotas of applets and security domains.	I, A
D.DK_APPLET_CODE	The source code of the DK Applet.	I, A

[2] C = Confidentiality, I = Integrity, A = Availability

Assets (TSF Data)	Description	Sensitivity (C, I, A)
D.LONG_TERM_KEY	Symmetric long-term key that is used to derive encryption and MAC session keys. It is stored in NVM on both vehicle's and device's sides.	C, I, A
D.DEVICE_KEY_PAIR	Asymmetric long-term key pair generated at applet creation that is used to sign command payloads. It is fully stored in NVM on the device's side, and the public key is stored in NVM on the vehicle's side.	C, I, A
D.APPLET_ROOT_KEY	The root of trust of the SE storage that is used to bind the DK Applet and keys to the SE. (SE_root_SK/PK).	C, I, A
D.SESSION_KEYS	Temporary key material used to protect data in the DK communication protocol. This includes Kenc, Kmac and Krmac.	C, I, A
D.SEC_ATTRIBUTES	The runtime security data including all identifiers, context of execution.	C, I, A
D.OWNER_DK_SECRET	General term for Owner DK secret and/or Friend DK secret and/or IMMOTOKEN. **Application Note 6:** • Public keys are mutually exchanged through pairing of the owner device to the vehicle. The owner can then authorize the use of Digital Keys by friends and family members, by signing their public keys. DK secret corresponds to the private key associated to these public keys. KCMAC is a symmetric key derived from the symmetric long-term key according to [RFC 5869] Section 2. • There is one Digital Key per vehicle. During owner pairing, all Digital Key elements are provided by the vehicle and transferred to the device.	C, I, A
D.OWNER_DK_DATA	Information attached to the digital key concept except the DK secret. E.g. mailbox data, public DK key.	I, A
D.DK_API_DATA	Data of the DK Applet API, such as the contents of its private fields.	C, I, A
D.RNG	Generated random numbers. **Application Note 7:** In addition to the confidentiality and integrity properties, unpredictability, sufficient entropy, and forward secrecy are to be considered for this asset.	C, I, A

Table 3-3 is intended to highlight where each cryptographic key could be stored.

Table 3-3: Storage of Cryptographic Keys

Cryptographic keys	Storage
D.KCMAC_KEY	DK Applet within SE & Vehicle-ECU module
D.IMMOTOKEN	DK Applet within SE & Vehicle-ECU module
D.SECRET_SHARED_KEY	DK Applet within SE & Vehicle-ECU module
D.LONG_TERM_KEY	DK Applet within SE & Vehicle-ECU module
D.DEVICE_KEY_PAIR	DK Applet within SE (both public and private keys) & Vehicle-ECU module (public key only)
D.APPLET_ROOT_KEY	SE
D.SESSION_KEYS	DK Applet within SE & Vehicle-ECU module
D.OWNER_DK_SECRET	DK Applet within SE

3.2 THREATS

Table 3-4 presents the threats to the assets against which specific protection within the TOE or its environment is required. The threats that are countered exclusively environment by the operational environment are mapped to security objectives for the TOE operational in Table 4-4. Several groups of threats are distinguished according to the means used in the attack. The classification is also inspired by the components of the TOE that are expected to counter each threat.

Table 3-4: Threats and Related Assets

Threats	Description	Related Assets
T.DK_PHYSICAL	An attacker, with physical access to the TOE, may attempt to access the DK sensitive assets when it is stored. These physical access threats may involve attacks, which attempt to access the device through external hardware ports, impersonate the user authentication mechanisms, through its user interface, and also through direct and possibly destructive access to its storage media.	All
T.UNAUTHORIZED_SE_MNG	The attacker performs unauthorized secure element management operations (for instance impersonates one of the actors represented on the secure element) in order to take benefit of the privileges or services granted to this actor on the secure element such as fraudulent: • load of a package file • installation of a package file • extradition of a package file or an applet • personalization of an applet or a Security Domain • deletion of a package file or an applet • privileges update of an applet or a Security Domain Directly threatened	D.SE_MNGT_DATA D.DK_APPLET_CODE

Threats	Description	Related Assets
T.LIFE_CYCLE	An attacker accesses to an application outside of its expected availability range thus violating irreversible life cycle phases of the application (for instance, an attacker re-personalizes the application).	All TSF data
T.IT_DISCLOSURE	Attacker gets unauthorized access to the Immobilizer Token during storage, deletion or processing	D.IMMOTOKEN
T.DK_DISCLOSURE	Attacker is predicting (lack of randomness) or recovering the Digital Key from a Device in order to provision it on his own device and enter the paired Vehicle. The attacker could target any event in the key lifetime (creation/use/access/storage) and particularly events that require the secret material to be transferred from one memory location to another.	All TSF data
T.FLAW_SW	Attacker loads a malicious or exploitable code into the software of a component of the DK ecosystem in order to change the behaviour of the DK feature, to attempt to exfiltrate restricted data, or to gain additional privilege into the system of the component. For instance, a malicious DK Sharing request is produced to the attacker's device.	D.DK_APPLET_CODE D.APPLET_ROOT_KEY
T.NON_REVOKED	An attacker could benefit of preventing the processing of a revocation request in order to keep using a provisioned DK (on a stolen device for instance) to access and steal a Vehicle. For instance, revocation would be issued in order to prevent a component (unit or family or maker or SW version…) from participating in the DK ecosystem when it has shown vulnerabilities. If the other components are not verifying the revocation status of presented certificates, then it would be possible for an attacker to use the vulnerable component to perform its attacks.	D.GP_CODE D.SE_MNGT_DATA D.DK_APPLET_CODE D.LONG_TERM_KEY D.DEVICE_KEY_PAIR D.APPLET_ROOT_KEY D.SEC_ATTRIBUTES D.DK_API_DATA
T.DATA_BREACH	Data breach could also take place during the execution of a transaction using NFC (or BLE if the module is present in the device) communication. The attacker might be able to obtain confidential data as well as shared keys.	D.OWNER_DK_SECRET D.OWNER_DK_DATA D.OWNER_DATA
T.KTS_DATA_LEAK	Unnecessary confidential information of the device's user is sent to the Key Tracking Server, leaking the Vehicle owner’s confidential data.	D.OWNER_DK_DATA D.OWNER_DATA
T.RETRIEVE_SECRET-SHARED-KEY_DKA	Attackers retrieve the previous Secret shared Key generated (Standard or Fast transaction) **Application Note 8:**	D.SECRET_SHARED_KEY

Threats	Description	Related Assets
	• Dump NVM SE memory (Physical attacks, Logical attacks (Malware) or Combine attacks) • Exploit chip power consumption or electromagnetic radiation leakages (Side channel attacks) • Flipping a bit allowing to bypass a security mechanism and providing access to memory (Fault injection attacks) If the attack succeeds, then the attacker can open a secure channel during fast transactions with the owner/friend Device to retrieve the immo token and data to lock/unlock. Next the attacker can open new secure channel to lock/unlock the Vehicle.	
T.RETRIEVE_KCMAC-KEY_DKA	Attackers retrieve Kcmac: **Application Note 9:** Dump the NVM SE memory If the attack succeeds, then the attacker can open secure channel during fast transactions with the owner/friend Device to retrieve the immo token and data to lock/unlock. Next the attacker can open new secure channel to lock/unlock the Vehicle.	D.KCMAC_KEY
T.RETRIEVE_SESSION-KEYS_DKA	Attackers retrieve Kenc / Kmac / Krmac **Application Note 10:** Dump the NVM SE memory An attacker would need to perform the attack for each session during transactions (standard or fast). Further attacks may be needed to lock/unlock the Vehicle.	D.SESSION_KEYS
T.UPDATE_KEY_OPTIONS	Attackers modify the keys options to give all access to lock/unlock and Engine Start.	D.KEY_OPTIONS
T.UNAUTHORIZED_ACCESS_DK_ASSET	Attackers access crypto primitives using DK Applet assets (secret shared key, …) **Application Note 11:** Through relay attacks via rogue DK Applet in the Device.	D.KCMAC_KEY D.IMMOTOKEN D.SECRET_SHARED_KEY D.LONG_TERM_KEY D.DEVICE_KEY_PAIR D.APPLET_ROOT_KEY D.SESSION_KEYS
T.DEVICE_THEFT	An attacker may attempt to steal a user's device and use it to access or start the vehicle.	All
T.CA_KEY_LEAK	An attacker may attempt to steal the private key used by the device or vehicle root key. An attacker could use this key to generate fraudulent attestations.	D.APPLET_ROOT_KEY

Threats	Description	Related Assets
T.RADIO_SNIFF	An attacker may attempt to sniff the traffic between a device and a vehicle during an exchange.	D.SESSION_KEYS D.SECRET_SHARED_KEY D.SEC_ATTRIBUTES D.OWNER_DK_SECRET D.OWNER_DK_DATA D.OWNER_DATA D.DK_API_DATA D.MESSAGES_EXCHANGES
T.RADIO_MITM	An attacker may attempt to gain a MITM presence between a device and a vehicle during an exchange and might be able to modify the keys being shared.	D.SESSION_KEYS D.KCMAC_KEY D.IMMOTOKEN D.SECRET_SHARED_KEY
T.PROTOCOL_DOWNGRADE	An attacker may attempt to downgrade the protocol to an older version that has known weaknesses.	D.KCMAC_KEY D.IMMOTOKEN D.SECRET_SHARED_KEY D.SE_MNGT_DATA D.SESSION_KEYS
T.SIGN_COMPROMISE_VERIFY_COMPROMISE	Attacker manipulates creation and validation of electronic signatures	D.IMMOTOKEN D.DK_APPLET_CODE D.APPLET_ROOT_KEY D.SESSION_KEYS D.OWNER_DK_SECRET D.DEVICE_KEY_PAIR
T.DENIAL_OF_LEGITIMATE_DELETIONS	An attacker prevents a legitimate DK deletion request from the user or a backend system.	D.DK_API_DATA
T.ILLEGITIMATE_RKE	An attacker is able to sign the challenge sent by the vehicle, thus authenticating as the device, and perform RKE functions	D.DEVICE_KEY_PAIR
T.DK_SK_MODIFICATIONS	An attacker modifies DK secret keys in memory.	D.KCMAC_KEY D.IMMOTOKEN D.SECRET_SHARED_KEY D.DK_APPLET_CODE D.LONG_TERM_KEY D.DEVICE_KEY_PAIR D.APPLET_ROOT_KEY
T.RADIO_RELAY_TRANSACTION	An attacker may try to relay a transaction with radio equipment.	D.SESSION_KEYS D.SEC_ATTRIBUTES D.RNG
T.INTERNET_CONNECTIVITY_DOS	A device OEM may attempt to prevent or otherwise limit the use of a DK from a competing device OEM.	D.SEC_ATTRIBUTES D.OWNER_DATA D.DK_API_DATA D.KEY_OPTIONS
T.TIME_CHANGE	An attacker may attempt to change the time on the device or vehicle in order to enable a	D.KEY_OPTIONS

Threats	Description	Related Assets
	currently invalid key or disable a currently valid key.	
T.REPLAY_TRANSACTION	An attacker may try to replay an observed transaction to the vehicle.	D.SESSION_KEYS D.SEC_ATTRIBUTES D.RNG
T.UNAUTHORIZED_KEY_SHARING	Two or more collaborating adversaries share (copy) access credentials without the vehicle owner's consent or knowledge. E.g. • resale of car access credentials (e.g. rental/fleet car) • {regulatory, user} ban evasion • use of uncertified applications and/or devices.	D.KCMAC_KEY D.IMMOTOKEN D.SECRET_SHARED_KEY D.LONG_TERM_KEY D.DEVICE_KEY_PAIR D.APPLET_ROOT_KEY D.OWNER_DK_SECRET
T.INSTANCE_CA_DISCLOSURE	An attacker is extracting the Instance CA from the secure storage of a Device or signs its own Instance CA with a valid signature in order to provision digital keys which are not located in a certified DK Applet / DK Applet EE. DKs created by this attacker may not be secure and prone to various attacks.	D.OWNER_DK_DATA D.KEY_OPTIONS

3.3 ORGANIZATIONAL SECURITY POLICIES

Table 3-5 describes the organizational security policies to which both the TOE and its operational environment SHALL comply.

Table 3-5: Organizational Security Policies

Organizational Security Policies	Description
OSP.APPS_VALIDATION	The applications SHALL be associated with a digital signature and it SHALL be validated by a validation authority before loading it into the TOE.
OSP.OEM_SERVERS	A security policy SHALL be defined in order to ensure the security of the applications being stored on the OEM servers These policies can include access control policy, regular verification of integrity & encryption, isolation, etc. Site inspections SHALL also take place in order to ensure that the policies have been enforced as per the definitions inside the server security guidance documents.
OSP.KTS_SERVER	Policies SHALL be implemented for the data handled by the KTS server in order to ensure that unnecessary confidential data of the user may not be shared to the KTS server preventing data leakage.
OSP.OS_DOWNGRADE	A policy SHALL be put in place explaining the OS version with which the DK Applet is compatible with and also ensure that the downgraded version of the OS is not in use.
OSP.SECURE_KEY_RETRIEVE	The implemented policies SHALL ensure that the key retrieval takes place in a secure manner (secure channel) leaving the attacker from accessing the keys during a transaction.
OSP.KEY_SHARE	A policy SHALL be enforced in the servers ensuring that key sharing and distribution takes place in a secure manner.
OSP.KEY_OPTIONS	The key-options SHALL be secured against unauthorised modifications/access.
OSP.SERVER_COMMUNICATION	The communication channels established between the servers SHALL be secure. **Application Note 12:** • Sensitive data elements, where applicable, SHALL be protected with additional encryption protocols. • Server APIs SHALL be supported only over https with mutual authentication, i.e., 2-Way TLS.
OSP.PROTOCOL_FAILURE	A policy SHALL be defined in order to notify in case of a protocol failure and ensure continuity of working.
OSP.DOS_DETECT	A mechanism SHALL be implemented in order to detect the DOS attacks.
OSP.CERTIFICATE	The confidentiality & integrity of the certificate is protected & verified before installation/usage.
OSP.PKI_POLICY	A PKI policy SHALL be implemented which covers the secure management of PKI signature keys and secure operation of the PKI instances.

3.4 ASSUMPTIONS

Table 3-6 presents the assumptions on the product operational environment, after the delivery of the TOE.

Table 3-6: Assumptions

Assumptions	Description
A.USER_AUTHENTICATION	The device provides robust user authentication mechanisms to identify the DK user for performing any authorized actions such as deletion of keys, keys sharing etc.
A.USER_PRIVACY_CONSENT	Privacy consent SHALL be asked to the user before sharing any private data of the user to the server/other devices.
A.CERTIFICATION_REVOCATION_SET	Upon request of revocation of any certificate part of a digital key certificate chain (vehicle side or device side), the ecosystem/Certificate authority informs the relevant parties concerning the revocation.
A.OEM_ADMIN	Administrators of the OEM server are trusted people. They have been well trained to use and administrate the server securely. They are well aware of the sensitivity of the assets the server deals with and also the responsibilities they have to carry out.
A.PRODUCTION_ENV	The production environment SHALL be trusted and secure (prevents attacks from internal attackers).
A.DEVICE_OEM	The Device OEM is a trusted actor who has full control on the content of the SE. It is the responsibility of the Device OEM to ensure that the DK Applet that is deployed has been certified following the CCC Certification Program.
A.OS_CLOCK	The software uses a reliable clock for the proper functioning of the clock.
A.CAR_LOCATION	A locked device never provides information on vehicle location, which it can access.
A.DEVICE_OEM_INSIDER	It is assumed that an insider with access to the device OEM server will not make security changes to the Digital Key content or configuration (for example: may attempt to steal an owner's key or issue new keys)
A.SHARING_MASQUERADE	It is assumed that an attacker will not masquerade as an owner's friend using social engineering or other means and causes the owner to unwittingly share a key with the attacker. Attackers may not also masquerade as owners to get friends to reveal their identity to an attacker.
A.DEVICE_SAFETY	The device is assumed to be protected by the owner from getting stolen as this could lead to unauthorised access to the keys or vehicle by an attacker.
A.REPLAY	It is assumed that the device is protected against replay attacks within the communication protocol such as on NFC or BLE signals.
A.RADIO_RELAY	It is assumed that the device ensures protection against a relayed transaction with radio equipment attack.
A.OEM_SERVER_SECURITY	It is assumed that the Device OEM Server and the Vehicle OEM Server are hosted in secure data centers.
A.VEHICLE_ROOT_KEY	The vehicle root key is protected by security measures in the operational environment that ensure its confidentiality

4 SECURITY OBJECTIVES

4.1 SECURITY OBJECTIVES FOR THE TOE

Table 4-1 presents the security objectives for the TOE.

Table 4-1: TOE Security Objectives

Security Objectives	Description
O.SE_MANAGEMENT	The TOE SHALL provide secure element management functionalities (loading, installation, extradition, deletion of applications) in charge of the life cycle of the whole DK Applet and installed applications (applets).
O.IMMO_TOK_CONFID	The TOE SHALL ensure the confidentiality of the Immobilizer Token during storage (data at rest), deletion, processing. The Execution Environment SHALL prevent other applets from accessing the DK Applet secret data.
O.IMMO_TOK_INTEG	The TOE SHALL ensure the integrity of the Immobilizer Token during storage (data at rest), deletion, processing. The Execution Environment SHALL prevent other applets from modifying the DK Applet secret data.
O.DK_CONFID	The TOE SHALL ensure the confidentiality of the assets (to be protected in Confidentiality - including cryptographic keys) of the Digital Key during generation, usage (strong cryptography operations), storage, deletion, such that those that are never known outside the DK Applet within its DK Applet EE.
O.DK_INTEG	The TOE SHALL ensure the integrity of assets (to be protected in Integrity - including cryptographic keys) of the Digital Key when it is generated, used, deleted and stored.
O.LONG_TERM_KEY_CONFID	The TOE SHALL ensure the confidentiality of the Long Term key.
O.LONG_TERM_KEY_INTEG	The TOE SHALL ensure the integrity of the Long Term key.
O.DEVICE_KEY_PAIR_CONFID	The TOE SHALL ensure the confidentiality of the device key pair
O.DEVICE_KEY_PAIR_INTEG	The TOE SHALL ensure the integrity of the device key pair
O.SEC_SHARED_KEY_CONFID	The TOE SHALL ensure the confidentiality of the Secret Shared key.
O.SEC_SHARED_KEY_INTEG	The TOE SHALL ensure the integrity of the Secret Shared key.
O.KCMAC_KEY_CONFID	The TOE SHALL ensure the confidentiality of the Kcmac key.
O.KCMAC_KEY_INTEG	The TOE SHALL ensure the integrity of the Kcmac key.
O.SESSION_KEYS_CONFID	The TOE SHALL ensure the confidentiality of the session keys.
O.SESSION_KEYS_INTEG	The TOE SHALL ensure the integrity of the session keys.
O.ATTESTATION_ON_DELETION	The TOE SHALL ensure that it creates a deletion attestation for the requested key, and that it is securely deleted before the attestation is transferred to the requesting party.
O.RANDOMNESS	Only random number generators (RNG) generating sufficient entropy[3] SHALL be used in the TOE.

[3] Please refer to standards such as the NIST SP800-90B or the AIS20/31 for an accurate definition of "Sufficient Entropy"

Security Objectives	Description
O.IC_SUPPORT	The TOE SHALL provide protection against manipulation of the TOE (including its software and TSF data), the Security IC Embedded Software and the user data of the Composite TOE. This includes protection against: • reverse-engineering (understanding the design and its properties and functions), • manipulation of the hardware and any data, as well as • undetected manipulation of memory contents. (see O.Phys-Manipulation[PP0084]). The TOE SHALL provide protection against disclosure/reconstruction of user data while stored in protected memory areas and processed or against the disclosure of other critical information about the operation of the TOE (see O.Phys-Probing [PP0084]).
O.RECOVERY	The TOE SHALL ensure its correct operation. The TOE SHALL indicate or prevent its operation outside the normal operating conditions where reliability and secure operation has not been proven or tested. This is to prevent malfunctions. Examples of environmental conditions are voltage, clock frequency, temperature, or external energy fields. The TOE SHALL be able to recover to a stable secure state. (see O.MALFUNCTION [PP0084]).
O.OS_SUPPORT	The TOE SHALL provide protection against disclosure of confidential data stored and/or processed in the Security IC - by measurement and analysis of the shape and amplitude of signals (for example on the power, clock, or I/O lines) and - by measurement and analysis of the time between events found by measuring signals (for instance on the power, clock, or I/O lines). (see O.Leak-Inherent [PP0084]).
O.FAST_TRANSACTION_AUTH	The TOE SHALL guarantee at least a secure Device authentication to the Vehicle (Fast Transaction).
O.STD_TRANSACTION_AUTH	The TOE SHALL guaranty mutual authentication with the Vehicle (Standard Transaction) and from the device's perspective, this guarantees that no private assets can be leaked by a MITM attack. This principle also allows the device to transmit data to the vehicle without any possibility of leakage by a passive or active eavesdropper.
O.KEY_EXCHANGE_AUTH	The TOE SHALL be able to guarantee the authenticity of the key exchange operation.
O.NON-TRACEABILITY	The TOE SHALL be able to ensure the non-traceability of data and keys being shared through an NFC or a BLE channel.

4.2 SECURITY OBJECTIVES FOR THE TOE OPERATIONAL ENVIRONMENT

Table 4-2 introduces the security objectives to be achieved by the environment associated to the TOE. These are related to the assumptions and in some cases to threats and OSPs.

Table 4-2: Security Objectives for the TOE Operational Environment

Security Objectives	Description
OE.DEVICE_PERSISTENCE	The device SHALL perform self-tests to ensure the integrity of critical functionality, software/firmware and data is maintained in order to ensure the integrity of the Mobile Device is maintained conformant.
OE.APPLET_EE_HW_MALFUNCTION_PROTECTION	The DK Applet EE SHALL ensure its correct operation and is expected to indicate or prevent its operation outside the normal operating conditions where reliability and secure operation has not been proven or tested. This is to prevent malfunctions. Examples of environmental conditions are voltage, clock frequency, temperature, or external energy fields.
OE. CERTIFICATE_REVOCATION_SET	The ecosystem SHALL inform all the relevant parties of the ecosystem (i.e. those who could be presented this certificate at any point in time), upon request of revocation of a certificate part of a Digital Key certificate chain (device side or vehicle side). This includes the CA that issued the certificate.
OE.APPLET_ABUSE_PROTECTION	The DK EE, Device OEM Server SHALL prevent those functions (which may not be used after Delivery) from being abused in order to disclose, manipulate critical assets such as Owner DK Secret, or manipulate, bypass, deactivate, change or explore security features or security services of the DK EE, Applet or Device OEM Server.
OE. INSTANCE_CA_CONFIDENTIALITY	The Certificate Chain (especially the INSTANCE CA, INSTANCE CA ATTESTATION and DEVICE OEM CA) SHALL be protected such that an attacker is not able to create a correctly attestable INSTANCE CA outside of the certified DK Applet / DK Applet EE.
OE. SECURE_DEVELOPMENT_AND_PRODUCTION	This objective SHALL ensure that any attack by internal attackers (employees, visitors) in development, production and provisioning, to directly or indirectly compromise the certificate chain or the Digital Key secret itself are prevented. This SHALL enforce that only trusted personnel are appointed for the above-mentioned processes.
OE.DEVICE_OEM	A Device OEM SHALL verify the validity of the DK Applet certificate provided by the CCC (according to the CCC Certification Program) before deploying it.
OE.OEM_SERVERS	Administrators of the OEM server are trusted people. They have been well trained to use and administrate the server securely. The Device OEM Server and the Vehicle OEM Server SHALL be hosted in secure data centers. PKI signature keys and secure operation of the PKI instances must be managed securely.
OE.KTS_SERVER	The device SHALL ensure that unnecessary PII of the device's user is not sent to the Key Tracking Server, thus preventing to track the Vehicle owner. The KTS server SHALL preserve the confidentiality of the user data being received and transmitted.
OE.APPS_VALIDATION	There SHALL be a mechanism to verify/validate the applications before being loaded/installed.
OE.PRODUCTION_ENV	The production environment SHALL be equipped with trusted personnel and the development SHALL take place based on the security guidelines that has been put in place. Also, production and provisioning should take place based on the guidelines, to prevent direct or indirect compromise of the certificate chain or the Digital Key secret itself

Security Objectives	Description
OE.OS_DOWNGRADE	Adequate issuance measures SHALL be in place to prevent loading older versions of the device's software and firmware that has publicly known security flaws (downgrade).
OE.ANTI_DOWNGRADE	An attacker SHALL not be able to downgrade the protocol to an older version that has publicly known security flaws. DK material SHALL be cleared if downgrade is performed.
OE. DK_PROTOCOL_SECURITY	The device SHALL implement strong communication protocol to prevent anti-relay, anti-replay, Man in the middle. This includes implementation or robust integrity mechanisms, use of strong cryptography and random number generators.
OE.SECURE_KEY_RETRIEVE	The Device SHALL implement a secure key retrieval mechanism such that it prevents unauthorized key retrieval by attackers for gaining access to the communication channel.
OE.KEY_SHARE	The device SHALL protect the integrity & confidentiality of the keys being shared.
OE.KEY_OPTIONS	The Device OS SHALL be able to protect the integrity & confidentiality of the KEY options.
OE.CLOCK	The Device OS should provide a reliable source for the clock.
OE.CAR_LOCATION	The Device OS SHALL prevent any display of information related to the location of paired vehicles when locked.
OE.USER_AUTHENTICATION_DK_SHARING	The device SHALL prevent the sharing if the origin of the authorization (Owner Authentication) is not ensured. Additionally, an unambiguous signal of friend identity SHALL be established before initiating friend sharing. The device SHALL prevent the sharing if the recipient identity cannot be traced back to the friend.
OE. USER_PRIVACY_CONSENT	OEM native app or device framework SHALL make user aware and asks for consent for any private information that is shared by the device with the servers and vehicle.
OE.CERT_VALIDATION	The processing of a certificate provided by an external entity to the device SHALL be verified before being processed. The verification SHALL cover the certificate format and validity.
OE.RADIO_RELAY	There SHALL be a mechanism to detect the relay of a transaction using radio equipment.
OE. UNAUTHORIZED_KEY_SHARING	Without the consent/knowledge of the owner, the collaborating entities SHALL not share the keys/access credentials.
OE.DOS_DETECT	A DOS detection mechanism SHALL be implemented to ensure the availability by detecting whether the device is disabled by some means, if the internet connectivity is available, etc..
OE.REPLAY	A detection mechanism SHALL be implemented to ensure that the communication signals are not being observed and replayed.
OE.COMMUNICATION	A mechanism SHALL be implemented in order to ensure the integrity, confidentiality and authentication of the data transferred through the communication channel.
OE.VEHICLE_ROOT_KEY_CONFID	The vehicle ECU shall ensure the confidentiality of the vehicle root key (long term key).

4.3 SECURITY OBJECTIVES RATIONALE

4.3.1 *Threats*

Table 4-3 and Table 4-4 present the coverage of threats by the security objectives and the associated rationale. Table 4-3 maps the threats that are either covered by security objectives for the TOE or by a combination of security objectives for the TOE and for the TOE operational environment. Table 4-4 maps the threats that are covered exclusively by security objectives for the TOE operational environment.

Table 4-3: Threats and Security Objectives - Coverage

Threats from this PP	Security Objectives	Rationale
T.DK_PHYSICAL	O.OS_SUPPORT O.IC_SUPPORT O.RANDOMNESS	This threat is countered by physical protections which rely on the underlying platform and the secure element physical protection capabilities
T.UNAUTHORIZED_SE_MNG	O.SE_MANAGEMENT OE.COMMUNICATION	This threat is covered by • O.SE_MANAGEMENT which controls the access to card management functions such as the loading, installation, extradition or deletion of applets and • OE.COMMUNICATION which ensures the integrity, confidentiality and authentication of the data transferred through the communication channel.
T.LIFE_CYCLE	O.SE_MANAGEMENT	This threat is covered by O.SE_MANAGEMENT which controls the access to card management functions such as the loading, installation, extradition or deletion of applets and prevent attacks intended to modify or exploit the current life cycle of applications
T.IT_DISCLOSURE	O.IMMO_TOK_CONFID O.IC_SUPPORT	This threat is covered by O.IMMO_TOK_CONFID which ensures the confidentiality of the Immobilizer Token during storage (data at rest), deletion, processing. It is also supported by O.IC_SUPPORT which protects IMMOTOKEN from disclosure due to physical attacks.
T.DK_DISCLOSURE	O.DK_CONFID O.RANDOMNESS	This threat is covered by O.DK_CONFID which ensures the confidentiality of the secret elements of the Digital Key during generation, usage (strong cryptography operations), storage, deletion, such that those are never known outside the DK Applet within its DK Applet EE. It is also covered by O.RANDOMNESS which ensure covering a lack of randomness that could allow an attacker to predict communication.
T.FLAW_SW	O.SE_MANAGEMENT	This threat is covered by:

Threats from this PP	Security Objectives	Rationale
	OE.DEVICE_PERSISTENCE OE.APPLET_ABUSE_PROTECTION	• O.SE_MANAGEMENT which controls the access to card management functions such as the loading, installation, extradition or deletion of applets • OE.DEVICE_PERSISTENCE ensures that the device will perform self-tests to ensure the integrity of critical functionality, software/firmware and data is maintained • OE.APPLET_ABUSE_PROTECTION ensures that DK EE, Applet, Device OEM Backend prevents that functions which may not be used after Delivery can be abused in order to disclose, manipulate critical assets
T.RETRIEVE_SECRET-SHARED-KEY_DKA	O.SEC_SHARED_KEY_CONFID	This threat is covered by O.SEC_SHARED_KEY_CONFID which ensures that the DK Applet ensures the confidentiality of the Secret Shared key.
T.RETRIEVE_KCMAC-KEY_DKA	O.KCMAC_KEY_CONFID	This threat is covered by O.KCMAC_KEY_CONFID which ensures that the DK Applet ensures the confidentiality of the Kcmac key.
T.RETRIEVE_SESSION-KEYS_DKA	O.SESSION_KEYS_CONFID	This threat is covered by O.SESSION_KEYS_CONFID which ensures that the DK Applet ensures the confidentiality of the session keys.
T.UNAUTHORIZED_ACCESS_DK_ASSET	O.IMMO_TOK_CONFID	This threat is covered by • O.IMMO_TOK_CONFID which ensures the confidentiality of the Immobilizer Token during storage (data at rest), deletion, processing.
T.CA_KEY_LEAK	O.DK_CONFID O.LONG_TERM_KEY_CONFID OE.VEHICLE_ROOT_KEY_CONFID	This threat is covered by • O.DK_CONFID which ensures the confidentiality of the secret elements of the Digital Key during generation, usage (strong cryptography operations), storage, deletion, such that those are never known outside the DK Applet within its DK Applet EE. • O.LONG_TERM_KEY_CONFID which ensures the confidentiality of long term key. • OE.VEHICLE_ROOT_KEY_CONFID which ensures the confidentiality of the long term key residing on the vehicle – ECU.
T.DENIAL_OF_LEGITIMATE_DELETIONS	O.ATTESTATION_ON_DELETION	This threat is covered by O.ATTESTATION_ON_DELETION which

Threats from this PP	Security Objectives	Rationale
		ensures that the DK Applet creates a deletion attestation for the requested key, and that it is securely deleted before the attestation is transferred to the requesting party.
T.ILLEGITIMATE_RKE	O.DEVICE_KEY_PAIR_CONFID	This threat is covered by O.DEVICE_KEY_PAIR_CONFID which ensures that the device key pair is protected in confidentiality, and therefore not accessible to an attacker, whom in turn cannot use it to impersonate the device during the challenge-signature-based authentication needed to perform RKE.
T.DK_SK_MODIFICATIONS	O.DK_INTEG O.IMMO_TOK_INTEG	This threat is covered by • O.DK_INTEG which ensures that the DK Applet and its Execution Environment ensure the integrity of the secret elements of the Digital Key when it is generated, used, deleted and stored • O.IMMO_TOK_INTEG which ensures the integrity of the Immobilizer Token during storage (data at rest), deletion, processing.
T.RADIO_SNIFF	O.FAST_TRANSACTION_AUTH O.STD_TRANSACTION_AUTH	This threat is covered by O.FAST_TRANSACTION_AUTH and O.STD_TRANSACTION_AUTH which ensure a secure channel is used when communicating between device and vehicle.
T.RADIO_MITM	O.KCMAC_KEY_INTEG O.SEC_SHARED_KEY_INTEG O.LONG_TERM_KEY_INTEG O.SESSION_KEYS_INTEG O.FAST_TRANSACTION_AUTH O.STD_TRANSACTION_AUTH O.KEY_EXCHANGE_AUTH O.DEVICE_KEY_PAIR_INTEG	This threat is covered by • O.KCMAC_KEY_INTEG which ensures the integrity of Kcmac key. • O.SEC_SHARED_KEY_INTEG which ensures the integrity of secret shared keys • O.LONG_TERM_KEY_INTEG which ensures the integrity of long term key • O.SESSION_KEYS_INTEG which ensures the integrity of session keys • O.FAST_TRANSACTION_AUTH which ensures the authentication takes place from device side • O.STD_TRANSACTION_AUTH which ensures that mutual authentication takes place between device and vehicle. • O.KEY_EXCHANGE_AUTH ensures the authenticity of key share operation.

Threats from this PP	Security Objectives	Rationale
		• O.DEVICE_KEY_PAIR_INTEG which ensures the integrity of the device key pair.
T.DATA_BREACH	O.DK_CONFID O.NON-TRACEABILITY	This threat is covered by • O.DK_CONFID which ensures the confidentiality of data/keys being shared during a transaction through NFC or BLE (if present in the device) channel. • O.NON-TRACEABILITY which ensures that the users are non-traceable across different vehicles through the same app.

Table 4-4: Threats and Security Objectives for the TOE Operational Environment - Coverage

Threats from this PP	OE Security Objectives	Rationale
T.NON_REVOKED	OE.CERTIFICATE_REVOCATION_SET	This threat is covered by OE. CERTIFICATE_REVOCATION_SET which ensures the ecosystem informs all the relevant parties of the ecosystem upon request of revocation of a certificate part of a digital key certificate chain (device side or vehicle side)
T.KTS_DATA_LEAK	OE.KTS_SERVER	This threat is covered by OE.KTS_SERVER which ensures that unnecessary PII of the device's user are not sent to the Key Tracking Server, thus preventing to track the Vehicle owner.
T.UPDATE_KEY_OPTIONS	OE.KEY_OPTIONS	This threat is covered by OE.KEY_OPTIONS ensures that the DK Applet ensures the integrity of the key options
T.DEVICE_THEFT	OE.USER_AUTHENTICATION_DK_SHARING	This threat is covered by OE.USER_AUTHENTICATION_DK_SHARING which ensures that device provides robust User Authentication methods to identify the DK User & an unambiguous signal of friend identity be established before initiating friend sharing.
T.PROTOCOL_DOWNGRADE	OE.ANTI_DOWNGRADE	This threat is covered by OE.ANTI_DOWNGRADE which ensures that an attacker will not be able to downgrade the protocol to an older version that has publicly known security flaws.
T.SIGN_COMPROMISE_VERIFY_COMPROMISE	OE.CERT_VALIDATION	This threat is covered by OE.CERT_VALIDATION which ensures the protection of creation and validation of electronic signatures
T.RADIO_RELAY_TRANSACTION	OE.RADIO_RELAY	This threat is covered by OE.RADIO_RELAY which ensures protection against relay a transaction with radio equipment attack.

Threats from this PP	OE Security Objectives	Rationale
T.INTERNET_CONNECTIVITY_DOS	OE.DOS_DETECT	This threat is covered by OE.DOS_DETECT which ensures availability by detecting whether the device is disabled by some means, if the internet connectivity is available, etc.
T.TIME_CHANGE	OE.CLOCK	This threat is covered by OE.CLOCK which ensures that the software uses a reliable source for the clock (date/time).
T.REPLAY_TRANSACTION	OE.REPLAY	This threat is covered by OE.REPLAY which protects against attack such as replay an observed NFC and BLE (if present in the device) transaction to the vehicle
T.UNAUTHORIZED_KEY_SHARING	OE.UNAUTHORIZED_KEY_SHARING	This threat is covered by OE. UNAUTHORIZED_KEY_SHARING which ensures that two or more collaborating adversaries cannot share (copy) access credentials without the vehicle owner's consent or knowledge.
T.INSTANCE_CA_DISCLOSURE	OE.INSTANCE_CA_CONFIDENTIALITY	This threat is covered by OE. INSTANCE_CA_CONFIDENTIALITY which ensures that the Certificate Chain (especially the INSTANCE CA, INSTANCE CA ATTESTATION and DEVICE OEM CA) is protected such that an attacker is not able to create a correctly attestable INSTANCE CA outside of the certified DK Applet / DK Applet EE.

The OE's listed in the following table participates in covering the identified threats but cannot be solely implemented to cover these threats sufficiently.

Table 11 Security Objectives and Threats - Coverage

Security Objectives	Threats
O.SE_MANAGEMENT	T.UNAUTHORIZED_SE_MNG T.LIFE_CYCLE
O.IMMO_TOK_CONFID	T.IT_DISCLOSURE T.UNAUTHORIZED_ACCESS_DK_ASSET
O.IMMO_TOK_INTEG	T.DK_SK_MODIFICATIONS
O.DK_CONFID	T.DK_DISCLOSURE T.CA_KEY_LEAK T.DATA_BREACH
O.DK_INTEG	T.DK_SK_MODIFICATIONS
O.LONG_TERM_KEY_CONFID	T.CA_KEY_LEAK
O.LONG_TERM_KEY_INTEG	T.RADIO_MITM
O.DEVICE_KEY_PAIR_CONFID	T.ILLEGITIMATE_RKE
O.DEVICE_KEY_PAIR_INTEG	T.RADIO_MITM
O.SEC_SHARED_KEY_CONFID	T.RETRIEVE_SECRET-SHARED-KEY_DKA
O.SEC_SHARED_KEY_INTEG	T.RADIO_MITM

Security Objectives	Threats
O.KCMAC_KEY_CONFID	T.RETRIEVE_KCMAC-KEY_DKA
O.KCMAC_KEY_INTEG	T.RADIO_MITM
O.SESSION_KEYS_CONFID	T.RETRIEVE_SESSION-KEYS_DKA
O.SESSION_KEYS_INTEG	T.RADIO_MITM
O.ATTESTATION_ON_DELETION	T.DENIAL_OF_LEGITIMATE_DELETIONS
O.RANDOMNESS	T.DK_PHYSICAL
O.IC_SUPPORT	T.DK_PHYSICAL
O.OS_SUPPORT	T.DK_PHYSICAL
O.FAST_TRANSACTION_AUTH	T.RADIO_MITM
O.STD_TRANSACTION_AUTH	T.RADIO_MITM
O.KEY_EXCHANGE_AUTH	T.RADIO_MITM
O.NON-TRACEABILITY	T.DATA_BREACH
OE. DEVICE_PERSISTENCE	T.FLAW_SW
OE. APPLET_ABUSE_PROTECTION	T.FLAW_SW
OE.COMMUNICATION	T.UNAUTHORIZED_SE_MNG
OE.CERTIFICATE_REVOCATION_SET	T.NON_REVOKED
OE.KTS_SERVER	T.KTS_DATA_LEAK
OE.KEY_OPTIONS	T.UPDATE_KEY_OPTIONS
OE.USER_AUTHENTICATION_DK_SHARING	T.DEVICE_THEFT T.RADIO_SNIFF
OE.ANTI_DOWNGRADE	T.PROTOCOL_DOWNGRADE
OE.CERT_VALIDATION	T.SIGN_COMPROMISE_VERIFY_COMPROMISE
OE.RADIO_RELAY	T.RADIO_RELAY_TRANSACTION
OE.DOS_DETECT	T.INTERNET_CONNECTIVITY_DOS
OE.CLOCK	T.TIME_CHANGE
OE.REPLAY	T.REPLAY_TRANSACTION
OE. UNAUTHORIZED_KEY_SHARING	T.UNAUTHORIZED_KEY_SHARING
OE.INSTANCE_CA_CONFIDENTIALITY	T.INSTANCE_CA_DISCLOSURE
OE.VEHICLE_ROOT_KEY_CONFID	T.CA_KEY_LEAK

4.3.2 *Organizational Security Policies*

Table 4-5 presents the coverage of OSPs by the security objectives for the TOE operational environment and the associated rationale. Table 4-6 provides the mapping of the security objectives for the TOE environment to the OSPs.

Table 4-5: OSPs and Security Objectives - Coverage

Organizational Security Policies	Security Objectives	Rationale
OSP.APPS_VALIDATION	OE.APPS_VALIDATION	This OSP is enforced by the security objective for the operational environment of the TOE OE.APPS_VALIDATION
OSP.OEM_SERVERS	OE.OEM_SERVERS	This OSP is enforced by the security objective for the operational environment of the TOE OE.OEM_SERVERS
OSP.KTS_SERVER	OE.KTS_SERVER	This OSP is enforced by the security objective for the operational environment of the TOE OE.KTS_SERVER
OSP.OS_DOWNGRADE	OE.OS_DOWNGRADE	This OSP is enforced by the security objective for the operational environment of the TOE OE.OS_DOWNGRADE
OSP.SECURE_KEY_RETRIEVE	OE.SECURE_KEY_RETRIEVE	This OSP is enforced by the security objective for the operational environment of the TOE OE.SECURE_KEY_RETRIEVE
OSP.KEY_SHARE	OE.KEY_SHARE	This OSP is enforced by the security objective for the operational environment of the TOE OE.KEY_SHARE
OSP.KEY_OPTIONS	OE.KEY_OPTIONS	This OSP is enforced by the security objective for the operational environment of the TOE OE.KEY_OPTIONS
OSP.SERVER_COMMUNICATION	OE.OEM_SERVERS OE.KTS_SERVER	This OSP is enforced by the security objective for the operational environment of the TOE OE.OEM_SERVERS and OE.KTS_SERVER
OSP.PROTOCOL_FAILURE	OE. DK_PROTOCOL_SECURITY	This OSP is enforced by the security objective for the operational environment of the TOE OE. DK_PROTOCOL_SECURITY
OSP.DOS_DETECT	OE.DOS_DETECT	This OSP is enforced by the security objective for the operational environment of the TOE OE.DOS_DETECT
OSP.CERTIFICATE	OE.CERT_VALIDATION	This OSP is enforced by the security objective for the operational environment of the TOE OE.CERT_VALIDATION
OSP.PKI_POLICY	OE.OEM_SERVERS	This OSP is enforced by the security objective for the operational environment of the TOE OE.OEM_SERVERS

Table 4-6: Security Objectives and OSPs - Coverage

Security Objectives	OSP
OE. DEVICE_PERSISTENCE	
OE.APPLET_EE_HW_MALFUNCTION_PROTECTION	
OE. CERTIFICATE_REVOCATION_SET	
OE. APPLET_ABUSE_PROTECTION	
OE. INSTANCE_CA_CONFIDENTIALITY	
OE. SECURE_DEVELOPMENT_AND_PRODUCTION	

Security Objectives	OSP
OE.DEVICE_OEM	
OE.OEM_SERVERS	OSP.OEM_SERVERS OSP.SERVER_COMMUNICATION OSP.PKI_POLICY
OE.KTS_SERVER	OSP.KTS_SERVER OSP.SERVER_COMMUNICATION
OE.APPS_VALIDATION	OSP.APPS_VALIDATION
OE.PRODUCTION_ENV	
OE.OS_DOWNGRADE	OSP.OS_DOWNGRADE
OE.ANTI_DOWNGRADE	
OE. DK_PROTOCOL_SECURITY	OSP.PROTOCOL_FAILURE
OE.SECURE_KEY_RETRIEVE	OSP.SECURE_KEY_RETRIEVE
OE.KEY_SHARE	OSP.KEY_SHARE
OE.KEY_OPTIONS	OSP.KEY_OPTIONS
OE.CLOCK	
OE.CAR_LOCATION	
OE.USER_AUTHENTICATION_DK_SHARING	
OE.USER_PRIVACY_CONSENT	
OE.CERT_VALIDATION	OSP.CERTIFICATE
OE.RADIO_RELAY	
OE. UNAUTHORIZED_KEY_SHARING	
OE.DOS_DETECT	OSP.DOS_DETECT
OE.REPLAY	
OE.COMMUNICATION	
OE.VEHICLE_ROOT_KEY_CONFID	

4.3.3 *Assumptions*

Table 4-7 presents the coverage of the assumptions by the security objectives for the TOE operational environment and the associated rationale. Table 4-8 provides the mapping of the security objectives for the TOE environment to the assumptions.

Table 4-7: Assumptions and Security Objectives for the TOE Operational Environment - Coverage

Assumptions	Security Objectives for the Operational Environment	Rationale
A.USER_AUTHENTICATION	OE.USER_AUTHENTICATION_DK_SHARING	This assumption is directly upheld by OE.USER_AUTHENTICATION_DK_SHARING
A.USER_PRIVACY_CONSENT	OE.USER_PRIVACY_CONSENT	This assumption is directly upheld by OE.USER_PRIVACY_CONSENT

Assumptions	Security Objectives for the Operational Environment	Rationale
A.CERTIFICATION_REVOCATION_SET	OE.CERTIFICATE_REVOCATION_SET	This assumption is directly upheld by OE.CERTIFICATE_REVOCATION_SET
A.OEM_ADMIN	OE.OEM_SERVERS	This assumption is directly upheld by OE.OEM_SERVERS
A.PRODUCTION_ENV	OE.PRODUCTION_ENV	This assumption is directly upheld by OE.PRODUCTION_ENV
A.DEVICE_OEM	OE.DEVICE_OEM	This assumption is directly upheld by the OE.DEVICE_OEM
A.OS_CLOCK	OE.CLOCK	This assumption is directly upheld by OE.CLOCK
A.CAR_LOCATION	OE.CAR_LOCATION	This assumption is directly upheld by OE.CAR_LOCATION
A.DEVICE_OEM_INSIDER	OE. SECURE_DEVELOPMENT_AND_PRODUCTION	This assumption is directly upheld by OE. SECURE_DEVELOPMENT_AND_PRODUCTION
A.SHARING_MASQUERADE	OE.KEY_SHARE	This assumption is directly upheld by OE.KEY_SHARE
A.DEVICE_SAFETY	OE.USER_AUTHENTICATION_DK_SHARING	This assumption is directly upheld by OE.USER_AUTHENTICATION_DK_SHARING
A.REPLAY	OE.REPLAY	This assumption is directly upheld by OE.REPLAY
A.RADIO_RELAY	OE.RADIO_RELAY	This assumption is directly upheld by OE.RADIO_RELAY
A.OEM_SERVER_SECURITY	OE.OEM_SERVERS	This assumption is directly upheld by OE.OEM_SERVERS
A.VEHICLE_ROOT_KEY	OE.VEHICLE_ROOT_KEY_CONFID	This assumption is directly upheld by OE.VEHICLE_ROOT_KEY_CONFID

Table 4-8: Security Objectives for the Operational Environment and Assumptions - Coverage

Security Objectives	Assumptions
OE. DEVICE_PERSISTENCE	
OE.APPLET_EE_HW_MALFUNCTION_PROTECTION	
OE.CERTIFICATE_REVOCATION_SET	A.CERTIFICATE_REVOCATION_SET
OE.APPLET_ABUSE_PROTECTION	
OE.INSTANCE_CA_CONFIDENTIALITY	
OE.SECURE_DEVELOPMENT_AND_PRODUCTION	
OE.DEVICE_OEM	A.DEVICE_OEM
OE.OEM_SERVERS	A.OEM_ADMIN A.OEM_SERVER_SECURITY
OE.KTS_SERVER	
OE.APPS_VALIDATION	
OE.PRODUCTION_ENV	A.PRODUCTION_ENV

Security Objectives	Assumptions
OE.OS_DOWNGRADE	
OE.ANTI_DOWNGRADE	
OE.DK_PROTOCOL_SECURITY	
OE.SECURE_KEY_RETRIEVE	
OE.KEY_SHARE	A.SHARING_MASQUERADE
OE.KEY_OPTIONS	
OE.CLOCK	A.OS_CLOCK
OE.CAR_LOCATION	A.CAR_LOCATION
OE.USER_AUTHENTICATION_DK_SHARING	A.USER_AUTHENTICATION A.DEVICE_SAFETY
OE. USER_PRIVACY_CONSENT	A.USER_PRIVACY_CONSENT
OE.CERT_VALIDATION	
OE.RADIO_RELAY	A.RADIO_RELAY
OE. UNAUTHORIZED_KEY_SHARING	
OE.DOS_DETECT	
OE.REPLAY	A.REPLAY
OE.COMMUNICATION	
OE.VEHICLE_ROOT_KEY_CONFID	A.VEHICLE_ROOT_KEY

5 SECURITY REQUIREMENTS

This Protection Profile uses the following text formatting to make the Security Functional Requirements (SFRs) operations more visible to the reader and to simplify the work for the Security Target writer. It highlights how the specific instantiations in the SFRs are derived from the functional components in Part 2 of the CC.

- The **refinement** operation is used to add detail to a requirement and thus further restricts a requirement. Refinements of security requirements are denoted in such a way that added words are in bold text and removed words are crossed out. If a refinement is added as a separate paragraph to an SFR instead of modifying its wording, this paragraph starts with the word "Refinement:" in **bold text**.
- The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections having been made by the PP author are denoted as underlined text and in addition a footnote will show the original text from CC, Part 2. Selections to be filled in by the ST author appear in square brackets with an indication that a selection is to be made [selection:] and are *italicized*.
- The **assignment** operation is used to assign a specific value to an unspecified parameter such as the length of a password. Assignments having been made by the PP author are denoted as underlined text and in addition a footnote will show the original text from CC, Part 2. Assignments to be filled in by the ST author appear in square brackets with an indication that an assignment is to be made [assignment:] and are *italicized*. In some cases, the assignment made by the PP authors defines a selection to be performed by the ST author, indicated by [selection:] and text, which is underlined and *italicized* like this.
- The **iteration** operation is used when a component is repeated with varying operations. The fact, that an iteration operation was used is obvious from the fact, that a component is contained (at least) twice in the PP. In order to distinguish the individual instances of a component, the component title is amended by showing a slash "/" and an individual name after the component identifier.
    - **Note:** For the sake of a better readability this notion may also be applied to some single components (being not repeated) in order to indicate that these SFRs belong to the same functional cluster.

In this PP, not all SFRs operations are completed. These are delegated to the author of a PP-conformant ST. However, our goal is to provide sufficient information to ST authors so that in the end the operations completed in the ST reflect at least the amount of information provided by the security objectives of the PP. To achieve this, the following options for each operation in an SFR are used:

- If there are no restrictions for possible completions by the ST author, this PP leaves the operation completely open;
- When the reader notices a given operation is partly completed, it means that the ST author is left only with a restricted choice.
- The ST author can complete the operation already defined in the PP.

Finally, **Application Notes** are also used to give other types of information to authors of PP-conformant STs or PPs. For example, guidance on how an ST author can apply the SFR in the

specific context of the TOE or simply translates the SFR into natural language close to the relevant security objectives for the TOE.

5.1 SECURITY FUNCTIONAL REQUIREMENTS

This section describes the requirements imposed on the TOE in order to achieve the security objectives laid down in the previous chapter. All the requirements identified in this section are instances of those stated in [CC2].

The SFRs listed below state requirements specific to the DK Applet part of the composite TOE.

5.1.1 *Cryptographic Key Management*

Cryptographic keys SHALL be managed throughout their life cycle. The following SFRs are intended to support that lifecycle and consequently defines requirements for the following activities:

- cryptographic key generation,
- cryptographic key distribution and
- cryptographic key destruction.

FCS_CKM.1 Cryptographic key generation | EC Point Generation

Hierarchical to:	**No other components.**
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, FCS_CKM.5 Cryptographic key derivation or FCS_COP.1 Cryptographic operation] [FCS_RBG.1 Random bit generation, or FCS_RNG.1 Generation of random numbers] FCS_CKM.6 Timing and event of cryptographic key destruction
FCS_CKM.1.1/ECC	The TSF SHALL generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ECC with P-256 (SECP256r1)[4] and specified cryptographic key sizes 256-bit[5] that meet the following standards: [selection: *[BSI TR-03111], ANSI X9.62 [X9.62a], ANSI X9.63 [X9.63], [FIPS PUB 186-4]*][6] **Application Note 13:** concerned assets are D.APPLET_ROOT_KEY and D.DEVICE_KEY_PAIR.

Application Note 14	• The keys can be generated and diversified in accordance with Java Card specification in classes KeyPair (at least Session key generation) and RandomData. • This component SHALL be instantiated according to the version of the Java Card API applying to the security target and the implemented algorithms.

[4] **[assignment:** ***cryptographic key generation algorithm*****]**
[5] **[assignment:** ***cryptographic key sizes*****]**
[6] **[assignment*****: list of standards*****]**

FCS_CKM.5 Cryptographic key derivation

Hierarchical to:	**No other components.**
Dependencies:	FCS_COP.1 Cryptographic operation FCS_CKM.6 Timing and event of cryptographic key destruction
FCS_CKM.5.1/Session_keys	The TSF shall derive cryptographic symmetric [7] keys from D.SECRET_SHARED_KEY or D.LONG_TERM_KEY [8] in accordance with a specified key derivation algorithm HKDF-SHA-256[9] and specified cryptographic key sizes 128[10] that meet the following: IETF [RFC 5869][11]. **Application Note 15**: The derivation from D.SECRET_SHARED_KEY or D.LONG_TERM_KEY depends on the command: AUTH0 or AUTH1.
FCS_CKM.5.1/Long_Term_key	The TSF shall derive cryptographic symmetric [12] keys from D.SECRET_SHARED_KEY [13] in accordance with a specified key derivation algorithm HKDF-SHA-256[14] and specified cryptographic key sizes 256[15] that meet the following: IETF [RFC 5869][16]. **Application Note 16**: Initially filled with random bits.

FCS_CKM.5/Secret_Shared_key Cryptographic key derivation (refinement for key establishment)

Hierarchical to:	**No other components.**
Dependencies:	FCS_COP.1 Cryptographic operation FCS_CKM.6 Timing and event of cryptographic key destruction
FCS_CKM.5.1/Secret_Shared_key	The TSF shall derive cryptographic ephemeral keys [17] from agreed ephemeral vehicle public key and endpoint private key[18] in accordance with a specified key **establishment** algorithm and specified cryptographic key sizes Elliptic curve-based Diffie-Hellman Ephemeral key agreement and cryptographic key sizes 256-bit [19] that meet the following: [BSI TR-03111][20].

FCS_RNG.1 Random number generation

FCS_RNG.1.1	The TSF shall provide a [selection*: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*] random number generator **[selection: *DRG.2, DRG.3, DRG.4, PTG.2, PTG.3, NTG.1*]**[21] that implements: generation of strong cryptographic random numbers, key

[7] **[assignment: *key type*]**
[8] **[assignment: *input parameters*]**
[9] **[assignment: *key derivation algorithm*]**
[10] **[assignment: *list of key sizes*]**
[11] **[assignment*: list of standards*]**
[12] **[assignment: *key type*]**
[13] **[assignment: *input parameters*]**
[14] **[assignment: *key derivation algorithm*]**
[15] **[assignment: *list of key sizes*]**
[16] **[assignment*: list of standards*]**
[17] **[assignment: *key type*]**
[18] **[assignment: *input parameters*]**
[19] **[assignment: *list of key sizes*]**
[20] **[assignment: *list of standards*]**
[21] This is a refinement. Refer to [AIS20] or [AIS31]

	generation functions use adequate entropy source from approved random number generator(s)[22].
FCS_RNG.1.2	The TSF shall provide [selection: *bits, octets of bits, numbers* [assignment: *format of the numbers*]] that meet [assignment: *a defined quality metric*].

FCS_CKM.6 Timing and event of 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, or FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6.1	The TSF SHALL destroy [assignment: *list of cryptographic keys (including keying material)*] when [selection: *no longer needed*, *[assignment: other circumstances for key or keying material destruction]*].
FCS_CKM.6.2	The TSF SHALL destroy cryptographic keys and keying material specified in FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method [selection: *zeroes, ones, pseudo-random pattern, a new value of a key of the same size, a value that does not contain any security attribute]* that meets the following: None[23].

5.1.2 *Cryptographic Operation*

In order for a cryptographic operation to function correctly, the operation SHALL be performed in accordance with a specified algorithm and with a cryptographic key of a specified size. The following SFRs specify all this latter information to be enforced by the TSF.

It covers the following:

- data encryption and/or decryption,
- digital signature generation and/or verification,
- cryptographic checksum generation for integrity and/or verification of checksum,
- secure hash (message digest),
- cryptographic key encryption and/or decryption.

FCS_COP.1 Cryptographic operation

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, or FCS_CKM.5 Cryptographic key derivation] FCS_CKM.6 Timing and event of cryptographic key destruction
FCS_COP.1.1/Hash	The TSF SHALL perform Cryptographic Hashing[24] in accordance with

[22] **[assignment: *list of security capabilities*]**
[23] **[assignment: *list of standards*]**
[24] **[assignment: *list of cryptographic operations*]**

	a specified cryptographic algorithm SHA-256[25] and cryptographic key sizes None[26] that meet the following**:** [selection*: ISO/IEC 10118-3:2018, FIPS 180-4*]**[27].**
FCS_COP.1.1/HMAC	The TSF SHALL perform Keyed Hash Message Authentication[28] in accordance with a specified cryptographic algorithm HMAC-SHA-256[29], and cryptographic key sizes 256-bit[30] that meet the following: ISO/IEC 9797-2:2011[31].
FCS_COP.1.1/Encryption/decryption	The TSF SHALL perform data encryption or decryption[32] in accordance with a specified cryptographic algorithm AES with CBC mode of operation[33] and cryptographic key sizes 128-bit[34] that meet the following: FIPS PUB 197, NIST SP 800-38A[35].
FCS_COP.1.1/CMAC	The TSF SHALL perform Message Authentication Code[36] in accordance with a specified cryptographic algorithm AES CMAC[37] and cryptographic key sizes 128-bit[38] that meet the following: NIST SP 800-38B[39].
FCS_COP.1.1/ECDSA	The TSF SHALL perform signature generation and signature verification[40] accordance with a specified cryptographic algorithm ECDSA with NIST P-256 curve[41] and cryptographic key sizes 256-bit[42] that meet the following: ANSI X9.62 or FIPS186-4[43]. **Application Note 17:** D.DEVICE_KEY_PAIR is used with FIPS186–4 standard for the challenge-signature-based authentication for RKE while ANSI X9.62 is used for the trust chain's certificates signature verification.

[25] **[assignment: *cryptographic algorithm*]**
[26] **[assignment: *cryptographic key sizes*]**
[27] **[assignment: *list of standards*]**
[28] **[assignment: *list of cryptographic operations*]**
[29] **[assignment: *cryptographic algorithm*]**
[30] **[assignment: *cryptographic key sizes*]**
[31] **[assignment: *list of standards*]**
[32] **[assignment: *list of cryptographic operations*]**
[33] **[assignment: *cryptographic algorithm*]**
[34] **[assignment: *cryptographic key sizes*]**
[35] **[assignment: *list of standards*]**
[36] **[assignment: *list of cryptographic operations*]**
[37] **[assignment: *cryptographic algorithm*]**
[38] **[assignment: *cryptographic key sizes*]**
[39] **[assignment: *list of standards*]**
[40] **[assignment: *list of cryptographic operations*]**
[41] **[assignment: *cryptographic algorithm*]**
[42] **[assignment: *cryptographic key sizes*]**
[43] **[assignment: *list of standards*]**

5.1.3 *Access Control Policy | Security Domain*

The following SFRs define the Security Functional Policy for access control to the TOE, which is called SD_SFP. For better readability SD_FSP is defined in Table 5-1 and the SFRs refer to it.

Table 5-1: Access control SFP - SD_SFP

Type	Short name	Definition
Subjects[44]	S.INSTALLER, (from [PP0099])	The installer is the on-card entity which acts on behalf of the card issuer. This subject is involved in the loading of packages and installation of applets
	S.CAD (from [PP0099])	The CAD represents off-card entity that communicates with the S.INSTALLER
	S.SD	SD stands for Security Domain and here S.SD can be representing an off-card entity on the card such as a validation authority, application provider etc.
Objects	O.Load_File	DK Applet Load file or Executable File, in case of application loading, installation, extradition or registry update, with a set of intended privileges and its targeted associated SD AID.
	O.Delegation_Token	The Delegation Token, in case of Delegated Management operations, with the attributes Present or Not Present;
	O.DAP	The DAP Block, in case of application loading, with the attributes Present or Not Present;
Operations	O.GP_CCM	GlobalPlatform's card content management commands
	O.API	API methods
Rules	R_GPF	Runtime behavior rules defined by GlobalPlatform |GP] for: • loading (Section 9.3.5 of [GP]); • installation (Section 9.3.6 of [GP]); • extradition (Section 9.4.1 of [GP]); • registry update (Section 9.4.2 of [GP]); • content removal (Section 9.5 of [GP]).

FDP_ACC.2 Complete access control

Hierarchical to:	FDP_ACC.1 Subset access control
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.2.1	The TSF SHALL enforce the SD_SFP[45] on *all subjects, objects defined by the SD_FSP*[46] and all operations among subjects and objects covered by the SFP.

[44] Subjects are active components of the TOE that (essentially) act on the behalf of users. The users of the TOE include people or institutions (like the applet developer, the card issuer, the verification authority), hardware and software components (like the application packages installed on the card). Some of the users may just be aliases for other users. For instance, the verification authority in charge of the bytecode verification of the applications may be just an alias for the card issuer.

[45] **[assignment: *access control SFP*]**

[46] **[assignment: *list of subjects and objects*]**

FDP_ACC.2.*2*	The TSF SHALL ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP

5.1.4 *Access Control Functions | Security Domain*

FDP_ACF.1 Security attribute base access control

Hierarchical to:	**No other components.**
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1	The TSF SHALL enforce the SFP_AC[47] to objects based on the following: All subjects and objects together with their respective security attributes as defined in SD_SFP[48].
FDP_ACF.1.2	The TSF SHALL enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: Rules for all access methods and access rules defined in SD_SFP[49].
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: when at least one of the rules R_GPF defined in the SD_SFP does not hold[50]

Application Note 18	The dependency FMT_MSA.3 will not be fulfilled here, since there is no initialisation of attributes necessary.

5.1.5 *Information Flow Control Policy | Secure Channel Protocol*

Table 5-2 defines the elements of the information flow control policy SC_SFP for card content management.

Table 5-2: Information Flow Control SFP - SC_SFP

Type	Short name	Definition
Subjects	S.INSTALLER, (from [PP0099])	The installer is the on-card entity which acts on behalf of the card issuer. This subject is involved in the loading of packages and installation of applets
	S.CAD (from [PP0099])	The CAD represents off-card entity that communicates with the S.INSTALLER
	S.SD	SD stands for Security Domain and here S.SD can be representing an off-card entity on the card such as a validation authority, application provider etc.

[47] **[assignment: *access control SFP*]**
[48] **[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*]**
[49] **[assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]**
[50] **[assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]**

Type	Short name	Definition
Information	I.CCM	The information controlled by this policy is the card content management command, including personalization commands, in the APDUs sent to the card and their associated responses returned to the CAD.
Operations	O.GP_CCM	GlobalPlatform's card content management commands
	O.API	API methods
Rules	R_GPF	Runtime behavior rules defined by GlobalPlatform |GP] for: • loading (Section 9.3.5 of [GP]); • installation (Section 9.3.6 of [GP]); • extradition (Section 9.4.1 of [GP]); • registry update (Section 9.4.2 of [GP]); • content removal (Section 9.5 of [GP]).

FDP_IFC.2 Complete Information Flow Control

Hierarchical to:	FDP_IFC.1 Subset information flow control
Dependencies:	FDP_IFF.1 Simple security attributes
FDP_IFC.2.1/SCP	The TSF SHALL enforce the SCP_SFP[51] on subjects, information and operations[52] and all operations that cause that information to flow to and from subjects covered by the SCP_SFP.
FDP_IFC.2.2/SCP	The TSF SHALL ensure that all operations that cause any information in the TOE to flow to and from any subject in the TOE are covered by an information flow control SFP.

5.1.6 *Information Flow Control Functions | Secure Channel Protocol*

FDP_IFF.1 Simple security attributes

Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation
FDP_IFF.1.1/SCP	The TSF SHALL enforce the SCP_SFP[53] based on the following types of subject and information security attributes: Subjects and information as defined by the SCP_SFP, and for each, the security attributes as defined in [GP] and [assignment: *list of additional security attributes*][54].
FDP_IFF.1.2/SCP	The TSF SHALL permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: Rules R_GPF as defined by the SCP_SFP[55].
FDP_IFF.1.3/SCP	The TSF SHALL enforce the [assignment: *additional information flow control SFP rules*].

[51] **[assignment: *information flow control SFP*]**
[52] **[assignment: *list of subjects and information*]**
[53] **[assignment: *information flow control SFP*]**
[54] **[assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*].**
[55] **[assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*]**

FDP_IFF.1.4/SCP	The TSF SHALL explicitly authorise an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly authorise information flows]*.
FDP_IFF.1.5/SCP	The TSF SHALL explicitly deny an information flow based on the following rules: When none of the conditions listed in the element FDP_IFF.1.4 of this component hold and at least one of those listed in the element FDP_IFF.1.2 does not hold[56].

Application Note 19	For authors of complying STs: In order to allow a more detailed specification, further security attributes may be added as suitable.

Application Note 20	The on-card and the off-card subjects have security attributes such as MAC, Cryptogram, Challenge, Key Set, Static Keys, etc.

Application Note 21	An SFR FMT_MSA.3 is not used here, since the security attributes used in the SCP_SFP are already contained in the I.CCM when entering the TOE, therefore rules for creation of information and default values of security attributes are not applicable

5.1.7 *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[57] the following objects: Cryptographic buffers[58].

Application Note 22	Cryptographic Buffers can be Cryptographic data used in runtime cryptographic computations, like a seed used to generate a key.

5.1.8 *Stored data integrity*

FDP_SDI.2 Stored data integrity monitoring and action

Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring
Dependencies:	No dependencies.
FDP_SDI.2.1	The TSF SHALL monitor user data stored in containers controlled by the TSF for [assignment**:** *integrity errors***]** on all objects, based on the following attributes: [assignment: *user data attributes*].
FDP_SDI.2.2	Upon detection of a data integrity error, the TSF SHALL prohibit the use of the altered data, send notification of the error where applicable[59].

[56] **[assignment: *rules, based on security attributes, that explicitly deny information flows*].**
[57] **[selection: *allocation of the resource to, deallocation of the resource from*]**
[58] **[assignment: *list of objects*].**
[59] **[assignment: *action to be taken*].**

5.1.9 *Inter-TSF user data integrity transfer protection*

FDP_UIT.1 Data exchange Integrity | Card Content Management

Hierarchical to:	**No other components**
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]
FDP_UIT.1.1/CCM	The TSF SHALL enforce the Secure channel protocol Information flow control policy[60] to [selection: *transmit, receive*] user data in a manner protected from [selection: *modification, deletion, insertion, replay*] errors.
FDP_UIT.1.2/CCM	The TSF SHALL be able to determine on receipt of user data, whether modification, deletion, insertion, replay[61] has occurred.

5.1.10 *Identification and Authentication*

FIA_UAU.3 Unforgeable authentication

Hierarchical to:	**No other components**
Dependencies:	No dependencies.
FIA_UAU.3.1	The TSF SHALL [selection: *detect, prevent*] use of authentication data that has been forged by any user of the TSF.
FIA_UAU.3.2	The TSF SHALL [selection: *detect, prevent*] use of authentication data that has been copied from any other user of the TSF.

5.1.11 *Security Management | 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/deletion of keys	The TSF SHALL restrict the ability to delete[62] the keys[63] to Digital Key framework, [**ass**ignment: *other authorised identified roles*] [64].

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 the DK Applet's AID[65].

[60] **[assignment: *access control SFP(s) and/or information flow control SFP(s)*]**
[61] **[selection: *modification, deletion, insertion, replay*]**
[62] **[selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]**
[63] **[assignment: *list of TSF data*]**
[64] **[assignment: the authorized identified roles]**
[65] **[assignment: *list of TSF data*].**

Application Note 23	The value of the Applet's AID is defined in [CCC-DK-TS], Section 15.3.2.1.

5.1.12 *Specifications of Management Functions | TSF data*

FMT_SMF.1 Specification of Management Functions

Hierarchical to:	**No other components**
Dependencies:	No dependencies
FMT_SMF.1.1	The TSF SHALL be capable of performing the following management functions: creates a deletion attestation for the requested key (for deletion), and that it is securely deleted before the attestation is transferred to the requesting party[66].

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 Digital Key framework, Vehicle, [assignment: *other authorised identified roles*] [67].
FMT_SMR.1.2	The TSF SHALL be able to associate users with roles.

Application Note 24	The dependency to FIA_UID.1 is not applicable to this TOE. This PP does not require the identification of the roles to be assigned which is handled by the operational environment.

5.1.13 *Unlinkability*

FPR_UNL.1 Unlinkability

Hierarchical to:	**No other components**
Dependencies:	No dependencies
FPR_UNL.1.1	The TSF shall ensure that any entity (other than the TOE, the DK Framework or the Vehicle)[68] is unable to determine whether data and key exchanged over NFC or BLE (between the TOE and the Vehicle)[69] were caused by the same user[70].

5.1.14 *Protection of the TSF*

FPT_ITC.1 Inter-TSF confidentiality during transmission

Hierarchical to:	**No other components**
Dependencies:	No dependencies
FPT_ITC.1.1	The TSF SHALL protect all TSF data transmitted from the TSF to another trusted IT product from unauthorised disclosure during transmission.

[66] **[assignment: *list of management functions to be provided by the TSF*].**
[67] **[assignment: the authorized identified roles]**
[68] **[assignment: *set of entities and/or operations*]**
[69] **[assignment: *list of entities and/or operations*]**
[70] **[selection: *were caused by the same user, are related as follows [assignment: list of relations]*]**

FPT_ITI.1/Vehicle_Integrity Inter-TSF detection of modification

Hierarchical to:	**No other components**
Dependencies:	No dependencies
FPT_ITI.1.1/ Vehicle_Integrity	The TSF SHALL provide the capability to detect modification of all TSF data during transmission between the TSF and another trusted IT product within the following metric: metric as defined in FCS_COP.1.1/CMAC[71].
FPT_ITI.1.2/ Vehicle_Integrity	The TSF SHALL provide the capability to verify the integrity of all TSF data transmitted between the TSF and another trusted IT product and perform terminate the on-going process[72] if modifications are detected.

5.1.15 *Internal TOE TSF data transfer*

FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to:	**No other components**
Dependencies:	No dependencies
FPT_ITT.1.1/ IMMO_TOKEN	The TSF SHALL protect TSF data from disclosure[73] when it is transmitted between separate parts of the TOE.
FPT_ITT.1.1/ DIGITAL_KEY	The TSF SHALL protect TSF data from disclosure[74] when it is transmitted between separate parts of the TOE.

FPT_ITT.3 TSF data integrity monitoring

Hierarchical to:	**No other components**
Dependencies:	No dependencies
FPT_ITT.3.1/ DIGITAL_KEY	The TSF SHALL be able to detect [selection: *modification of data, substitution of data, re-ordering of data, deletion of data*, [assignment: *other integrity errors*]] for TSF data transmitted between separate parts of the TOE.
FPT_ITT.3.1/IMMO_TOKEN	The TSF SHALL be able to detect [selection: *modification of data, substitution of data, re-ordering of data, deletion of data*, [assignment: *other integrity errors*]] for TSF data transmitted between separate parts of the TOE.
FPT_ITT.3.2/DIGITAL KEY	Upon detection of a data integrity error, the TSF SHALL take the following actions: **[**assignment: *specify the action to be taken*].
FPT_ITT.3.2/IMMO_TOKEN	Upon detection of a data integrity error, the TSF SHALL take the following actions: **[**assignment: *specify the action to be taken*].

5.1.16 *Replay Detection*

The following SFR uses the Subject defined hereafter:

- S.Transaction data: This can be the data/keys being shared between the DK Applet and the vehicle (through NFC, or BLE if present in the device) or between the DK Framework and the DK Applet.

FPT_RPL.1 Replay detection

Hierarchical to:	**No other components**

[71] **[assignment: *a defined modification metric*]**
[72] **[assignment: *action to be taken*]**
[73] **[selection: *disclosure, modification*]**
[74] **[selection: *disclosure, modification*]**

Dependencies:	No dependencies
FPT_RPL.1.1	The TSF SHALL detect replay for the following entities: S.Transaction data[75].
FPT_RPL.1.2	The TSF SHALL perform terminate the transaction[76] when replay is detected.

5.1.17 *Trusted Recovery*

FPT_RCV.2 Automated recovery

Hierarchical to:	FPT_RCV.1 Manual recovery
Dependencies:	AGD_OPE.1 Operational user guidance
FPT_RCV.2.1	When automated recovery from power failure[77] is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.
FPT_RCV.2.2	For power loss[78], the TSF SHALL ensure the return of the TOE to a secure state using automated procedures.

5.1.18 *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.
FTP_ITC.1.2	The TSF SHALL permit another trusted IT product[79] to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF SHALL initiate communication via the trusted channel for sharing of secret keys, user data, immobiliser token[80].

5.1.19 *Physical Resistance*

FPT_PHP.3 Resistance to physical attack

Hierarchical to:	**No other components**
Dependencies:	No dependencies
FPT_PHP.3.1	The TSF SHALL resist physical manipulation and physical probing[81] to the TSF[82] by responding automatically such that the SFRs are always enforced.

Application Note 25	This SFR is being included to the PP to highlight the possibility that security features implemented on the application level could be

[75] **[assignment: *list of identified entities*]**
[76] **[assignment: *list of specific actions*]**
[77] **[assignment: *list of failures/service discontinuities*]**
[78] **[assignment: *list of failures/service discontinuities*]**
[79] **[selection: *the TSF, another trusted IT product*]**
[80] **[assignment: *list of functions for which a trusted channel is required*]**
[81] **[assignment: *physical tampering scenarios*]**
[82] **[assignment: *list of TSF devices/elements*]**

required to support the detection of physical tampering provided by the FPT_PHP.3.1 of the IC [PP0084]. In that case the Security Target writer SHALL refine this SFR and map it to the related security functionality in the TSS.

5.1.20 *TSF Self-Tests*

Application Note 26	Startup of the TOE (TSF-testing) can be covered by FPT_TST.1. This SFR component is not mandatory in [PP0099] but appears in most of security requirements documents for masked applications. Testing could also occur randomly. Self-tests may become mandatory in order to comply with FIPS certification [FIPS140-3]

5.2 SECURITY ASSURANCE REQUIREMENTS

The security assurance requirement level is EAL4 augmented with ALC_DVS.2, ALC_FLR.2, AVA_VAN.5, and with the composite product package COMP. This covers all the security assurance components that are highlighted in Table 5-3 and listed in Table 5-4.

Table 5-3: EAL 4 augmented with ALC_DVS.2, ALC_FLR.2 and AVA_VAN.5

Assurance Class	Assurance Family	Assurance Components by Evaluation Assurance Level							
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7	
Development	ADV_ARC		1	1	1	1	1	1	
	ADV_FSP	1	2	3	4	5	5	6	
	ADV_IMP				1	1	2	2	
	ADV_INT					2	3	3	
	ADV_SPM						1	1	
	ADV_TDS		1	2	3	4	5	6	
Guidance Documents	AGD_OPE	1	1	1	1	1	1	1	
	AGD_PRE	1	1	1	1	1	1	1	
Life-cycle support	ALC_CMC	1	2	3	4	4	5	5	
	ALC_CMS	1	2	3	4	5	5	5	
	ALC_DEL		1	1	1	1	1	1	
	ALC_DVS			1	1	1	2	2	
	ALC_FLR								2
	ALC_LCD			1	1	1	1	2	
	ALC_TAT				1	2	3	3	
Security Target evaluation	ASE_CCL	1	1	1	1	1	1	1	
	ASE_ECD	1	1	1	1	1	1	1	
	ASE_INT	1	1	1	1	1	1	1	
	ASE_OBJ	1	2	2	2	2	2	2	
	ASE_REQ	1	2	2	2	2	2	2	
	ASE_SPD		1	1	1	1	1	1	
	ASE_TSS	1	1	1	1	1	1	1	
Test	ATE_COV		1	2	2	2	3	3	
	ATE_DPT			1	1	3	3	4	
	ATE_FUN		1	1	1	1	2	2	
	ATE_INT	1	2	2	2	2	2	3	
Vulnerability assessment	AVA_VAN	1	2	2	3	4	5	5	

Table 5-4: Composite product package COMP

Assurance class	Assurance components
ASE: Security Target evaluation	ASE_COMP.1 Consistency of Security Target (ST)
ADV: Development	ADV_COMP.1 Design compliance with the base component- related user guidance, ETR for composite evaluation and report of the base component evaluation authority
ALC: Life-cycle support	ALC_COMP.1 Integration of the dependent component into the related base component and consistency check for delivery and acceptance procedures
ATE: Tests	ATE_COMP.1 Composite product functional testing
AVA: Vulnerability assessment	AVA_COMP.1 Composite product vulnerability assessment

5.3 SECURITY REQUIREMENTS RATIONALE

5.3.1 *Rationale for the Security Functional Requirements*

Table 5-5 presents the coverage of the TOE security objectives by the SFRs defined in this PP and the associated rationale.

Table 5-5: Security Objectives and SFRs - Coverage

Security Objectives	SFR	Rationale
O.SE_MANAGEMENT	FDP_UIT.1/CCM FDP_ACC.2 FDP_ACF.1 FMT_SMF.1 FMT_SMR.1 FDP_IFC.2 FDP_IFF.1 FCS_CKM.1 FCS_CKM.5.1/Session_keys FCS_CKM.5.1/Long_term_keys FCS_CKM.5/Secret_Shared_key FCS_CKM.6 FDP_RIP.1 FMT_MTD.3	The security objective O.SE_MANAGEMENT is met by the following SFRs: • FDP_UIT.1/CCM which enforces the Secure Channel Protocol information flow control policy and the Security Domain access control policy to ensure the integrity of card management operations. • All SFRs related to Security Domains (FDP_ACC.2, FDP_ACF.1, FMT_SMF.1, FCS_CKM.1, FCS_CKM.5.1/Session_keys, FCS_CKM.5.1/Long_term_keys, • FCS_CKM.5/Secret_Shared_key, FCS_CKM.6, FDP_RIP.1, FMT_MTD.3, FMT_SMR.1 (as an SFR-supporting)) cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management. • All SFRs related to the secure channel (FDP_IFC.2, FDP_IFF.1) support this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.

Security Objectives	SFR	Rationale
O.IMMO_TOK_CONFID	FPT_ITT.1 FTP_ITC.1 FPT_PHP.3	The security objective O.IMMO_TOK_CONFID is met by the following SFRs: • FPT_ITT.1 which ensures that the data is protected when transmitted between separate parts of the TOE against disclosure, thus ensuring the confidentiality of immobilizer token. • FTP_ITC.1 which requires that the TSF provide a trusted communication channel between itself and another trusted IT product, which will further ensure the confidentiality of the immobilizer token being transmitted. • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical probing.
O.IMMO_TOK_INTEG	FPT_ITT.3	The security objective O.IMMO_TOK_INTEG is met by the following SFRs: • FPT_ITT.3 which enforces that the immobilizer token transmitted between separate parts of the TOE is monitored for identified integrity errors. • FPT_ITT.3 which enforces the actions to be taken in the event of an integrity violation detection.
O.DK_CONFID	FPT_ITT.1 FPT_PHP.3	The security objective O.DK_CONFID is met by the following SFRs: • FPT_ITT.1 which ensures that the secret elements of the Digital Key are protected when transmitted between separate parts of the TOE against disclosure. • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical probing.
O.DK_INTEG	FPT_ITT.3 FDP_SDI.2	The security objective O.DK_INTEG is met by the following SFRs: • FPT_ITT.3 which enforces that the assets of the Digital Key transmitted between separate parts of the TOE are monitored for identified integrity errors. • FDP_SDI.2 ensures that the user data imported into the TOE are monitored for integrity violations
O.LONG_TERM_KEY_CONFID	FPT_PHP.3 FPT_ITT.1	The security objective O.LONG_TERM_KEY_CONFID is met by the following SFRs: • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing.

Security Objectives	SFR	Rationale
		• FPT_ITT.1 which ensures that the Long Term key is protected when transmitted between separate parts of the TOE against disclosure.
O.LONG_TERM_KEY_INTEG	FDP_SDI.2 FPT_PHP.3	The security objective O.LONG_TERM_KEY_INTEG is met by the following SFRs: • FDP_SDI.2 which monitors stored user data for integrity errors. • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing.
O.DEVICE_KEY_PAIR_CONFID	FPT_PHP.3 FPT_ITT.1	The security objective O.DEVICE_KEY_PAIR_CONFID is met by the following SFRs: • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing. • FPT_ITT.1 which ensures that the Device key pair is protected when transmitted between separate parts of the TOE against disclosure.
O.DEVICE_KEY_PAIR_INTEG	FDP_SDI.2 FPT_PHP.3	The security objective O.DEVICE_KEY_PAIR_INTEG is met by the following SFRs: • FDP_SDI.2 which monitors stored user data for integrity errors. • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing.
O.SEC_SHARED_KEY_CONFID	FPT_PHP.3 FPT_ITT.1	The security objective O.SEC_SHARED_KEY_CONFID is met by the following SFRs: • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing. • FPT_ITT.1 which ensures that the Secret shared key is protected when transmitted between separate parts of the TOE against disclosure.
O.SEC_SHARED_KEY_INTEG	FDP_SDI.2 FPT_PHP.3	The security objective O.SEC_SHARED_KEY_INTEG is met by the following SFRs: • FDP_SDI.2 which monitors stored user data for integrity errors. • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing.
O.KCMAC_KEY_CONFID	FPT_PHP.3 FPT_ITT.1	The security objective O.KCMAC_KEY_CONFID is met by the following SFRs: • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing.

Security Objectives	SFR	Rationale
		• FPT_ITT.1 which ensures that the Kcmac key is protected when transmitted between separate parts of the TOE against disclosure.
O.KCMAC_KEY_INTEG	FDP_SDI.2 FPT_PHP.3	The security objective O.KCMAC_KEY_INTEG is met by the following SFRs: • FDP_SDI.2 which monitors stored user data for integrity errors. • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing.
O.SESSION_KEYS_CONFID	FPT_PHP.3 FPT_ITT.1	The security objective O.SESSION_KEYS_CONFID is met by the following SFRs: • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing. • FPT_ITT.1 which ensures that the session keys are protected when transmitted between separate parts of the TOE against disclosure.
O.SESSION_KEYS_INTEG	FDP_SDI.2 FPT_PHP.3	The security objective O.SESSION_KEYS_INTEG is met by the following SFRs: • FDP_SDI.2 which monitors stored user data for integrity errors. • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing.
O.ATTESTATION_ON_DELETION	FMT_SMF.1 FMT_MTD.1	The security objective O.ATTESTATION_ON_DELETION is met by the following SFRs: • FMT_SMF.1 which defines the management functions concerning the attestation creation and secure transferring of the same during a deletion operation. • FMT_MTD.1 which defines the management functions to be enforced and defines the concerned roles involved during a deletion operation.
O.RANDOMNESS	FCS_RNG.1	The security objective O.RANDOMNESS is met by FCS_RNG.1 which enforces the algorithms to be used for Random number generation and the entropy to be used based on certain standards.
O.IC_SUPPORT	FPT_PHP.3	The security objective O.IC_SUPPORT is met by the following SFR: • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing.
O.RECOVERY	FPT_RCV.2	The security objective O.RECOVERY is met by the following SFR: • FPT_RCV.2.1 which enforces the TOE to enter a maintenance mode where the ability to

Security Objectives	SFR	Rationale
		return to a secure state is provided, when automated recovery from certain failures is not possible. • FPT_RCV.2.2 which enforces the return of the TOE to a secure state using automated procedures during certain failures which can occur as defined.
O.OS_SUPPORT	FPT_PHP.3	The security Objective O.OS_SUPPORT is met by the following SFR: • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing.
O.FAST_TRANSACTION_AUTH	FCS_COP.1/CMAC FPT_ITC.1 FPT_ITI.1/Vehicle_Integrity FPT_RPL.1 FTP_ITC.1	The security objective O.FAST_TRANSACTION_AUTH is met by the following SFRs: • FCS_COP.1/CMAC provides the MAC used to detect modifications. • FPT_ITC.1 requires protection of TSF data from unauthorised disclosure during transmission • FPT_ITI.1/Vehicle_Integrity requires that modifications to TSF data are detected when transmitted between the DK Applet and vehicle • FTP_ITC.1 which requires that the TSF provide a trusted communication channel between itself and the vehicle guaranteeing a secure Device authentication to the Vehicle. • FPT_RPL.1 which protects against replay attacks over for such transactions
O.STD_TRANSACTION_AUTH	FCS_COP.1/CMAC FPT_ITC.1 FPT_ITI.1/Vehicle_Integrity FTP_ITC.1 FPT_RPL.1	The security objective O.STD_TRANSACTION_AUTH is met by the following SFRs: • FCS_COP.1/CMAC provides the MAC used to detect modifications. • FPT_ITC.1 requires protection of TSF data from unauthorised disclosure during transmission • FPT_ITI.1/Vehicle_Integrity requires that modifications to TSF data are detected when transmitted between the DK Applet and the vehicle • FTP_ITC.1 which requires that the TSF provide a trusted communication channel between itself and the vehicle allowing the device to transmit data to the vehicle without any possibility of leakage by a passive or active eavesdropper and protecting the private assets from an MITM attack.

Security Objectives	SFR	Rationale
		• FPT_RPL.1 which protects against replay attacks over for such transactions
O.KEY_EXCHANGE_AUTH	FIA_UAU.3 FCS_COP.1	The security objective O.KEY_EXCHANGE_AUTH is met by the following SFRs: • FIA_UAU.3 which prevents and detects forged data which could be used for key exchange operation, guaranteeing the authenticity of the key exchange operation. • FCS_COP.1 which ensures that the key exchange takes place accordance with a specified cryptographic algorithm & are based on defined standards.
O.NON-TRACEABILITY	FPR_UNL.1	The security objective O.NON-TRACEABILITY is met by the following SFR: • FPR_UNL.1 enforces that any entity (other than the TOE, the DK Framework or the Vehicle) is unable to determine whether data and key exchanged between the TOE and the Vehicle over NFC or BLE (if present in the device) were caused by the same user.

5.3.2 *Rationale for the Exclusion of Dependencies*

The dependency to FIA_UID.1 is not applicable to this TOE. This PP does not require the identification of the roles to be assigned which is handled by the operational environment.

5.3.3 *Rationale for the Security Assurance Requirements*

EAL4 allows a developer to attain a reasonably high assurance level without the need for highly specialized processes and practices. It corresponds to a white-box analysis and it can be considered as a reasonable level that can be applied to an existing product line without undue expense and complexity.

The TOE is intended to operate in open environments, where attackers can easily exploit vulnerabilities. According to the claimed intended usage of the TOE, it is very likely that it may represent a significant value and then constitute an attractive target for attackers. In some malicious usages of the TOE, the TOE's statistical or probabilistic mechanisms, for instance, may be subjected to analysis and attack in the normal course of operation.

For the matter of fact, we present below some concerns from the Vehicle industry clarifying the potential of attackers.

5.3.3.1 General relevance of Motor Vehicle Crime

Interpol issued a report in 2014 that provides an overview titled "Motor vehicle crime in global perspective" which present the types of crimes, the estimated annual damage (multi-billion Dollar per Year) and the actors involved.

Beyond the commercial damage, Interpol also raises the relevance of Motor vehicle crime for global crime incl. terrorism: "In fighting transnational organized crime, however, stolen motor vehicles should, in many cases, literally be seen as the "vehicle" of the crime. Stolen vehicles are

found to be the way of transport for bank robbers; illegal drugs are paid for with stolen vehicles; victims of trafficking in human beings are being discovered in stolen vehicles and car bombs are traditionally hidden in a stolen vehicle."

5.3.3.2 Attacker profile

Experience from international law enforcement organizations and vehicle OEMs prove that attacks on the information security assets are also conducted by criminal organizations that have established all necessary means to market stolen cars or parts internationally. These criminal organizations may have extensive financial resources and organisational skills.

5.3.3.3 Attacker motivation

Above mentioned criminal organizations which have established ways to market stolen cars or parts internationally can expect immense income from their criminal activities. Quote from Interpol: Beyond that Factors that influence the cost/benefit calculation of the attacker (quote Interpol):

- A relatively small investment requirement for the necessary tools to commit the crime (this is what we should change!);
- In comparison to other crimes, there is a generally mild punishment if convicted;
- The ample supply and opportunity in origin areas in combination with plenty of prospective customers in destination areas.

5.3.3.4 Capabilities of the assumed attacker

Based on the before mentioned attacker profile, we must assume the organized crime as an attacker. Organized crime is generally considered **capable to conduct attacks of level "high"**. The following list of potential capabilities (which reflects a similar methodology which is used in evaluation) may explain this**:**

- The attacker may have the financial resources and organizational skills to employ teams of experts and to conduct "Multiple Expert" attacks.
- The attacker may have the financial resources and organizational skills to get access to highest level equipment and to conduct "Multiple bespoke tool" attacks.
- The attacker may have the financial resources and organizational skills to get access to "critical" knowledge of the components (TOE). This may be conducted by bribing or putting employees of DK stakeholders or their suppliers under pressure so that they provide confidential specifications, cryptographic secrets etc.
- The attacker may have the financial resources and organizational skills to get access to a large number of TOE (e.g. devices or even the relevant parts of vehicles) and to perform attacks on this large number of TOEs in parallel.
- The attacker may have the financial resources and organizational skills to get access to "Open Samples" of the components (TOE). This may be conducted by bribing or putting employees of DK stakeholders or their suppliers under pressure so that they provide these open components (e.g. engineering samples of SE).
- The attacker may have the financial resources and organizational skills to develop and insert malware into components. This may be conducted by bribing or putting employees of DK~~S~~ stakeholders or their suppliers under pressure.

- The attacker may have the organizational and technical skills to conduct concatenated attack scenarios e.g. by obtaining confidential information first, weakening certain functions of the system (e.g. key generation or implementing malware) and finally to perform attacks on the weakened device.

5.3.3.5 Likelihood of attacks

Based on the motivation of potential attackers described before it SHALL be assumed that attacks will be conducted immediately after market introduction of components and that attacks will be performed permanently. It is probable that attackers will focus on stakeholders (i.e. device or vehicle OEMs) who enter the system. The overall likelihood of attacks SHALL be assumed as "high".

Based on all the assumptions presented above, EAL4 augmented with ALC_DVS.2, ALC_FLR.2 and AVA_VAN.5 seems to be the minimum reasonable assurance level for such type of sensitive TOE.

5.3.3.5.1 ALC_DVS.2 Sufficiency of security measures

The ALC_DVS.2 assurance component is added to provide a higher assurance on the security of the TOE development and manufacturing processes, especially for the secure handling of the embedded software and data. Those requirements appear as the most adequate for a manufacturing process in which several actors (Java Card Platform Developer, Application Developers, IC Manufacturer, etc.) exchange and store highly sensitive information (confidential code, cryptographic keys, personalisation data, etc.).

5.3.3.5.2 ALC_FLR.2 Flaw reporting procedures

The ALC_FLR.2 assurance component ensures that the developer has defined policies and procedures for receiving, tracking, correct and distribute fixes to security flaws of the TOE. This contributes to the continuous operation of the TOE and supports the assurance continuity.

5.3.3.5.3 AVA VAN.5 Advanced methodical vulnerability analysis

The AVA_VAN.5 assurance component is added to EAL 4 package to provide sufficient robustness to counter an attacker with high attack potential without the support of a protecting environment. Moreover, the TOE including the DK Applet are handling valuable assets such as an expensive vehicle. Potential attackers for such kind of products include international organizations, or even a state, disposing of important means and resources. Finally, the evaluator will base their evaluation methodology addressing vulnerability assessment on JIL Application of Attack Potential to Smart Cards and Similar Devices [JIL-attacks].

6 FUNCTIONAL PACKAGE FOR UWB SECURE RANGING

6.1 Identification

Name:	**Functional** Package for UWB Secure Ranging
Short name:	FP UWB-SR
Editor:	Internet of Trust
Sponsor:	Car Connectivity Consortium (CCC), LLC
CC Version:	CC:2022 Revision 1
Version:	2.0
Date:	March 24, 2025

6.2 Overview

The *Functional Package for UWB Secure Ranging* (FP UWB-SR) defines the security requirements for the DK Applet on SE regarding the Digital Key secure ranging functionality that is provided through the UWB module of the device.

The UWB-based secure ranging is used to locate the device in relation to the vehicle to allow e.g. passive entry and start of the vehicle. Indeed, the secure ranging functionality consists in determining the position of the user's device relatively to the vehicle. This is done by exchanging messages marked with time stamps thanks to the UWB module. The time difference between the message's stamp and the date of reception of the message allows the vehicle to measure the distance with the user's device. Since there are several antennas in the vehicle's UWB reader, it is possible to determine the actual position of the user's device relatively to the vehicle thanks to a triangulation mechanism.

In order to secure the ranging, a secure session for UWB communication between the vehicle and the device is established via Bluetooth commands. The processing of these commands triggers the derivation and generation of the UWB communication data, and the UWB Ranging Secret Key (URSK) if no pre-defined key is available. This data and the URSK are generated under the DK Applet in the SE. URSK subkeys derivation and associated cryptographic computations can be performed inside or outside the TOE. The actual secure ranging can then be performed using UWB communication via the UWB module.

This package is defined for use in STs conformant to the DK Applet PP whenever the TOE implements UWB-based secure ranging support. In that sense, the security functional requirements defined in the package are optional, of conditional type. Moreover, even in the case where the functional package applies, the cryptographic SFRs defined in section 6.5.8 remain conditional and applicable only if the TOE implements this specific functionality. This naturally extends to the specific security problem and objectives introduced in the package.

Notational convention: All the elements that are defined in this package carry the prefix 'UWB'.

6.3 Security Problem Definition

6.3.1 *Assets*

Table 6-1 lists the assets of the TOE that are related to the UWB-based secure ranging functionality, including cryptographic material and data used to establish the secure channel between the TOE and the UWB module.

Table 6-1: FP UWB-SR Assets, Description and Sensitivity

Assets	Description	Sensitivity (C, I, A)[83]
D.UWB_URSK	The UWB Ranging Secret Key, or URSK, which is used as the primary key from which the different cryptographic keys used for secure ranging session are derived. These derived keys, namely mURSK, mUPSK1, mUPSK2, dURSK, dUDSK and the salt shall also be included in this asset when the developer chose an architecture that allows their derivation within the TOE (as opposed to within the UWB module). **Application Note 27**: This asset is similar to D.SESSION_KEYS defined in the DK Applet PP but is declared separately due to the lifetime differences: the URSK may be present in the TOE during different sessions and is revoked/destroyed not later than 12 hours from activation. **Application Note 28**: if the URSK subkeys are included in the TOE, the following conditional SFRs automatically become part of the ST: - FCS_CKM.5/UWB_URSK_subkeys - FCS_COP.1/UWB	C, I
D.UWB_DATA	Any UWB communication transient data used during the secure ranging, such as the STS indexes and the timestamps. If these data were to be tampered with, the functionality would not be available. **Application Note 29**: If the UWB module handles all the data, which depends on implementation choices made by the vendor, this asset does not exist in the TOE.	I
D.UWB_BINDING_DATA	The permanent shared data used to bind the TOE and the UWB module in the device.	C, I

6.3.2 *Threats*

The threats that apply to the UWB related assets are described in Table 6-2. The threats in bold are defined in this package. The greyed threats are copied from the core DK Applet PP.

Table 6-2: FP UWB-SR Threats and Related Assets

Threats	Description	Covered Assets
T.UWB_RETRIEVE_URSK	Attackers retrieve the URSK. While this is not sufficient to impersonate the device, open and start the vehicle, it does defeat the secure ranging functionality.	D.UWB_URSK

[83] C = Confidentiality, I = Integrity, A = Availability

Threats	Description	Covered Assets
T.UWB_RETRIEVE_BINDING_DATA	Attackers retrieve the UWB binding data, e.g. a module's identifier, to impersonate the module and perform a MITM.	D.UWB_BINDING_DATA
T.UWB_MODIFICATIONS	Attackers modify the UWB-related data, thus falsifying the ranging measurement, or leading to denial of use of the secure ranging functionality.	D.UWB_BINDING_DATA D.UWB_DATA
T.UNAUTHORIZED_ACCESS_DK_ASSET	Attackers access crypto primitives using DK Applet assets (secret shared key, …) **Application Note 11:** Through relay attacks via rogue DK Applet in the Device.	D.KCMAC_KEY D.IMMOTOKEN D.SECRET_SHARED_KEY D.LONG_TERM_KEY D.DEVICE_KEY_PAIR D.APPLET_ROOT_KEY D.SESSION_KEYS D.UWB_URSK
T.RADIO_SNIFF	An attacker may attempt to sniff the traffic between a device and a vehicle during an exchange.	D.SESSION_KEYS D.SECRET_SHARED_KEY D.SEC_ATTRIBUTES D.OWNER_DK_SECRET D.OWNER_DK_DATA D.OWNER_DATA D.DK_API_DATA D.MESSAGES_EXCHANGES D.UWB_URSK
T.RADIO_MITM	An attacker may attempt to gain a MITM presence between a device and a vehicle during an exchange and might be able to modify the keys being shared.	D.SESSION_KEYS D.KCMAC_KEY D.IMMOTOKEN D.SECRET_SHARED_KEY D.DEVICE_KEY_PAIR D.UWB_URSK

6.3.3 *OSPs*

Table 6-3 describes the organizational security policies for the secure ranging that are specific to the UWB-based secure ranging functionality.

Table 6-3: FP UWB-SR OSPs

OSPs	Description
OSP.UWB_SECURE_RANGING_SUPPORT	The TOE provides support for UWB-based secure ranging and enforces the binding with the UWB module.

6.3.4 *Assumptions*

Table 6-4 describes the assumptions on the operational environment of the TOE that are specific to the UWB-based secure ranging functionality.

Table 6-4: FP UWB-SR Assumptions

Assumptions	Description
A.UWB_MODULE	The UWB module is a trusted component. It does not expose the assets that are sent to it by the TOE, either in confidentiality or integrity.

6.4 Security Objectives

6.4.1 *Security Objectives for the TOE*

Table 6-5 describes the security objectives for the TOE that are specific to the UWB-based secure ranging functionality.

Table 6-5: FP UWB-SR Security Objectives for the TOE

Security Objectives	Description
O.UWB_BINDING	The TOE SHALL ensure the binding with the UWB module.
O.UWB_SECURE_RANGING_SUPPORT	The TOE SHALL support the UWB module to perform the secure ranging.
O.UWB_BD_CONFID	The TOE SHALL ensure the confidentiality of the binding data of the UWB module.
O.UWB_BD_INTEG	The TOE SHALL ensure the integrity of the binding data of the UWB module.
O.UWB_DATA_INTEG	The TOE SHALL ensure the integrity of the UWB-related data.
O.UWB_URSK_CONFID	The TOE SHALL ensure the confidentiality of the URSK.
O.UWB_URSK_INTEG	The TOE SHALL ensure the integrity of the URSK.

6.4.2 *Security Objectives for the TOE Operational Environment*

Table 6-6 describes the security objectives for the TOE operational environment that are specific to the UWB-based secure ranging functionality.

Table 6-6: FP UWB-SR Security Objectives for the TOE Operational Environment

Security Objectives	Description
OE.UWB_MODULE	The UWB module SHALL ensure that the sensitive assets it receives from the TOE is protected in integrity and/or confidentiality as applicable, against both physical and logical attacks.

6.5 Security Functional Requirements

6.5.1 *Introduction*

The SFRs that apply to the UWB-based secure ranging functionality of the Digital Key are listed in Table 6-7. This includes three types of SFRs:

- SFRs that are defined in the core DK Applet PP, shown in grey in the table. These are not reproduced in this package as any conformant ST must include them.

- Iterations of SFRs that are defined in the core DK Applet PP. These are defined in this package.
- New SFRs, defined in this package.

Table 6-7: FP UWB-SR Security Functional Requirements

SFRs	Defined in the core PP	New iteration of an SFR defined in the core PP	Optional SFR (conditional type)	New SFR
FCS_CKM.5/UWB_URSK Cryptographic key derivation		x		
FCS_CKM.5/UWB_URSK_subkeys Cryptographic key derivation		x	x	
FCS_CKM.6 Timing and event of cryptographic key destruction	x			
FCS_COP.1.1/HMAC Cryptographic operation	x			
FCS_COP.1.1/UWB_CCM* Cryptographic operation		x	x	
FCS_COP.1.1/UWB_CMAC-256 Cryptographic operation		x	x	
FCS_RNG.1 Random number generation	x			
FDP_SDC.2/UWB Stored data confidentiality with dedicated method				x
FDP_SDI.2 Stored data integrity monitoring and action	x			
FPT_PHP.3 Resistance to physical attack	x			
FPT_ITC.1 Inter-TSF confidentiality during transmission	x			
FPT_ITI.1/UWB Inter-TSF detection of modification		x		
FPT_ITT.1/UWB Basic internal TSF data transfer protection		x		
FPT_ITT.3/UWB TSF data integrity monitoring		x		
FTP_ITC.1/UWB Inter-TSF trusted channel		x		

6.5.2 *Cryptographic key derivation*

FCS_CKM.5/UWB_URSK Cryptographic key derivation

Hierarchical to:	**No other components.**
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1

	Cryptographic operation FCS_CKM.6 Timing and event of cryptographic key destruction
FCS_CKM.5.1/UWB_URSK	The TSF shall derive cryptographic AES [84] keys from D.SECRET_SHARED_KEY [85] in accordance with a specified key derivation algorithm HKDF-SHA-256 [86] and specified cryptographic key sizes 256 [87] that meet the following: IETF [RFC 5869] [88].

6.5.3 *Stored data confidentiality with dedicated method*

FDP_SDC.2/UWB Stored data confidentiality with dedicated method

Hierarchical to:	FDP_SDC.1 Stored data confidentiality
Dependencies:	FCS_COP.1
FDP_SDC.2.1/UWB	The TSF shall ensure the confidentiality of the [selection: *all user data, the following user data [assignment: list of user data]*] according to [assignment: *data characteristics*] while it is stored under the control of the TSF. **Application Note 30**: User data stands for the confidential assets defined in this package.
FDP_SDC.2.2/UWB	The TSF shall ensure the confidentiality of the user data specified in FDP_SDC.2.1 without user intervention.

6.5.4 *Inter-TSF detection of modification*

FPT_ITI.1/UWB Inter-TSF detection of modification

Hierarchical to:	**No other components**.
Dependencies:	No dependencies
FPT_ITI.1.1/UWB	The TSF SHALL provide the capability to detect modification of all TSF data during transmission between the TSF and another trusted IT product within the following metric: [assignment: *a defined modification metric*]. **Application Note 31:** "IT product" stands for the UWB module.
FPT_ITI.1.2/UWB	The TSF SHALL provide the capability to verify the integrity of all TSF data transmitted between the TSF and another trusted IT product and perform terminate the on-going process [89] if modifications are detected.

6.5.5 *Basic internal TSF data transfer protection*

FPT_ITT.1/UWB Basic internal TSF data transfer protection

Hierarchical to:	**No other components.**
Dependencies:	No dependencies

[84] **[assignment: *key type*]**
[85] **[assignment: *input parameters*]**
[86] **[assignment: *key derivation algorithm*]**
[87] **[assignment: *list of key sizes*]**
[88] **[assignment*: list of standards*]**
[89] **[assignment: *action to be taken*]**

FPT_ITT.1.1/UWB_URSK	The TSF SHALL protect TSF data from disclosure and modifications[90] when it is transmitted between separate parts of the TOE. **Application Note 32**: TSF data stands for D.UWB_URSK.
FPT_ITT.1.1/ UWB_BINDING_DATA	The TSF SHALL protect TSF data from disclosure and modifications[91] when it is transmitted between separate parts of the TOE. **Application Note 33**: TSF data stands for D.UWB_BINDING_DATA.
FPT_ITT.1.1/ UWB_DATA	The TSF SHALL protect TSF data from modifications[92] when it is transmitted between separate parts of the TOE. **Application Note 34**: TSF data stands for D.UWB_DATA.

6.5.6 *TSF data integrity monitoring*

FPT_ITT.3/UWB TSF data integrity monitoring

Hierarchical to:	**No other components**
Dependencies:	No dependencies
FPT_ITT.3.1/UWB_URSK	The TSF SHALL be able to detect [selection: *modification of data, substitution of data, re-ordering of data, deletion of data*, [assignment: *other integrity errors*]] for TSF data transmitted between separate parts of the TOE. **Application Note 35**: TSF data stands for D.UWB_URSK.
FPT_ITT.3.1/UWB_BINDING_DATA	The TSF SHALL be able to detect [selection: *modification of data, substitution of data, re-ordering of data, deletion of data*, [assignment: *other integrity errors*]] for TSF data transmitted between separate parts of the TOE. **Application Note 36**: TSF data stands for D.UWB_BINDING_DATA.
FPT_ITT.3.1/UWB_DATA	The TSF SHALL be able to detect [selection: *modification of data, substitution of data, re-ordering of data, deletion of data*, [assignment: *other integrity errors*]] for TSF data transmitted between separate parts of the TOE. **Application Note 37**: TSF data stands for D.UWB_DATA.

6.5.7 *Inter-TSF trusted channel*

FTP_ITC.1/UWB Inter-TSF trusted channel

Hierarchical to:	**No other components**
Dependencies:	No dependencies
FTP_ITC.1.1/UWB	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. **Application Note 38**: "IT product" stands for the UWB module. **Application Note 39**: The TSF identifies the UWB module with D.UWB_BINDING_DATA.

[90] **[selection: *disclosure, modification*]**
[91] **[selection: *disclosure, modification*]**
[92] **[selection: *disclosure, modification*]**

FTP_ITC.1.2/UWB	The TSF shall permit [selection: *the TSF, another trusted IT product*] to initiate communication via the trusted channel.
FTP_ITC.1.3/UWB	The TSF shall initiate communication via the trusted channel for secure ranging[93].

6.5.8 *Optional (Conditional) Security Functional Requirements*

The cryptographic SFRs defined in sections 6.5.8.1 and 6.5.8.2 apply if the TOE has the ability to derive the URSK subkeys as part of D.UWB_URSK and perform cryptographic operations using these keys, regardless of whether the functionality is indeed performed in the field by the TOE and not the UWB module. Either all three SFRs apply, or none applies.

6.5.8.1 Optional (conditional) cryptographic key derivation

FCS_CKM.5/UWB_URSK_subkeys Cryptographic key derivation

Hierarchical to:	**No other components.**
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.6 Timing and event of cryptographic key destruction
FCS_CKM.5.1/UWB_URSK_subkeys	The TSF shall derive cryptographic symmetric[94] keys from URSK or its subkey mURSK [95] in accordance with a specified key derivation algorithm AES-CMAC[96] and specified cryptographic key sizes 256[97] that meet the following: NIST SP 800-38B[98]. **Application Note 40**: This SFR applies to the URSK subkeys: mURSK, mUPSK1, mUSPK2, dURSK, dUDSK. The keys mURSK, mUPSK1, mUPSK2 as well as the salt are derived from URSK directly while dURSK and dUDSK are derived from mURSK.

6.5.8.2 Optional (conditional) cryptographic operation

FCS_COP.1/UWB Cryptographic operation

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, or FCS_CKM.5 Cryptographic key derivation] FCS_CKM.6 Timing and event of cryptographic key destruction
FCS_COP.1.1/UWB_CCM*	The TSF SHALL perform data encryption or decryption[99] in accordance with a specified cryptographic algorithm AES with CCM mode of operation[100] and cryptographic key sizes 128-bit[101]

[93] **[assignment: *list of functions for which a trusted channel is required*]**
[94] **[assignment: *key type*]**
[95] **[assignment: *input parameters*]**
[96] **[assignment: *key derivation algorithm*]**
[97] **[assignment: *list of key sizes*]**
[98] **[assignment*: list of standards*]**
[99] **[assignment: *list of cryptographic operations*]**
[100] **[assignment: *cryptographic algorithm*]**
[101] **[assignment: *cryptographic key sizes*]**

	that meet the following: IEEE Std 802.15.4TM-2020[102]. **Application Note 41**: used for UWB payload encryption as shown in section 22.1 of [CCC-DK-TS] (with mUPSK1 and dUDSK) and the derivation of the STS sequence (with dURSK).
FCS_COP.1.1/UWB_CMAC-256	The TSF SHALL perform Message Authentication Code[103] in accordance with a specified cryptographic algorithm AES CMAC[104] and cryptographic key sizes 256-bit[105] that meet the following: NIST SP 800-38B[106]. **Application Note 42**: used for UWB subkeys and nonces derivation as shown in section 22.1 of [CCC-DK-TS] (with URSK, mURSK, mUPSK2 and the salt).

6.6 Rationales

6.6.1 *Rationale for the Security Objectives*

The mapping of the SPD elements and the security objectives defined in this package is presented in the Table 6-8.

Table 6-8: FP UWB-SR Mapping of SPD and Objectives

SPD	O.UWB_BINDING	O.UWB_SECURE_RANGING_SUPPORT	O.UWB_URSK_CONFID	O.UWB_URSK_INTEG	O.UWB_BD_CONFID	O.UWB_BD_INTEG	O.UWB_DATA_INTEG	OE.UWB_MODULE
T.UWB_RETRIEVE_URSK	x		x					
T.UWB_RETRIEVE_BINDING_DATA	x				x			
T.UWB_MODIFICATIONS	x					x	x	
T.UNAUTHORIZED_ACCESS_DK_ASSET								
T.RADIO_SNIFF	x			x				
T.RADIO_MITM	x			x				
OSP.UWB_SECURE_RANGING_SUPPORT		x						

[102] **[assignment: *list of standards*]**
[103] **[assignment: *list of cryptographic operations*]**
[104] **[assignment: *cryptographic algorithm*]**
[105] **[assignment: *cryptographic key sizes*]**
[106] **[assignment: *list of standards*]**

SPD	O.UWB_BINDING	O.UWB_SECURE_RANGING_SUPPORT	O.UWB_URSK_CONFID	O.UWB_URSK_INTEG	O.UWB_BD_CONFID	O.UWB_BD_INTEG	O.UWB_DATA_INTEG	OE.UWB_MODULE
A.UWB_MODULE								x

Note that the mapping of the threat T.UNAUTHORIZED_ACCESS_DK_ASSET defined in the core DK Applet PP, which applies to the URSK, is unchanged (it does not map to any new objective defined in this package).

Table 6-9, Table 6-10 and Table 6-11 provide the rationales of the coverage of the threats, OSPs and assumptions defined in this package by the security objectives. The greyed rationales are copied from the core DK Applet PP.

Table 6-9: FP UWB-SR Threats and Objectives - Coverage

Threats	Security Objectives	Rationale
T.UWB_RETRIEVE_URSK	O.UWB_URSK_CONFID O.UWB_BINDING	This threat is covered by • O.UWB_URSK_CONFID which guarantees that the TOE ensures the confidentiality of the URSK within the TOE. • O.UWB_BINDING which guarantees that the URSK is transmitted safely to the UWB module.
T.UWB_RETRIEVE_BINDING_DATA	O.UWB_BD_CONFID O.UWB_BINDING	This threat is covered by • O.UWB_BD_CONFID which guarantees that the TOE ensures the confidentiality of the binding data between the SE and the UWB module while stored within the TOE. • O.UWB_BINDING which ensures the binding is done with the legitimate UWB module and not another, potentially malicious component which would gain unallowed access to UWB binding data.
T.UWB_MODIFICATIONS	O.UWB_BD_INTEG O.UWB_DATA_INTEG	This threat is covered by the following three security objectives

Threats	Security Objectives	Rationale
	O.UWB_BINDING	• O.UWB_BD_INTEG which ensures that the TOE ensures the integrity of the UWB binding data when it is generated, stored, deleted and processed. • O.UWB_DATA_INTEG which ensures the integrity of the UWB data used for secure ranging during generation, storage, deletion and processing. • O_UWB_BINDING which ensures the binding is done with the legitimate UWB module and not another, potentially malicious component which would be able to modify the UWB related assets.
T.RADIO_MITM	O.KCMAC_KEY_INTEG O.SEC_SHARED_KEY_INTEG O.LONG_TERM_KEY_INTEG O.SESSION_KEYS_INTEG O.FAST_TRANSACTION_AUTH O.STD_TRANSACTION_AUTH O.KEY_EXCHANGE_AUTH O.UWB_URSK_INTEG O.UWB_BINDING	This threat is covered by • O.KCMAC_KEY_INTEG which ensures the integrity of Kcmac key. • O.SEC_SHARED_KEY_INTEG which ensures the integrity of secret shared keys • O.LONG_TERM_KEY_INTEG which ensures the integrity of long term key • O.SESSION_KEYS_INTEG which ensures the integrity of session keys • O.FAST_TRANSACTION_AUTH which ensures the authentication takes place from device side • O.STD_TRANSACTION_AUTH which ensures that mutual authentication takes place between device and vehicle. • O.KEY_EXCHANGE_AUTH ensures the authenticity of key share operation. • O.UWB_URSK_INTEG which ensures the integrity of the URSK. • O.UWB_BINDING which ensures that the UWB module is not impersonated by an attacker.
T.RADIO_SNIFF	O.FAST_TRANSACTION_AUTH O.STD_TRANSACTION_AUTH O.UWB_URSK_INTEG O.UWB_BINDING	This threat is covered by • O.FAST_TRANSACTION_AUTH • O.STD_TRANSACTION_AUTH which ensure a secure channel is used when communicating between device and vehicle • O.UWB_URSK_INTEG which ensures the integrity of the URSK.

Threats	Security Objectives	Rationale
		• O.UWB_BINDING which ensures that the UWB module is not impersonated by an attacker.

Table 6-10: FP UWB-SR OSPs and Objectives - Coverage

Organizational Security Policies	Security Objectives	Rationale
OSP.UWB_SECURE_RANGING_SUPPORT	O.UWB_SECURE_RANGING_SUPPORT	This OSP is enforced by the security objectives for the TOE O.UWB_SECURE_RANGING_SUPPORT, for the support of the secure ranging functionality, and O.UWB_BINDING which ensures that the ranging is performed through the legitimate UWB module.

Table 6-11: FP UWB-SR Assumptions and Objectives - Coverage

Assumptions	Security Objectives for the Operational Environment	Assumptions
A.UWB_MODULE	OE.UWB_MODULE	This assumption is directly upheld by OE.UWB_MODULE

6.6.2 *Rationale for the Security Functional Requirements*

The mapping of the security objectives for the TOE defined in this package and the SFRs is presented in the Table 6-12. Note that the iterated components are referenced to allow a fine-grained mapping.

The Table 6-13 presents the rationales of coverage of the security objectives for the TOE defined in this package.

Table 6-12: FP UWB-SR Mapping of Security Objectives for the TOE and SFRs

SFR	O.UWB_BINDING	O.UWB_SECURE_RANGING_SUPPORT	O.UWB_URSK_CONFID	O.UWB_URSK_INTEG	O.UWB_BD_CONFID	O.UWB_BD_INTEG	O.UWB_DATA_INTEG
FCS_CKM.5/UWB_URSK Cryptographic key derivation		x					

SFR	O.UWB_BINDING	O.UWB_SECURE_RANGING_SUPPORT	O.UWB_URSK_CONFID	O.UWB_URSK_INTEG	O.UWB_BD_CONFID	O.UWB_BD_INTEG	O.UWB_DATA_INTEG
Optional (conditional) FCS_CKM.5/UWB_URSK_subkeys Cryptographic key derivation		x					
FCS_CKM.6 Timing and event of cryptographic key destruction		x					
FCS_COP.1.1/HMAC Cryptographic operation		x					
Optional (conditional) FCS_COP.1.1/UWB_CCM* Cryptographic operation		x					
Optional (conditional) FCS_COP.1.1/UWB_CMAC-256 Cryptographic operation		x					
FCS_RNG.1 Random number generation		x					
FDP_SDC.2/UWB Stored data confidentiality with dedicated method			x		x		
FDP_SDI.2 Stored data integrity monitoring and action				x		x	x
FPT_PHP.3 Resistance to physical attack			x	x	x	x	x
FPT_ITC.1 Inter-TSF confidentiality during transmission			x		x		
FPT_ITI.1/UWB Inter-TSF detection of modification				x		x	x
FPT_ITT.1/UWB_URSK Basic internal TSF data transfer protection			x	x			
FPT_ITT.1/UWB_BINDING_DATA Basic internal TSF data transfer protection					x	x	
FPT_ITT.1/UWB_DATA Basic internal TSF data transfer protection							x
FPT_ITT.3/UWB_URSK TSF data integrity monitoring				x			
FPT_ITT.3/UWB_BINDING_DATA TSF data integrity monitoring						x	
FPT_ITT.3/UWB_DATA TSF data integrity monitoring							x
FTP_ITC.1/UWB Inter-TSF trusted channel	x		x	x	x	x	x

Table 6-13: FP UWB-SR Security Objectives for the TOE and SFRs - Coverage

Security Objectives	SFR	Rationale
O.UWB_BINDING	FTP_ITC.1/UWB	The security objective O.UWB_BINDING is met by FTP_ITC.1/UWB which enforces the communication through secure channel between the TOE and the identified UWB module.
O.UWB_SECURE_RANGING_SUPPORT	FCS_CKM.5/UWB_URSK FCS_CKM.6 FCS_COP.1.1/UWB_CCM* FCS_RNG.1 Optional (conditional): FCS_CKM.5/UWB_URSK_subkeys FCS_COP.1.1/UWB_CMAC-256 FCS_COP.1.1/HMAC	The security objective O.UWB_SECURE_RANGING_SUPPORT is met by the following SFRs: • FCS_CKM.5/UWB_URSK which ensures the generation of the URSK • FCS_COP.1.1/HMAC which enforces an algorithm supporting the secure ranging • FCS_RNG.1 which enforces the quality of the random numbers that are necessary for the cryptographic operations related to secure ranging. Optional (conditional): • FCS_CKM.5/UWB_URSK_subkeys which ensures the derivation of URSK subkeys • FCS_COP.1.1/UWB_CCM*, FCS_COP.1.1/UWB_CMAC-256 which enforce the algorithms supporting the secure ranging
O.UWB_URSK_CONFID	FDP_SDC.2 FPT_ITC.1 FPT_ITT.1.1/UWB_URSK FPT_PHP.3 FTP_ITC.1/UWB	The security objective O.UWB_URSK_CONFID is met by the following SFR: • FDP_SDC.2 which enforces the protection of the asset in terms of confidentiality while it's being stored. • FPT_ITC.1 which ensures the confidentiality during the transmission. • FPT_ITT.1.1/UWB_URSK which ensures that the URSK is protected when transmitted between separate parts of the TOE against disclosure. • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing. • FTP_ITC.1/UWB which ensures the confidentiality protection of the communication between the SE and the UWB module.
O.UWB_URSK_INTEG	FDP_SDI.2 FPT_ITI.1/UWB FPT_ITT.1.1/UWB_URSK	The security objective O.UWB_URSK_INTEG is met by the following SFR:

Security Objectives	SFR	Rationale
	FPT_ITT.3.1/UWB_URSK FPT_PHP.3 FTP_ITC.1/UWB	• FDP_SDI.2 which monitors stored user data for integrity errors. • FPT_ITI.1/UWB which requires that modifications of UWB related data and keys data are detected when transmitted between the TOE and the UWB module. • FPT_ITT.1.1/UWB_URSK which ensures that the URSK is protected when transmitted between separate parts of the TOE against modifications. • FPT_ITT.3.1/UWB_URSK which ensures that the URSK transmitted between separate parts of the TOE is monitored for identified integrity errors and actions are taken in the event of an integrity violation detection. • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing. • FTP_ITC.1/UWB which ensures the integrity protection of the communication between the SE and the UWB module.
O.UWB_BD_CONFID	FDP_SDC.2 FPT_ITC.1 FPT_ITT.1.1/UWB_BINDING_DATA FPT_PHP.3 FTP_ITC.1/UWB	The security objective O.UWB_BD_CONFID is met by the following SFR: • FDP_SDC.2 which enforces the protection of the asset in terms of confidentiality while it's being stored. • FPT_ITC.1 which ensures the confidentiality during the transmission. • FPT_ITT.1.1/UWB_BINDING_DATA which ensures that the UWB binding data is protected when transmitted between separate parts of the TOE against disclosure. • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing. • FTP_ITC.1/UWB which ensures the confidentiality protection of the communication between the SE and the UWB module.
O.UWB_BD_INTEG	FDP_SDI.2 FPT_ITI.1/UWB FPT_ITT.1.1/UWB_BINDING_DATA FPT_ITT.3.1/UWB_BINDING_DATA	The security objective O.UWB_BD_INTEG is met by the following SFRs: • FDP_SDI.2 which monitors stored user data for integrity errors.

Security Objectives	SFR	Rationale
	FPT_PHP.3 FTP_ITC.1/UWB	• FPT_ITI.1/UWB which requires that modifications of UWB related data and keys data are detected when transmitted between the TOE and the UWB module. • FPT_ITT.1/UWB_BINDING_DATA which ensures that the URSK is protected when transmitted between separate parts of the TOE against modifications. • FPT_ITT.3/UWB_BINDING_DATA which ensures that UWB binding data transmitted between separate parts of the TOE is monitored for identified integrity errors and actions are taken in the event of an integrity violation detection. • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing. • FTP_ITC.1/UWB which ensures the integrity protection of the communication between the SE and the UWB module.
O.UWB_DATA_INTEG	FDP_SDI.2 FPT_ITI.1/UWB FPT_ITT.1.1/UWB_DATA FPT_ITT.3.1/UWB_DATA FPT_PHP.3 FTP_ITC.1/UWB	The security objective O.UWB_ DATA_INTEG is met by the following SFRs: • FDP_SDI.2 which monitors stored user data for integrity errors. • FPT_ITI.1/UWB which requires that modifications of UWB related data and keys data are detected when transmitted between the TOE and the UWB module. • FPT_ITT.1.1/UWB_DATA which ensures that the URSK is protected when transmitted between separate parts of the TOE against modifications. • FPT_ITT.3.1/UWB_DATA which ensures that UWB data transmitted between separate parts of the TOE is monitored for identified integrity errors and actions are taken in the event of an integrity violation detection. • FPT_PHP.3 which enforces the appropriate mechanisms to continuously counter physical manipulation and physical probing. • FTP_ITC.1/UWB which ensures the integrity protection of the

Security Objectives	SFR	Rationale
		communication between the SE and the UWB module.

6.6.3 *Rationale for the Exclusion of Dependencies*

The Table 6-14 shows the satisfied and unsatisfied dependencies.

The dependency FCS_COP.1 of FDP_SDC.2 is not satisfied because this package does not mandate the use of cryptography for the realization of the data confidentiality protection.

Application Note 43: The ST author should add FCS_COP.1 if the TOE does use cryptography for this purpose.

Table 6-14: FP UWB-SR SFR Dependencies

SFRs	Satisfied dependencies	Unsatisfied dependencies
FCS_CKM.5/UWB_URSK Cryptographic key derivation	FCS_COP.1.1/HMAC Cryptographic operation FCS_CKM.6 Timing and event of cryptographic key destruction	
Optional (conditional) FCS_CKM.5/UWB_URSK_subkeys Cryptographic key derivation	FCS_COP.1.1/UWB-CMAC-256 Cryptographic operation FCS_CKM.6 Timing and event of cryptographic key destruction	
FCS_CKM.6 Timing and event of cryptographic key destruction		
FCS_COP.1.1/HMAC Cryptographic operation		
Optional (conditional) FCS_COP.1.1/UWB_CCM* Cryptographic operation	FCS_CKM.5/UWB_URSK_subkeys (for mUPSK1, dUDSK) Cryptographic key derivation FCS_CKM.6 Timing and event of cryptographic key destruction	
Optional (conditional) FCS_COP.1.1/UWB_CMAC-256 Cryptographic operation	FCS_CKM.5/UWB_URSK Cryptographic key derivation FCS_CKM.5/UWB_URSK_subkeys (for mURSK) Cryptographic key derivation FCS_CKM.6 Timing and event of cryptographic key destruction	
FCS_COP.1.1/UWB_AES-CTR Cryptographic operation	FCS_CKM.5/UWB_URSK_subkeys (for dURSK) Cryptographic key derivation FCS_CKM.6 Timing and event of cryptographic key destruction	
FCS_RNG.1 Random number generation		
FDP_SDC.2/UWB Stored data confidentiality with dedicated method		FCS_COP.1
FDP_SDI.2 Stored data integrity monitoring and action		

SFRs	Satisfied dependencies	Unsatisfied dependencies
FPT_PHP.3 Resistance to physical attack		
FPT_ITC.1 Inter-TSF confidentiality during transmission		
FPT_ITI.1/UWB Inter-TSF detection of modification	No dependencies	
FPT_ITT.1/UWB_URSK Basic internal TSF data transfer protection	No dependencies	
FPT_ITT.1/UWB_BINDING_DATA Basic internal TSF data transfer protection	No dependencies	
FPT_ITT.1/UWB_DATA Basic internal TSF data transfer protection	No dependencies	
FPT_ITT.3/UWB_URSK TSF data integrity monitoring	No dependencies	
FPT_ITT.3/UWB_BINDING_DATA TSF data integrity monitoring	No dependencies	
FPT_ITT.3/UWB_DATA TSF data integrity monitoring	No dependencies	
FTP_ITC.1/UWB Inter-TSF trusted channel	No dependencies	

REFERENCES

Short Name	Description
[AIS20]	Functionality classes and evaluation methodology for deterministic random number generators, reference: AIS 20, BSI (latest version)
[AIS31]	Functionality classes and evaluation methodology for physical random number generators, reference: AIS31, BSI (latest version)
[BSI TR-03111]	Technical Guideline BSI TR-03111 - Elliptic Curve Cryptography - Version 2.10- Date: 2018-06-01
[CC]	Common Criteria for Information Technology Security Evaluation documents version CC:2022, Revision 1 - Parts 1-5 - https://www.commoncriteriaportal.org/cc/
[CC1]	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model. Version CC:2022. Revision 1. Nov 2022. CCMB- 2022-11-001.
[CC2]	Common Criteria for Information Technology Security Evaluation, Part 2: Security functional requirements. Version CC:2022. Revision 1. Nov 2022. CCMB-2022-11-002.
[CC3]	Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance requirements. Version CC:2022. Revision 1. Nov 2022. CCMB-2022-11-003.
[CC5]	Common Criteria for Information Technology Security Evaluation, Part 5: Pre-defined packages of security requirements. Version CC:2022. Revision 1. Nov 2022. CCMB-2022-11-005.
[CCC-DK-TS]	Car Connectivity Consortium Digital Key - Technical Specification Version 3.1.3 (CCC-TS-101)
[CEM]	Common Methodology for Information Technology Security Evaluation, Evaluation Methodology. Version CEM:2022. Revision 1. Nov 2022. CCMB-2022-11-006.
[FIPS140-3]	Security Requirements for Cryptographic Modules - FIPS PUB 140-3 Federal Information Processing Standards Publication (Supersedes FIPS PUB 140-2)
[FIPS PUB 186-4]	The FIPS 186-4 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS) - March 18, 2014
[FIPS PUB 197]	Advanced Encryption Standard (AES) – FIPS 197 – November 26, 2001
[GP]	GlobalPlatform Specifications version 2.3.1 https://globalplatform.org/wp-content/uploads/2018/05/GPC_CardSpecification_v2.3.1_PublicRelease_CC.pdf
[JIL-attacks]	JIL Application of Attack Potential to Smartcards, version 3.2.1
[PP0084]	Security IC Platform Protection Profile with Augmentation Packages - Registered and Certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-CC-PP-0084-2014
[PP0099]	Java Card System – Open Configuration Protection Profile – July 2024 – Version 3.2
[SEPP]	GlobalPlatform Secure Element Protection Profile, GCP_SPE_174, version 1.0, February 2021 (or latest applicable version)
[RFC2119]	Key words for use in RFCs to Indicate Requirement Levels, Updated by 8174: http://www.ietf.org/rfcs/rfc2119.txt
[RFC 5869]	IETF - HMAC-based Extract-and-Expand Key Derivation Function (HKDF) May 2010
[SP-800-38A]	Recommendation for Block Cipher Modes of Operation: Methods and Techniques – NIST Special Publication 800-38A – December 2001

Short Name	Description
[SP-800-38B]	Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication – NIST Special Publication 800-38B – May 2005
[SP-800-56A]	Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography - SP 800-56A Rev. 3 - April 2018
[X9.62a]	The Elliptic Curve Digital Signature Algorithm (ECDSA), American National Standard for Financial Services, November 16, 2005
[X9.63]	Public Key Cryptography for the Financial Services Industry - Key Agreement and Key Transport Using Elliptic Curve Cryptography