Global Platform®

The standard for secure digital services and devices

GlobalPlatform Technology

FIDO2 SE Protection Profile

(for CTAP v2.1 and FIDO Authenticator Security and Privacy requirements v 1.5)

Version 0.0.0.17

Public Review Draft

September 2024

Document Reference: GPC_SPE_210

THIS SPECIFICATION OR OTHER WORK PRODUCT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE COMPANY, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER DIRECTLY OR INDIRECTLY ARISING FROM THE IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT.

Contents

Tables

Figures

1 INTRODUCTION

FIDO2 authentication provides a strong and secure authentication mechanism for web users while preserving their privacy that is based on the CTAP protocol[CTAP] which distinguishes between:

- Roaming authenticators, which are implemented externally, support the CTAP protocol and can communicate based on different technologies (e.g. USB, Bluetooth, or NFC);
- Platform authenticators, which are embedded inside a device such as a laptop or a mobile phone and cannot be disconnected from the device, e.g. a laptop with a Touch Bar supporting Touch ID fingerprint recognition.

The present document focuses specifically on external, off-device roaming authenticators consisting of a Secure Element (SE) that is conformant with GlobalPlatform SE Protection Profile [GPPPSE] on top of which a FIDO2 Authenticator Application (FIDO2 AA) allows to securely access online services and perform FIDO2 authentication. It covers Consumer and Enterprise authenticator modes[1].

This document is named as the FIDO2 SE Protection Profile (PP). It claims conformance to EAL4 augmented with ALC_DVS.2 and AVA_VAN.5.

1.1 Identification

Title	FIDO2 SE Protection Profile
Identification	GPC_SPE_210
Date	September 2024
Version	0.0.0.17
Sponsor	GlobalPlatform, Inc.
Editor	Carolina Lavatelli, Internet of Trust
CC Version	3.1 Revision 5

1.2 Audience

This document is intended primarily for the use of FIDO2 Authenticator developers and integrators, service providers, as well as evaluation laboratories, certification bodies, and consumers of GlobalPlatform and Common Criteria certificates.

1.3 IPR Disclaimer

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

[1] A Consumer Authenticator does not contain code supporting Enterprise Attestation. An Enterprise Attestation Authenticator does contain code to support Enterprise Attestation functionality and can be provisioned/configured by the Vendor for that purpose.

1.4 References

This section lists references applicable to this specification. The latest version of each reference applies unless a publication date or version is explicitly stated.

Table 1-1: References

Standard / Specification	Description	Ref
CC Part 1	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, April 2017, Version 3.1 revision 5	[CC1]
CC Part 2	Common Criteria for Information Technology Security Evaluation, Part 2: Security functional requirements, April 2017, Version 3.1 revision 5	[CC2]
CC Part 3	Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, April 2017, Version 3.1 revision 5	[CC3]
CTAP 2.1	Client to Authenticator Protocol (CTAP) Proposed Standard, June 21 2022	[CTAP]
FIDO Authenticator Allowed Cryptography	FIDO Authenticator Allowed Cryptography List, November 2021	[FCrypto]
FIDO Authenticator Security Requirements	FIDO Authenticator Security and Privacy Requirements, Version 1.5, November 2021	[FReq]
FIDO Security Reference	FIDO Security Reference Proposed Standard, Version 2.1, 23 May 2022	[FSec]
GPC_SPE_034	GlobalPlatform Card Specification v2.3	[GPCS]
GPC_SPE_174	GlobalPlatform Secure Element Protection Profile v1.0	[GPPPSE]
IETF RFC 2119	Key words for use in RFCs to Indicate Requirement Levels	[RFC 2119]
IETF RFC 5280	Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile	[RFC 5280]
Java Card API	Application Programming Interface, Java Card™ Platform, v2.2 to v3.1 Classic Edition, Oracle	[JCAPI]
Java Card VM	Virtual Machine Specification, Java Card™ Platform, v2.2 to v3.1 Classic Edition, Oracle	[JCVM]
Java Card JCRE	Runtime Environment Specification, Java Card™ Platform, v2.2 to v3.1 Classic Edition, Oracle	[JCRE]
Web Authentication	Web Authentication: An API for accessing Public Key Credentials, Level 2 (Webauthn), W3C Recommendation, April 2021	[Web Auth]

1.5 Terminology and Definitions

Selected terms used in this document are included in Table 1-2. We refer the reader to the definitions of the terms provided in the references listed in section 1.4.

Table 1-2: Terminology and Definitions

Term	Definition
Attestation	The process by which Authenticators make claims to a Relying Party that the keys they generate, and/or certain measurements they report, originate from genuine devices with certified characteristics.
Attestation Certificate	A public key certificate related to an Attestation Key.
Attestation Key	A private or public key used for FIDO2 Authenticator attestation.
Authentication	The process in which the end-users employ their FIDO2 Authenticator to prove possession of a registered key to a Relying Party.
Authenticator	See *FIDO2 Authenticator*.
Authenticator Attestation	The process of communicating a cryptographic assertion to a Relying Party that a key presented during authenticator registration was created and protected by a genuine authenticator with verified characteristics.
Authenticator Specific Module (ASM)	Software associated with a FIDO2 Authenticator that provides a uniform interface between the hardware and FIDO Client software.
Built-in User Verification Method	The authenticator supports a built-in on-device user verification method such as fingerprint or has an input UI with secure communication to the authenticator.
Certificate	An X.509v3 certificate defined by the profile specified in [RFC 5280] and its successors.
Evidence of user interaction	Collection of *evidence of user interaction* establishes a state of *user presence*. Also, if it is collected along with displaying a particular prompt to a user it may be considered collecting *user consent*. The general notion is that the user interacts with the authenticator in some fashion, also known as supplying a “user gesture”—e.g. touches a consent button, enters a password or a PIN, or supplies a biometric—to at least confirm their presence and possibly consent to some proposed action. Some “user gesture” approaches provide user verification in addition to establishing user presence, e.g. a fingerprint-based built-in user verification method.
FIDO2 Authenticator	An authentication entity that meets the FIDO Alliance’s requirements and has related metadata. A FIDO2 Authenticator is responsible for user verification, and maintaining the cryptographic material required for the Relying Party authentication.
FIDO Client	The software entity processing the CTAP protocol messages on the FIDO User Device. FIDO Clients may take one of two forms: 1. A software component implemented in a user agent (either web browser or native application) 2. A standalone piece of software shared by several user agents (web browsers or native applications)

Term	Definition
FIDO Server	Server software typically deployed in the Relying Party's infrastructure that meets UAF protocol server requirements.
Platform Authenticator	A FIDO2 Authenticator or combination of authenticator and ASM, which uses an access control mechanism to restrict the use of registered keys to trusted FIDO Clients and/or trusted FIDO User Devices. Compare to *Roaming Authenticator*.
Registration	A CTAP protocol operation in which a user generates and associates new key material with an account at the Relying Party, subject to policy set by the server and acceptable attestation that the authenticator and registration matches that policy.
Relying Party (RP)	A web site or other entity that uses the CTAP protocol to directly authenticate users (i.e. performs peer-entity authentication). Note that if CTAP is composed with federated identity management protocols (e.g. SAML, OpenID Connect, etc.), the identity provider will also be playing the role of a FIDO Relying Party.
Roaming Authenticator	A FIDO2 Authenticator configured to move between different FIDO Clients and FIDO User Devices lacking an established trust relationship by: 1. Using only its own internal storage for registrations 2. Allowing registered keys to be employed without access control mechanisms at the API layer. (Roaming Authenticators may still perform user verification.) Compare to *Platform Authenticator*.
User Presence Check	The process of obtaining some explicit gesture from a user (i.e. a natural person) that they are present. Examples are pressing a button, touching a touch screen or pad, or any biometrics that require a conscious action from the user such as touching a fingerprint sensor (but not passive biometrics such as looking at a device or checking an EKG).
User Verification	The process of verifying that a particular user, typically a person, has supplied some input so the authenticator can know it is that person. The input is typically something only the user knowns or something the user is (biometric). This definition is primarily used to refer to a single method, not multifactor authentication based on a combination of methods. Examples are a PIN, password, or fingerprint.
Verification Factor	The specific means by which local user verification is accomplished; e.g. fingerprint, voiceprint, or PIN.

1.6 Abbreviations

Table 1-3: Abbreviations

Abbreviation	Meaning
ASM	Authenticator Specific Module
ASP	Authenticator Security Parameter
CTAP	Client to Authenticator Protocol
FIDO	Fast IDentity Online
IC	Integrated Circuit
MAC	Message Authentication Code
MACKey	MAC computation secret key
OE	Operational Environment
OS	Operating System
OSP	Organisational Security Policy
PII	Personal Identifiable Information
PP	Protection Profile
PRNG	Pseudo Random Number Generator
RP	Relying Party
SA	Security Assumption
SAR	Security Assurance Requirement
SE	Secure Element
SFP	Security Function Policy
SFR	Security Functional Requirement
SG	Security Goal
SM	Security Measure
SPD	Security Problem Definition
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality

1.7 Revision History

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

Table 1-4: Revision History

Date	Version	Description
May 2021	0.0.0.1	Initial structure
April 2022	0.0.0.9	Member Review
February 2023	0.0.0.14	Second Member Review Version / FIDO Alliance
September 2024	0.0.0.17	Public Review Draft
TBD	v1.0	Public Release

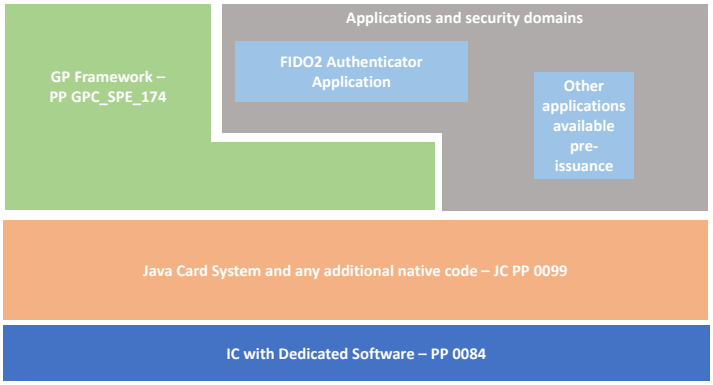
2 TOE OVERVIEW

This chapter defines the type of the Target of Evaluation (TOE), presents the high-level TOE architecture and functional principles, and describes the TOE's main security features and intended uses as well as the TOE's life cycle.

2.1 TOE Type

The TOE type is a contactless or contact-based GlobalPlatform-enabled Java Card including the FIDO2 Authenticator Application implementing the CTAP specification [CTAP] as depicted in Figure 2-1. The TOE interacts with a FIDO Client application (e.g. a web browser) which interacts with the Relying Party (RP) and indirectly with the FIDO Server. The TOE classifies in FIDO category 4, that is an entire Authenticator implemented in an Allowed Restricted Operating Environment (AROE).

Figure 2-1: TOE Components



The AROE consists of the IC, the IC Dedicated Software, the Java Card System, the GlobalPlatform Framework, and any platform native code. It must comply with GlobalPlatform SE Protection Profile [GPPPSE].

The AROE (or SE) provides the security services for key management and cryptographic operations, logical and physical protection of the application code and data, the application management, etc. It may also provide the PIN management.

The FIDO2 Authenticator Application provides essentially the non-cryptographic security functionality of the CTAP protocol.

The TOE comprises:

- All hardware, firmware, and software relied upon to provide the FIDO2 SE Authenticator security functionality.

The TOE does not comprise:

- The FIDO Client application
- The Relying Party
- The FIDO Server.

The TOE may support user verification and Enterprise Attestation[2].

The Transaction Confirmation Display is out of the scope of this Protection Profile.

Application Note: The ST author shall

1. Provide the model of the FIDO2 SE Authenticator.
2. Indicate whether the TOE is a first-factor or a second-factor Authenticator.
3. Indicate whether user verification is supported or not.
4. Indicate whether Enterprise Attestation is supported or not.
5. Indicate whether Signature Counters are supported or not.
6. Indicate whether Transaction Confirmation Display is supported or not.
7. Provide the overall cryptographic strength[3], which shall be less than or equal to the claimed cryptographic strength of all the Authenticator Security Parameters that are cryptographic keys.
8. Describe the boundary of the TOE, i.e. all hardware and software used for user presence check, user verification, key generation, signature generation, etc.
9. If the Transaction Confirmation Display is supported, describe where and how this is implemented, and explain any addition in the Security Target compared to this PP.
10. If Signature Counters are supported, it shall be documented whether one Signature Counter per authentication key or one (global) Signature Counter for all authentication keys (i.e. at least one counter covering multiple keys) is implemented.

2.2 TOE Description

2.2.1 FIDO2 Authentication Overview

The main goal of FIDO2-based authentication is to provide a strong authentication mechanism for web users while preserving their privacy. To this end, FIDO2 relies on public-key cryptography to support strong, password-free, multi-factor authentication to end-users based on (internal) platform authenticators (such as biometrics or PINs) or external authenticators (such as FIDO Security Keys, mobile devices, wearables, etc.).

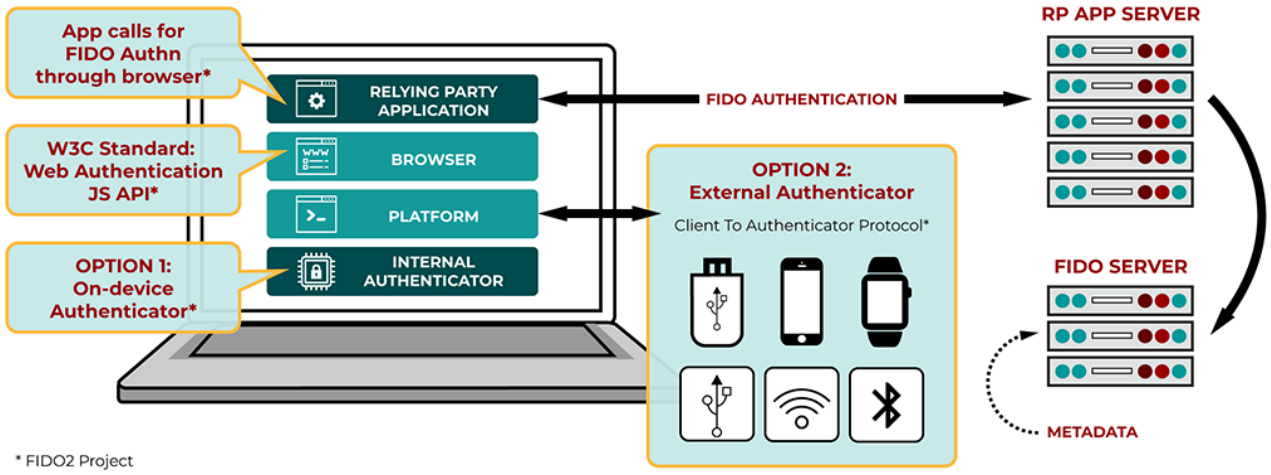
Conceptually, FIDO2 authentication involves an exchange between a computing environment controlled by a Relying Party (RP, as shown in Figure 2-2 on the right side) and a client requiring access to a service offered by the Relying Party. The client (as shown in Figure 2-2 on the left side) runs in an environment controlled by the user to be authenticated and communicates with a FIDO2 Authenticator. The Relying Party's environment consists of at least one application server plus a FIDO Server, which has a trusted storage containing the public trust anchors for the attestation of FIDO2 Authenticators. The user's environment consists of one or more internal/external FIDO2 Authenticators and the platform on which the FIDO Client is running. The amount

[2] Concerning privacy, an exception is made for FIDO2 SE Authenticators that have Enterprise Attestation enabled. These have a global identifier within an Enterprise Attestation enabled Enterprise network or within the Enterprise's Relying Party, where an Enterprise is some form of organization. The Relying Parties of the Enterprise's Data Processors are part of the Relying Party of the Enterprise. Enterprise Attestation enabled Authenticators are solely for use by the Enterprise's employees, contractors, and defined members, not its customers.

[3] The security strength is a number specified in bits and representing the number of operations required to break a cryptographic algorithm or system.

of personal identifiable information exposed by the FIDO2 SE Authenticator to a Relying Party is limited to the absolute minimum.

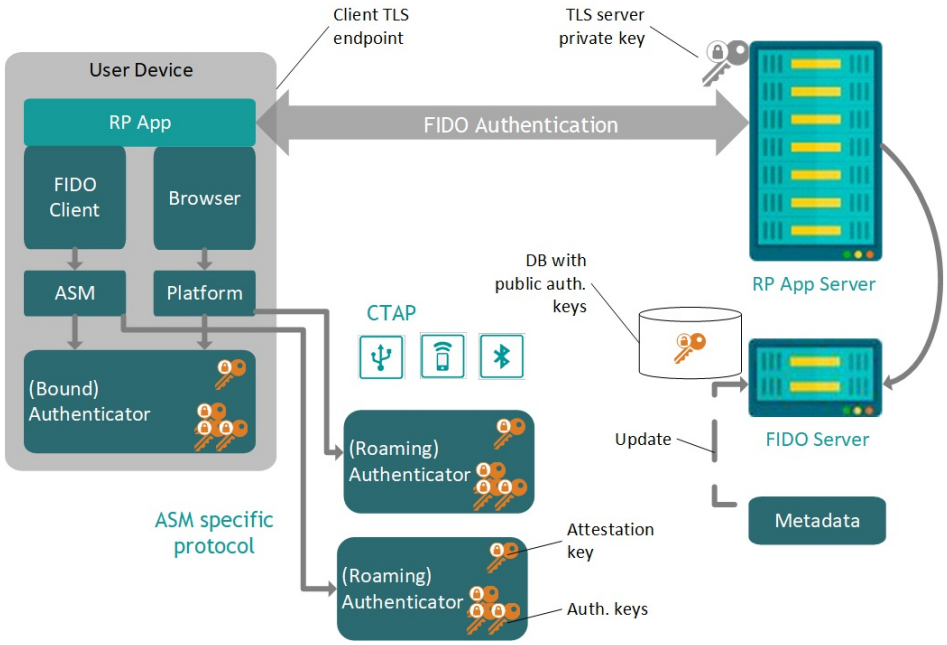
Figure 2-2: FIDO2 Overview (from FIDO Alliance)



FIDO2 Authenticators are responsible for user registration and authentication and for managing the cryptographic material required for authentication to the Relying Party. The general FIDO2 architecture, communication protocols, and main cryptographic keys are illustrated in Figure 2-3.

Although a large part of the specifications applies to any type of FIDO2 Authenticator, in the following sections we refer specifically to FIDO2 SE Authenticators, which are a form of external roaming authenticator.

Figure 2-3: FIDO2 Reference Architecture (from FIDO Alliance)



2.2.2 Functional Description

FIDO2 SE Authenticators' main functionality include:

- Registration to an online service

- Authentication to online services
- PIN management
- Reset to delivery state.

For both registration and authentication, a seed, and a secret key for MAC computation, i.e. a MACKey, are used. The seed and MACKey are different from each other, and they are generated during the initialisation process, either during the pre-personalisation phase performed by the manufacturer and described in further detail in section 2.6 or during the first usage or reset of the FIDO2 SE Authenticator by the user. Both the seed and the MACKey are stored securely on the FIDO2 SE Authenticator.

2.2.2.1 Registration

Registration is the process by which a user registers a new account on a website/service supporting the WebAuthn API [Web Auth]. The interactions happening during registration between the FIDO2 SE Authenticator, the browser running on the user device, and the Relying Party are shown in Figure 2-4.

During registration[4], a specific asymmetric key pair (not shared across websites/services) is generated and stored on the FIDO2 SE Authenticator. First, the Relying Party "sends an identifier"[5] to the FIDO2 SE Authenticator as well as a series of parameters, such as whether local authentication is required, that must be fulfilled for the registration to be performed. Then, one of the following processes takes place:

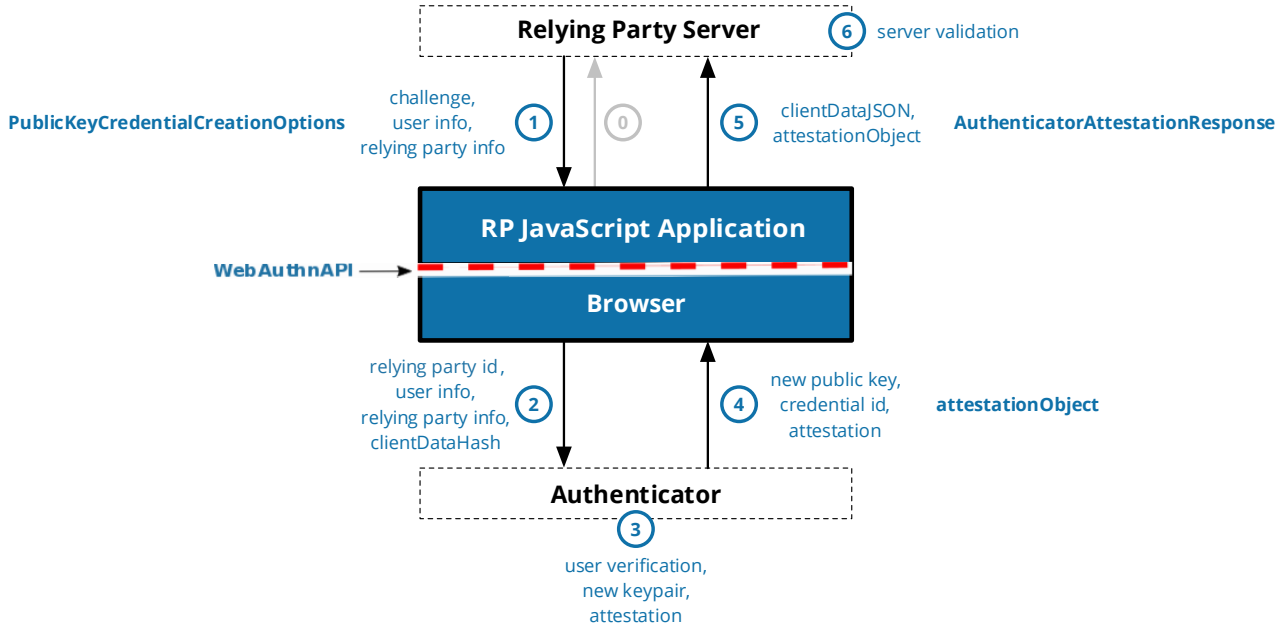
a) Server discoverable credentials case: The FIDO2 SE Authenticator generates a random nonce and a private key is generated using an established key derivation function. The previously generated nonce, the seed, and the identifier sent by the Relying Party are used as input to the key derivation function. The public asymmetric key pair is generated based on the private key using an underlying asymmetric cryptographic algorithm. The FIDO2 SE Authenticator then generates a key handle that subsequently allows it to recover the generated key pair during authentication. The key handle consists of but is not limited to the generated nonce, as well as on a message authentication code (MAC) over the Relying Party identifier and the nonce using the MACKey. The key handle, the identifier, the public key, and the challenge are signed with the attestation private key. The key handle together with the public key, the Attestation Certificate, and the signature are then transmitted to and stored by the Relying Party which also verifies the data and any other possible extension (e.g. HMAC). Optionally, a signature counter run by the FIDO2 SE Authenticator may be incremented and sent to the Relying Party.

b) Client discoverable credentials case: An asymmetric key pair is randomly generated. Then, after requiring the PIN, the FIDO2 SE Authenticator generates a key handle and associates it with the previously generated key pair. The private key is associated with the key handle and the RP ID, and it is stored securely in the authenticator. Subsequently, the key handle allows the FIDO2 SE Authenticator to recover the generated key pair during authentication. The key handle, the Relying Party identifier, the public key, and the challenge are signed with the attestation private key. The key handle together with the public key, the Attestation Certificate, and the signature are then transmitted to and stored by the Relying Party which also verifies the data and any other possible extension. Optionally, a signature counter run by the FIDO2 SE Authenticator may be incremented and sent to the Relying Party.

[4] PIN verification is mandatory if the PIN is configured; it is not required if the PIN has not been configured yet.

[5] Even though the RP can specify the details of its RP ID, the RP ID sent by the client to the authenticator is generated by the client from the information of the TLS trusted channel (the RP cannot impersonate any other RP, the identity is always bound to the TLS trusted channel).

Figure 2-4: Registration Flow (from FIDO Alliance)



In a nutshell, registration consists of the following steps:

- The user visits a website supporting the WebAuthn API with their web browser and clicks on “Sign up”.
- The user enters a username or email address and clicks the “Sign up” button.
- A physical user presence check is performed by the FIDO2 SE Authenticator and optionally PIN is required for user verification.
- If the check(s) are successful, the website notifies the user that they were successfully registered.

2.2.2.2 Authentication

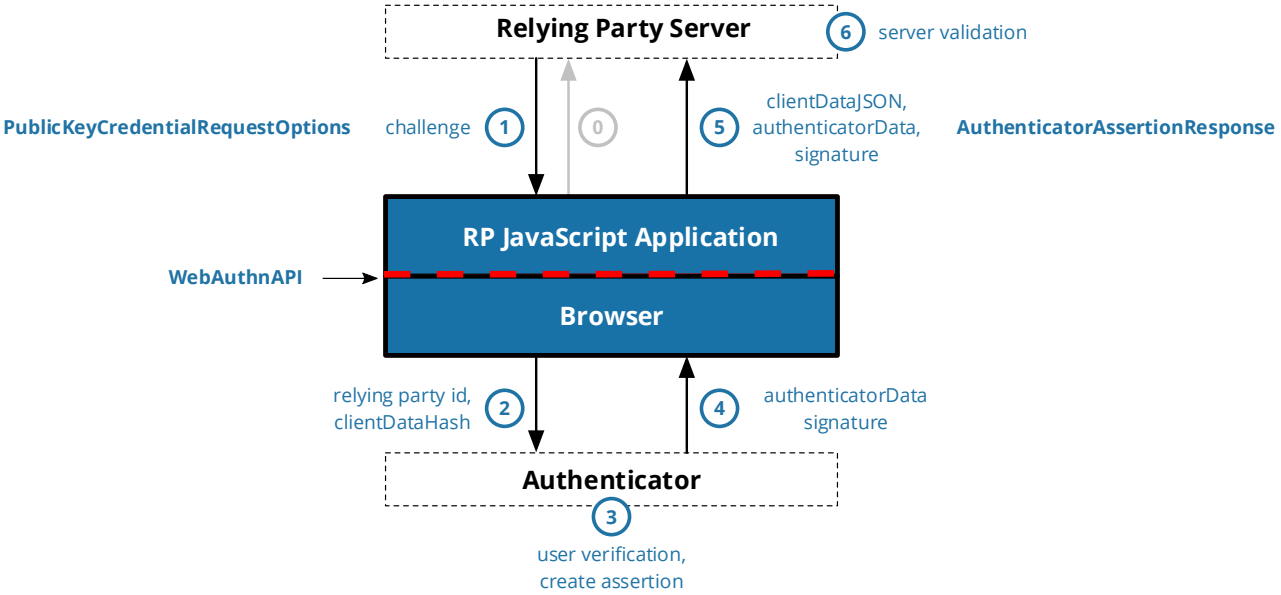
Authentication is the process during which a user authenticates on a previously registered website/service. The interactions happening during authentication between the FIDO2 SE Authenticator, the browser running on the user device, and the Relying Party are shown in Figure 2-5.

During authentication the user may authenticate without a password on a previously registered website/service using the FIDO2 SE Authenticator. PIN can be used in both scenarios a) and b) as an option, depending on whether the RP requests user verification.

a) If the first process described for registration took place, then the FIDO2 SE Authenticator verifies the received message authentication code, thus verifying whether the supplied key handle contains a nonce that corresponds to the Relying Party supplied identifier. After successful verification, the key derivation function is activated and the seed, the identifier, and the nonce are used to generate again the private key. Then, the challenge as well as the Relying Party identifier are signed with the private key, and the challenge as well as the signature are sent to the Relying Party. After successful verification of the signature with the public key of the FIDO2 SE Authenticator (stored during registration by the Relying Party), the Relying Party can conclude that the user authenticated successfully to the website/service.

b) If the second process described during registration took place, then the PIN is requested. Subsequently, the FIDO2 SE Authenticator receives the Relying Party identifier, the previously registered key handle, and a challenge. The FIDO2 SE Authenticator checks whether the key handle is associated with a stored key pair and selects it accordingly. Then the challenge and the identifier are signed with the private key and the signature along with the challenge are sent to the Relying Party. After a successful verification of the signature with the public key of the FIDO2 SE Authenticator (previously stored during the registration on the Relying Party side), the Relying Party can conclude that the user successfully authenticated to the website/service.

Figure 2-5: Authentication Flow (from FIDO Alliance)



In a nutshell, authentication consists of the following steps:

- The user visits a website supporting the WebAuthn API with their web browser and clicks on “Sign in”.
- The user enters a username or email address and clicks the “Sign in” button.
- The web browser asks the user to authenticate using their authenticator.
- If authentication was successful, the website notifies the user that they were successfully signed in.

2.2.2.3 PIN Management

For PIN management, FIDO2 SE Authenticators interact with a FIDO2 client platform to perform one of the following operations:

- PIN configuration
- PIN usage
- PIN renewal

To set a new PIN, the user first sends a *chosen user PIN* to the platform. The client platform collects the chosen user PIN, normalizes it, and then obtains the shared secret from the FIDO2 SE Authenticator. The platform encrypts the normalized PIN using the shared secret and sends it to the FIDO2 SE Authenticator. The latter decrypts it by using the shared secret, performs some internal verifications and processing, and if these are successful, it sets and stores the result as the new PIN. Additionally, the FIDO2 SE Authenticator stores the length of the currently stored PIN and resets the counter of PIN retries to the initial value.

To change an existing PIN, the user sends both the currently set PIN and the new chosen user PIN. The client platform normalizes the new chosen PIN and obtains the shared secret from the FIDO2 SE Authenticator. Using the shared secret, the platform encrypts the existing PIN and the newly chosen PIN, and it sends both as parameters in a request to the FIDO2 SE Authenticator. When receiving the request, after performing internal checks and processing, the FIDO2 SE Authenticator uses the shared secret to decrypt the existing hashed PIN sent by the platform. It then checks that the result corresponds to the currently stored PIN; if correct, it decrypts the new PIN sent by the platform, performs some internal checks and processing, and stores it as the new stored PIN. Additionally, the FIDO2 SE Authenticator stores the length of the currently stored PIN and resets the counter of PIN retries to the initial value.

During PIN usage, a hashed and encrypted PIN token is used.

2.2.2.4 Reset

The user can opt to reset the FIDO2 SE Authenticator, thus destructing all existing cryptographic material. Subsequently, the initialisation process must be repeated before the next usage and the seed and MACKey must be generated again. During both initialisation and reset, the user presence is checked.

2.3 Major Security Features

The essential security features of the FIDO2 SE Authenticator are:

- **Strong Authentication**: The FIDO2 SE Authenticator authenticates a user to a Relying Party with high cryptographic strength. The protocol also protects against typical attacks during authentication, such as man-in-the-middle or phishing attacks.
- **Unlinkability**: The generated asymmetric key pair is unique for each Relying Party and account. All other information, occurring within the CTAP protocol, that could be potentially used to link two accounts to the same user is protected by the TOE. Thus, linking two accounts (e.g. by using public keys or other protocol information) is impossible, even if these two accounts are with the same Relying Party.
- **Privacy**: The FIDO2 SE Authenticator does not store or associate any personal information with the identity of the user.
- **User Presence (UP)**: The FIDO2 SE Authenticator has a physical test of user presence to ensure that the Relying Party can trust that the authentication process is actively triggered by the users themselves.
- **User Verification (UV)**: Optionally, the FIDO2 SE Authenticator provides a user verification mechanism to authenticate the user, e.g. PIN verification or biometric verification. A user verification mechanism may implicitly implement user presence (e.g. fingerprint authentication).

FIDO2 SE Authenticators rely on the physical protection provided by the IC and the security features provided by the GlobalPlatform SE platform, which must be configured so that all security functions used by the FIDO2 SE Authenticator are enabled, especially:

1. Isolation of FIDO2 Authenticator Application by the Java Card execution environment, so that no other application can read or write the FIDO2 Authenticator Application and its associated memory.
2. Security services for card and application management, which
    a. prevent modification of the card content[6] in a way that undermines the security of the Authenticator

[6] E.g. loading malicious code or restoring old versions of the FIDO2 SE Authenticator which might contain obsolete or buggy functionality.

b. ensure that the SE security configuration is fully under control of the Authenticator Vendor or its delegates such that the security configuration present at commercial shipment cannot be changed except for in-the-field updates that are also fully under control of the Authenticator vendor or its delegates

c. ensure that security characteristics of the Authenticator are not modifiable by anyone other than the Authenticator vendor or its delegates

d. support secure software update, based on an Allowed Data Authentication or Signature Cryptographic Function, that is verify that the software being loaded has not been tampered with and do not update the software if the verification fails.

3. Security services for mutual authentication with off-card entities and to protect the information exchanged between card and off-card entities.
4. Cryptographic key management and operations: symmetric/asymmetric encryption and decryption, signature generation and verification, MACing, Random Number Generation, key generation, key derivation, key agreement, hashing, compliant with the *Allowed Cryptography List* [FCrypto].
5. Optionally CVM management for FIDO2 PIN management.

2.4 TOE Usage

FIDO2 SE Authenticators may be used for a variety of authentication processes such as:

- Log in to a web application
- Log in to a user account
- Log in to a VPN or online service
- Regulate access to services
- Control privileged account access.

All frequent use case scenarios involve the usage of a FIDO2 SE Authenticator for registration or authentication to a website/online service.

2.5 Available Non-TOE Hardware/Software/Firmware

FIDO2 Authenticators require non-TOE client and server components to run, as depicted in Figure 2-2.

2.6 TOE Life Cycle

The overall FIDO2 SE Authenticator life cycle maps to the SE life cycle defined in [GPPPSE] which follows the standard SE phases covering development, manufacturing, packaging, product integration, personalisation, and finally, the end-user phase.

Details regarding the FIDO2 SE Authenticator pertain to the (pre-)personalisation phase and the end-user phase.

The (pre-)personalisation phase is typically where the Attestation Key and Attestation Certificate are generated and added on the FIDO2 SE Authenticator. The Attestation Key serves as a trust anchor for the authenticity of the FIDO2 SE Authenticator to the Relying Party. The Attestation Certificate refers to the public Attestation Key typically signed by a certification authority endorsed by the manufacturer. The initialisation process can be performed at this point. The seed and MACKey may be generated and stored on the FIDO2 SE Authenticator before delivery to the end-user.

If the seed and MACKey have not been generated during the previous phase, they are generated and stored on the FIDO2 SE Authenticator during the first usage of the authenticator by the user in the end-user phase. The user may also choose to reset the FIDO2 SE Authenticator and repeat the initialisation process, thus regenerating the seed and MACKey.

Application Note: The ST author shall describe the actual TOE life cycle and explicitly address the following:

- each of the modes of operation, including Consumer and Enterprise modes, and identify all the reversible phases and all the phases where key import may happen and under which conditions.
- the factory reset operation, including conditions and consequences.

3 CONFORMANCE CLAIMS

3.1 CC Conformance Claim

This PP claims conformance to:

- CC Part 1 [CC1]
- CC Part 2 [CC2] extended
- CC Part 3 [CC3] conformant

3.2 Package Claim

This PP claims conformance to EAL4 augmented with:

- ALC_DVS.2 Sufficiency of security measures
- AVA_VAN.5 Advanced methodical vulnerability analysis

Application Note: The ST author shall consider adding an ALC_FLR Flaw reporting procedures component, which may be required in some schemes.

3.3 Conformance Claim of the PP

This PP does not claim conformance to any other PP.

3.4 Conformance Statement

This PP requires strict conformance of any ST or PP claiming conformance to it.

4 SECURITY PROBLEM DEFINITION

This chapter introduces the security problem addressed by the TOE and its operational environment. The security problem consists of the threats the FIDO2 SE Authenticators may face in the field, the assumptions on their operational environment, and the organisational policies that must be implemented by the FIDO2 SE Authenticator itself or within the operational environment.

4.1 Assets

4.1.1 Attestation Key

The cryptographic key used to attest a cryptographic authentication key and the type/model and manufacturing batch of an authenticator.

Properties: Authenticity, confidentiality, integrity.

Application Note: Attestation Keys are either ECDAA keys or the Attestation Keys and Certificates shared by a large number of authenticators in a device class from a given vendor to prevent their becoming a linkable identifier across relying parties. Authenticator Attestation Certificates are signed by an authority key controlled by the vendor. The vendor may use a per-vendor certificate authority key for this purpose.

4.1.2 Authenticator Identification

Authenticator identification data used to uniquely identify the authenticator across Relying Parties.

Properties: Unique and non-modifiable.

Application Note: The FIDO2 SE Authenticator identification should be returned to the Relying Parties. However, exposing such data may affect the privacy/unlinkability requirements.

4.1.3 Authentication Private Keys

The private keys of the asymmetric key pairs used to associate a user account and a Relying Party. These keys are transient when generated from a seed (case a in section 2.2.2.1) and persistent when stored in the FIDO2 SE Authenticator (case b).

Properties: Integrity, confidentiality, authenticity.

4.1.4 Authenticator Security Parameters (ASP)

Privacy and security-relevant data and user information stored by or used within the FIDO2 SE Authenticator.

Properties: Integrity, monotonicity (for counters), and confidentiality if/where needed.

Application Note: At a minimum ASPs include all configuration data and settings, user verification reference data, user verification tokens, key handle access tokens, signature or registration operation counters, privacy sensitive data, cryptographic keys used for PIN management.

4.1.5 MACKey

The cryptographic key used to generate the MAC; it is generated once during initialisation; the MAC itself is part of the key handle.

Properties: Confidentiality, integrity.

Application Note: To launch the generation of the MACKey, the end user must execute a command during the initialisation process. The MACKey must never be exported outside the TOE. In case of a reset initiated by the

end-user, the MACKey is deleted. A new MACKey is generated if the end-user starts the initialisation process again.

4.1.6 PIN

Data used to perform user verification and to authenticate subsequent commands exchanged between the FIDO2 SE Authenticator and a client platform. It is persistent data stored on the FIDO2 SE Authenticator.

Properties: Integrity, confidentiality.

Application Note: The PIN is a randomly generated, opaque byte-string that is large enough to be effectively unguessable (here the PIN is the PIN token not the plaintext user-chosen PIN; it ensures that the plaintext PIN is not sent to the FIDO2 SE Authenticator).

4.1.7 PIN Management Data

PIN management-relevant data, settings, and counters such as the PIN retry counter and PIN length.

Properties: Integrity.

4.1.8 RNG

The Random Number Generator.

Properties: Unpredictable random numbers, sufficient entropy.

Application Note: The RNG is used to generate the seed, the MACKey, and the private keys that are not derived from a seed (case b in section 2.2.2.1).

4.1.9 Seed

The seed used to compute the authentication private keys; it is generated once during initialisation.

Properties: Confidentiality, integrity.

Application Note: To launch the generation of the seed, the end-user must execute a command during the initialisation process. The seed must never be exported outside the FIDO2 SE Authenticator. If the end-user initiates a reset, the seed is deleted. A new seed is generated if the end-user starts the initialisation process again.

4.2 Users

This PP considers the following external entities (users).

4.2.1 Authenticator Specific Module (ASM)

The software associated with an authenticator which provides a uniform interface between the authenticator and the client platform. It connects the authenticator to the Relying Party.

4.2.2 Client/Client platform

The client or client platform consists of a client device, i.e. a hardware device such as a laptop or desktop computer, the operating system running on that hardware, and the client running on it as specified in [Web Auth].

4.2.3 Relying Party

The entity relying on the CTAP protocol to directly authenticate users (i.e. to perform peer-entity authentication).

4.2.4 User

The legitimate owner and end-user of the FIDO2 SE Authenticator. The human user relies on the authenticator for authentication to the Relying Parties.

4.3 Threats

4.3.1 T.ABUSE_DEBUG

An attacker abuses a hardware or a software debugging interface of the FIDO2 SE Authenticator, e.g. for a debug interface that is not completely disabled or through any means that allow reenabling a debug interface. This may allow obtaining sensitive data or compromising security functionality (bypass, deactivate, or modify security services).

Assets threatened directly: All.

Application Note: Threats consisting of abusing a hardware debugging interface are covered by the IC.

4.3.2 T.ABUSE_FUNCTIONALITY

An attacker abuses the authenticator's functionality by accessing it outside of the expected range or by sending invalid commands or commands with invalid parameters.

Assets threatened directly: Authentication Private Keys, MACKey, PIN, RNG, Seed.

Application Note:

- A classic example of this threat is the injection of a malformed input to trigger an error during input processing.
- A successful attack of this kind directly impacts all assets, but the result of malformed input injection is poorly controllable in the absence of strong internal segmentation of the FIDO2 SE Authenticator.

4.3.3 T.ASP_LEAK

An attacker gains access to confidential security-relevant data by bypassing the authenticator's security functionality and accessing the stored ASP data.

Assets threatened directly: ASPs.

4.3.4 T.ASP_MODIFICATION

An attacker modifies or substitutes all or part of the ASP. An attacker may reset ASPs, rollback to previous value, or modify the values of ASPs, e.g. counters such as the signature counter.

Assets threatened directly: ASPs.

4.3.5 T.BAD_KEY_OR_SIGNATURE_GENERATION

An attacker succeeds in tampering with the generation of a key or signature.

Assets threatened directly: Authentication Private Keys, MACKey, Seed.

Application Note: Injecting an error during the key or signature generation process may lead to this kind of threat.

4.3.6 T.CLONE

An attacker clones data from a FIDO2 SE Authenticator and uses it on a different authenticator for login at the Relying Party as the legitimate user.

Assets threatened directly: Attestation Key, Authentication Private Keys, MACKey, Seed.

4.3.7 (Optional) T.IMPERSONATION

An attacker modifies intercepted communications or injects pre-captured or leaked user verification data in the FIDO2 SE Authenticator to impersonate the legitimate user and login to the Relying Party.

An attacker succeeds in performing a brute-force attack to the user verification method.

Assets threatened directly: Authentication Private Keys, MACKey, PIN, Seed.

4.3.8 T.KEY_LEAK

An attacker succeeds in extracting cryptographic keys, i.e. the MACKey, the authentication private authentication keys, the Attestation Key, or the seed. By compromising such keys, an attacker can violate all security goals of FIDO. For instance, the cryptographic keys may be used in a different context, allowing an attacker to successfully impersonate the legitimate user using a cloned authenticator and gaining unauthorized access to the Relying Party.

Assets threatened directly: Attestation Key, Authentication Private Keys, MACKey, Seed.

Application Note: The private keys may be leaked (partially or entirely) by different means, such as:

- Injecting an error in the key or signature generation process (differential fault analysis).
- Using a timing attack from a remote location.
- Exploiting information leaking from an authenticator during its usage, potentially through a side-channel attack (e.g. by measuring the power consumption using differential power/electromagnetic analysis during operational use). Side-channel attacks may be local or internal. A local side-channel attack requires physical access to an authenticator, whereas an internal side-channel (or a covert channel) attack requires control over a process running on the same authenticator.
- Physical probing to disclose or to reverse engineer the sensitive cryptographic material. It requires physical access to an authenticator.

4.3.9 T.KEY_MODIFICATION

An attacker succeeds in modifying a cryptographic key or in substituting it with another value.

Assets threatened directly: Attestation Key, Authentication Private Keys, MACKey.

Application Note: Some encryption modes allow an attacker to target bit-level changes to the plaintext.

4.3.10 T.PIN_DATA_LEAK

An attacker gains access to data that is part of or relevant for the PIN used for user verification either by intercepting it during exchanges with the client platform or by bypassing the authenticator's security functionality and accessing the stored PIN.

Assets threatened directly: PIN, PIN Management Data.

4.3.11 T.PIN_DATA_MODIFICATION

An attacker modifies or substitutes the PIN used for user verification or any other data or information used for PIN management. An attacker may reset, rollback to a previous value, or modify the value of a counter such as the PIN retry counter.

Assets threatened directly: PIN, PIN Management Data.

4.3.12 T.PRIVACY_VIOLATION

An attacker traces user information by information exchanged between the FIDO2 SE Authenticator and one or more Relying Parties.

Note that a small set of FIDO2 Authenticators that share an Attestation Key leaks information about the user across Relying Parties.

Assets threatened directly: Attestation Key, Authenticator Identification.

4.3.13 T.RANDOM_NUMBER_PREDICTION

An attacker gets access to information allowing the prediction of RNG data. This may occur, for instance, because the generated random numbers have insufficient entropy, or because the attacker forces the output of a partially or totally predefined value. A set of randomly generated parameters colliding could serve as a preliminary step.

Loss of unpredictability (the main property of random numbers) is a problem in case they are used to generate cryptographic keys. Malfunctions or premature ageing may also allow getting information about random numbers.

Assets threatened directly: Authentication Private Keys, RNG.

Application Note: A variation of this threat could amount to compromising the entropy source or the internal PRNG state at different stages, namely:

- Prior to the generation of the seed and the MACKey during initialisation; in this case the attacker can know and specify all future entropy inputs to the PRNG;
- After the generation of the seed and the MACKey during initialisation; in this case, such an attack may be a preliminary step towards other attacks, such as a forgery attack;
- Prior to the generation of the seed and the MACKey during initialisation but with subsequent loss of control and access to the entropy source.

4.3.14 T.SIGNATURE_ALGORITHM_COMPROMISE

A cryptographic attack is discovered and mounted against the public key cryptography system used to sign data.

Assets threatened directly: Authentication Private Keys, MACKey, Seed.

4.3.15 T.USER_PRESENCE_BYPASS

An attacker succeeds in bypassing the user presence check and thus using the authenticator without physical access; abusing the user presence caching; or by compromising, modifying, or deactivating the authenticator's security functionality.

Assets threatened directly: All.

4.3.16 (Optional) T.USER_VERIFICATION_BYPASS

An attacker succeeds in bypassing the user verification process and thus using the cryptographic keys without user verification; abusing the user verification caching; or by compromising, modifying, or deactivating the authenticator's security functionality.

Assets threatened directly: MACKey, PIN, Seed.

Application Note:

- User verification caching is an option allowing user verification to be skipped if the FIDO2 SE Authenticator has verified the user within a specified time frame. To mount such an attack, a large timeframe could be specified/modified so that the user is not freshly verified.
- To alter the authenticator's security functionality, techniques commonly employed in IC failure analysis and reverse engineering may be used after having identified the hardware security mechanisms and layout characteristics. Modifications may lead to the deactivation of a security function, for instance, or an unauthorized usage of a legitimate functionality of an authenticator. For instance, by bypassing user verification, an attacker may trigger an unauthorized reinitialisation of the authenticator, thus leading to a new generation of the seed and the MACKey and invalidating/deleting the previous ones.

4.4 Organisational Security Policies (OSP)

4.4.1 OSP.ATTESTABLE_PROPERTIES

The TOE shall allow Relying Parties to verify the FIDO2 SE Authenticator's model/type to calculate the associated risk.

4.4.2 OSP.CRED_PROTECTION_EXTENSION

The TOE shall implement the CredProtection extension. The TOE shall reveal information only under the conditions stipulated in the CTAP specification [CTAP] for the implemented CredProtection Level.

4.4.3 (For Consumer) OSP.DISABLED_ENTERPRISE_ATTESTATION

In the Consumer configuration, any firmware supporting Enterprise Attestation shall be disabled before entering the end-user phase for FIDO2 SE Authenticators.

4.4.4 (For Enterprise) OSP.ENTERPRISE_ATTESTATION

A TOE that supports Enterprise Attestation and inserts a unique identifier in its Enterprise Attestation Certificate shall use a unique private key per identifier.

A TOE shall return an Enterprise Attestation only for RP identifiers included in the configured list of RP identifiers.

4.4.5 (For Enterprise) OSP.ENTERPRISE_ATTESTATION_PROVISIONING_RPLIST

Only the Vendor or its delegates shall be able to configure the TOE's Enterprise Attestation Certificate and the list of RP identifiers (if such list is supported).

4.4.6 (Optional) OSP.FACTORY_RESET

A TOE may implement factory reset functionality, and in that case, the TOE shall ensure that original (factory) state is reached, i.e. deleting all user specific information.

4.4.7 OSP.LIMITED_PII

The amount of personal identifiable information exposed by the TOE to a Relying Party shall be limited to the absolute minimum.

Depending on whether Enterprise Attestation is supported by the TOE, one of the following holds true:

A) For FIDO2 SE Authenticators that support Enterprise Attestation, the only information that is visible across multiple RPs is the unique identifier present in the Enterprise Attestation Certificate or the Enterprise Attestation Certificate itself.

B) For FIDO2 SE Authenticators that do not support Enterprise Attestation, if the same exact Attestation Key is put into a group of FIDO2 SE Authenticators, then the group of FIDO2 SE Authenticators must be at least 100.000 in number. If less than 100.000 FIDO2 SE Authenticators are made, they all must have the exact same Attestation Key. Key Handles or Key identifiers shall not be visible across multiple RPs.

4.4.8 OSP.USER_CONSENT

The FIDO2 SE Authenticator shall require explicit user consent before a relationship to a new Relying Party is established (rejecting the request if no proof can be established).

4.5 Assumptions

4.5.1 A.ASSURANCE_LEVEL_LIFE

Users and Relying Parties consider the fact that the authenticator's assurance level is linked to the confidence in the cryptographic keys which is a commodity that decays overs time, irrespective of any compromising event.

4.5.2 A.AUTHENTICATOR_CHECK

A Relying Party allows registration of reliable authenticators that have passed Authenticator Certification only.

4.5.3 A.PROTECTION_AFTER_DELIVERY

The FIDO2 SE Authenticator, any ASP and cryptographic keys are protected by the operational environment after delivery and before entering the end-user phase. The persons using the FIDO2 SE Authenticator in the operational environment have the required skills to understand and apply the security guidelines.

4.5.4 A.SECURE_VERIFIER_DATABASE

The Relying Party stores only the public portion of an asymmetric key pair, or an encrypted key handle, as a cryptographic authentication key reference.

4.5.5 A.TRUSTWORTHY_CE

The computing environment on the FIDO client platform and the client applications involved in a FIDO2 operation act as trustworthy agents of the user. That is, the client acts in good faith, at least for:

- sending the correct RP ID and verifying the TLS session,
- managing the user presence in NFC by asking the user to tap the device on the reader or by physically connecting the device to the client,
- not storing the user PIN improperly.

4.5.6 A.TRUSTWORTHY_RP

The computing resources at the Relying Party involved in processing a FIDO2 operation act as trustworthy agents of the Relying Party.

4.5.7 (For Enterprise) A.TRUSTWORTHY_RP_IDENTIFIERS_LIST

For FIDO2 SE Authenticators that support Enterprise Attestation the configured RP identifiers list contains only valid RP identifiers.

4.5.8 A.TRUSTWORTHY_SERVER_AUTHENTICATION

Any application on the user device establishes secure channels that provide trustworthy server authentication, and confidentiality and integrity for messages.

4.5.9 A.USER_DEVICE_SEPARATION_MECHANISM

The privilege separation mechanisms of the operating system relied upon by the software modules involved in a FIDO2 operation on the user device perform as specified (e.g. boundaries between user and kernel mode, between user accounts, and between applications are securely enforced and security principles can be mutually, securely identifiable, etc.).

4.5.10 (For Enterprise) A.ENTERPRISE_USERS

FIDO2 SE Authenticators that support Enterprise Attestation are sold only to an Enterprise for use by the Enterprise's employees, contractors or defined members. It is not used with the customers of the Enterprise. It is not sold to end-users on the open market.

5 SECURITY OBJECTIVES

5.1 Security Objectives for the TOE

5.1.1 O.ALLOWED_CRYPTOGRAPHY

The TOE ensures that a cryptographic protocol from the allowed cryptography list [FCrypto] is employed and that all cryptography-related functionality is up to date.

Application Note: The cryptographic algorithms and parameters (key size, mode, output length, etc.) included in [FCrypto] and used by the TOE are not subject to known weaknesses that make them unfit for their purpose in encrypting, digitally signing, and authenticating messages.

5.1.2 O.ASP_PROTECTION

The TOE shall protect ASPs from unauthorized modification and substitution; in particular, the TOE shall ensure that counters such as the signature counter are either monotonically increasing or monotonically decreasing and protected from reset, value modification (decrease or increase, respectively), and rollback to a previous value.

The TOE shall protect confidential ASPs from unauthorized disclosure.

Moreover,

1. Any ASP stored outside the TOE shall be protected against modification and replay using a Data Authentication, Signature, or Key Protection Cryptographic Function included in [FCrypto].
2. Any confidential ASP stored outside the TOE shall be protected using a Key Protection Algorithm included in [FCrypto].
3. Any key used with a Key Protection Cryptographic Function included in [FCrypto] to protect an ASP stored in the operating environment shall have a claimed cryptographic strength that is equal to or greater than the claimed cryptographic strength of the key being wrapped.

5.1.3 O.ATTESTABLE_PROPERTIES

The TOE shall allow Relying Parties to verify the FIDO2 SE Authenticator model/type.

5.1.4 O.CORRECT_KEY_AND_SIGNATURE_GENERATION

The TOE shall ensure that cryptographic keys and signatures are correctly generated and that an algorithm from the allowed cryptography list [FCrypto] is employed to this end.

The TOE shall ensure that signatures can only be performed by using keys that were generated by the Authenticator for that purpose.

Application Note: This objective applies to any key generation resulting in an ASP or a cryptographic key used for PIN management and to any random input used for signature generation.

5.1.5 O.DISABLED_DEBUG

The TOE shall ensure that no sensitive information of the FIDO2 SE Authenticator is shared through physical or logical debug interfaces in the end-user phase.

5.1.6 (For Enterprise) O.ENTERPRISE_ATTESTATION

The TOE shall return an Enterprise Attestation only for RP identifiers included in the configured list of RP identifiers.

5.1.7 (For Enterprise) O.ENTERPRISE_ATTESTATION_PROVISIONING

The TOE shall only allow the Vendor or its delegate to configure the TOE's Enterprise Attestation Certificate and the list of RP identifiers if such list is supported.

5.1.8 (Optional) O.FACTORY_RESET

A TOE may implement factory reset functionality, and in that case, the TOE shall ensure that original (factory) state is reached, i.e. deleting all user specific information.

5.1.9 O.FUNCTIONALITY_PROTECTION

The TOE shall protect against abuse of functions which may not be used in the operational phase, and which may lead to manipulation or disclosure of user data or TSF-data or to the abuse of the TOE's security functionality (i.e. bypassing, deactivation, or modification).

The TOE shall protect against abuse of the user presence caching and user verification caching.

The TOE shall protect all direct interactions of the user with the Authenticator, especially for the purpose of user presence and user verification.

The TOE shall ensure the correct operation of its security functions. It shall:

- Protect itself against abnormal situations caused by errors or malformed inputs;
- Enter a secure state upon failure detection, without exposure of any sensitive data.
- Allow only the Vendor (or its delegates) to modify the TOE's security characteristics or configure Enterprise Attestation (i.e. change the configuration mode).

5.1.10 O.IMPERSONATION_RESILIENCE

The TOE shall protect against impersonation of the legitimate user by exploiting information leaked by a Relying Party or an Authenticator. In particular, the TOE shall be resistant to cloning and replay attacks, i.e. information from a particular FIDO2 SE Authenticator cannot be replayed or used by another FIDO2 SE Authenticator. The TOE shall send a monotonically increasing or decreasing signature counter to the Relying Party.

5.1.11 O.LEAKAGE_RESISTANCE

The TOE shall provide protection against disclosure and extraction of confidential data; e.g. the Attestation Key, the MACKey, the Authentication Private Keys, the Seed, and any confidential ASP by information leakage during the operation of the TOE.

5.1.12 O.PIN_DATA_PROTECTION

The TOE shall ensure that the user's PIN is protected against modification, injection, and disclosure or extraction when stored or in transit (i.e. when transmitted to/from the client platform). The TOE shall ensure that any PIN management-relevant data is protected against modification and injection. In particular, the TOE shall ensure that the PIN retry counter is monotonic and protected from reset, value modification, and rollback to a previous value.

Application Note: The PIN retry counter may be monotonically increasing or monotonically decreasing.

Application Note: This objective is a specialization of O.ASP_PROTECTION to the PIN and PIN-related data.

5.1.13 O.RNG

The TOE shall ensure the quality of random number generation. Random numbers shall not be predictable and shall have sufficient entropy.

5.1.14 (Optional) (For Enterprise) O.RP_IDENTIFIERS_LIST_MANAGEMENT

For FIDO2 SE Authenticators supporting Enterprise Attestation and being configured with a list of valid RP identifiers, this configured RP identifiers list shall be modifiable only by the Vendor (not the user).

5.1.15 O.TAMPER_RESISTANCE

The TOE shall protect all assets from physical tampering leading to information modification or extraction.

5.1.16 O.TRUSTWORTHY_DATA

The TOE accepts and processes sensitive data only if these were generated by itself and are linkable to the Relying Party with the identifier supplied during registration.

5.1.17 O.UNLINKABLE_LIMITED_PII

The TOE ensures that the amount of personal identifiable information exposed to the Relying Party is limited to the absolute necessary minimum. More precisely,

1. Such information cannot be used to uniquely identify the FIDO2 SE Authenticator instance to a different Relying Party;
2. No two Relying Parties can link a separate conversation to one user, i.e. the conversation is unlinkable;
3. No Relying Party can determine whether the same FIDO2 SE Authenticator is used by two different user accounts; i.e. possessing two tuples of authentication information produced by the same FIDO2 SE Authenticator does not allow establishing that both tuples were produced by the same FIDO2 SE Authenticator;
4. The information revealed to third parties is controlled as stipulated in the CTAP specification [CTAP] for the implemented CredProtection Level.

5.1.18 O.USER_CONSENT

The TOE shall require explicit user consent before a relationship to a new Relying Party is established (rejecting the request if no proof can be established).

5.1.19 O.USER_PRESENCE_CHECK

The TOE shall provide user presence check functionality. The TOE shall ensure that user presence is checked for any user-invoked process for which the CTAP specification [CTAP] requires a prior demonstration of user presence.

Application Note: The corresponding option or flag indicating that the FIDO2 SE Authenticator performs user presence check shall be set accordingly.

5.1.20 (Optional) O.USER_VERIFICATION

The TOE shall support user verification with or without implicit user presence check. The TOE shall ensure that user verification is performed for any user-invoked process for which the CTAP specification [CTAP] requires a prior verification of the user.

The TOE shall limit the number of unsuccessful user verification attempts.

Application Note: If user verification implicitly performs user presence check, then the corresponding option or flag indicating that the FIDO2 SE Authenticator performs user presence check shall explicitly indicate this.

5.2 Security Objectives for the TOE Operational Environment

5.2.1 OE.ASSURANCE_LEVEL_LIFE

In the end-user phase, a possibly reduced assurance level of the authenticators shall be taken into consideration over time. Users and Relying Parties shall consider that the authenticator's assurance level is linked to the confidence in the cryptographic keys which is a commodity that decays overs time, irrespective of any compromising event.

5.2.2 OE.AUTHENTICATOR_CHECK

A Relying Party shall use Authenticator Class Attestation to identify and allow registration only for reliable authenticators that have passed Authenticator Certification.

Application note: The FIDO Metadata Service includes the Authenticator Certification status and sends regular updates of the trust store to the Relying Parties. Additionally, the Relying Parties may check if an attestation presented by a malicious authenticator has been marked as compromised.

5.2.3 OE.PROTECTION_AFTER_DELIVERY

The FIDO2 SE Authenticator, any ASP and cryptographic keys shall be protected by the operational environment after delivery and before entering the end-user phase. The persons using the FIDO2 SE Authenticator in the operational environment have the required skills to understand and apply the security guidelines.

5.2.4 OE.DISABLED_DEBUG

All physical and logical debug interfaces shall be disabled before entering the end-user phase.

5.2.5 (For Consumer) OE.DISABLED_ENTERPRISE_ATTESTATION

In the Consumer configuration, any firmware supporting Enterprise Attestation shall be disabled before entering the end-user phase for FIDO2 SE Authenticators.

Application Note: This objective applies if the operation is performed in the operational environment before entering the end-user phase. For the manufacturing case, a refinement of ALC_LCD.1 has been added.

5.2.6 (For Enterprise) OE.ENTERPRISE_ATTESTATION

A unique private key per identifier shall be used by any FIDO2 SE Authenticator that supports Enterprise Attestation and that inserts a unique identifier in its Enterprise Attestation Certificate.

Application Note: This objective applies if the operation is performed in the operational environment before entering the end-user phase. For the manufacturing case, a refinement of ALC_LCD.1 has been added.

5.2.7 OE.LIMITED_PII

Depending on whether Enterprise Attestation is supported by the TOE, one of the following holds true:

A) (For Enterprise) The only information that is visible across multiple RPs is the unique identifier present in the Enterprise Attestation Certificate or the Enterprise Attestation Certificate itself.

B) (For Consumer) If the same exact Attestation Key is put into a group of FIDO2 SE Authenticators, then the group of FIDO2 SE Authenticators must be at least 100.000 in number. If less than 100.000 FIDO2 SE Authenticators are made, then they all must have the exact same Attestation Key.

Application Note: This objective applies if the operation is performed in the operational environment before entering the end-user phase. For the manufacturing case, a refinement of ALC_LCD.1 has been added.

5.2.8 OE. SECURE_VERIFIER_DATABASE

The Relying Party shall store only the public portion of an asymmetric key pair, or an encrypted key handle, as a cryptographic authentication key reference.

5.2.9 OE.TRUSTWORTHY_CE

The computing environment on the FIDO client platform and the client applications shall ensure that:

- the correct RP ID is sent, and the TLS session is verified,
- the user presence in NFC is managed by asking the user to tap the device on the reader,
- the user PIN is not stored.

5.2.10 OE.TRUSTWORTHY_RP

The computing resources at the Relying Party involved in processing a FIDO2 operation shall act as trustworthy agents of the Relying Party.

5.2.11 (For Enterprise) OE.TRUSTWORTHY_RP_IDENTIFIERS_LIST

For FIDO2 SE Authenticators that support Enterprise Attestation the list of configured RP identifiers (if supported) shall contain only valid RP identifiers.

Application Note: The process of RPIDs configuration requires the check by the Vendor that the provided list of RPIDs is owned by the Customer or the Customer's Data Processors as defined by the GPDR.

5.2.12 OE.TRUSTWORTHY_SERVER_AUTHENTICATION

Any application on the user device shall establish secure channels that provide trustworthy server authentication, and confidentiality and integrity for messages.

5.2.13 OE.USER_DEVICE_SEPARATION_MECHANISM

The operating system shall enforce privilege separation mechanisms (e.g. boundaries between user and kernel mode, between user accounts, and between applications are securely enforced and security principles can be mutually, securely identifiable, etc.). The registrations and key material as a shared system resource are appropriately protected according to the operating environment privilege boundaries set in place on the FIDO2 user device.

5.2.14 (For Enterprise) OE.ENTERPRISE_USERS

FIDO2 SE Authenticators that support Enterprise Attestation shall be sold only to an Enterprise for use by the Enterprise's employees, contractors or defined members. It shall not be used with the customers of the Enterprise. It shall not be sold to end-users on the open market.

5.3 Security Objectives Rationale

5.3.1 Threats

This section presents the rationale of coverage of threats by objectives for the TOE and the TOE operational environment. Table 5-1 provides the mapping between these elements.

T.ABUSE_DEBUG: The threat is covered by the conjunction of the following objectives:

- O.DISABLED_DEBUG, which ensures that sensitive data cannot be obtained by any debug interface in the end-user phase, and
- OE.DISABLED_DEBUG, which ensures that there is no debug interface enabled in the end-user phase.

T.ABUSE_FUNCTIONALITY: The threat is directly covered by the objective O.FUNCTIONALITY_PROTECTION.

T.ASP_LEAK: The threat is covered by the conjunction of the following objectives:

- O.ASP_PROTECTION, which ensures that the TOE does not disclose confidential ASP and that confidential ASP that is stored outside the TOE is cryptographically protected,
- O.LEAKAGE_RESISTANCE, which ensures that the TOE protects against information leakage,
- O.TAMPER_RESISTANCE, which ensures that the TOE protects data extraction by physical tampering.

T.ASP_MODIFICATION: The threat is directly covered by the objective O.ASP_PROTECTION.

T.BAD_KEY_OR_SIGNATURE_GENERATION: The threat is covered by the conjunction of the following objectives:

- O.ALLOWED_CRYPTOGRAPHY, which ensures that only allowed cryptography is used to generate keys and signatures,
- O.CORRECT_KEY_AND_SIGNATURE_GENERATION, which ensures the proper generation of keys and signatures,
- O.RNG, which ensures that the random numbers have sufficient entropy.

(Optional) T.IMPERSONATION: The threat is covered by the conjunction of the following objectives:

- O.IMPERSONATION_RESILIENCE, which ensures that leaked information cannot be used to impersonate the user,
- O.TRUSTWORTHY_DATA, which ensures that the TOE only relies on data generated inside its boundary.

T.KEY_LEAK: The threat is covered by the conjunction of the following objectives:

- O.ASP_PROTECTION, which ensures that confidential ASP, including keys, is protected from disclosure,
- O.LEAKAGE_RESISTANCE, which ensures that emanations do not lead to the disclosure of keys,
- O.TAMPER_RESISTANCE, which ensures that data, including keys, are protected from disclosure by physical tampering.

T.KEY_MODIFICATION: The threat is covered by the conjunction of the following objectives:

- O.ASP_PROTECTION, which ensures that confidential ASP, including keys, is protected from modification,
- O.TAMPER_RESISTANCE, which ensures that data, including keys, are protected from modification by physical tampering.

T.PIN_DATA_LEAK: The threat is directly covered by the objective O.PIN_DATA_PROTECTION.

T.PIN_DATA_MODIFICATION: The threat is directly covered by the objective O.PIN_DATA_PROTECTION.

T.PRIVACY_VIOLATION: The threat is covered by the conjunction of the following objectives:

- O.UNLINKABLE_LIMITED_PII, which ensures that a minimum of personal information is exposed to Relying Parties,
- OE.LIMITED_PII, which puts conditions to limit the information that is directly or indirectly shared, for Enterprise and for Consumer authenticators.

T.RANDOM_NUMBER_PREDICTION: The threat is directly covered by the objective O.RNG.

T.SIGNATURE_ALGORITHM_COMPROMISE: The threat is directly covered by the objective O.ALLOWED_CRYPTOGRAPHY.

T.USER_PRESENCE_BYPASS: The threat is covered by the conjunction of the following objectives:

- O.FUNCTIONALITY_PROTECTION, which directly covers the threat,
- O.TAMPER_RESISTANCE, which ensures protection against physical tampering, including tampering leading to the bypassing of the user presence check,
- O.USER_PRESENCE_CHECK, which ensures that the user presence is checked as required by the CTAP specification.

(Optional) T.USER_VERIFICATION_BYPASS: The threat is covered by the conjunction of the following objectives:

- O.FUNCTIONALITY_PROTECTION, which directly covers the threat,

- O.TAMPER_RESISTANCE, which ensures protection against physical tampering, including tampering leading to the bypassing of the user verification,
- O.USER_VERIFICATION, which ensures that the user verification is performed as required by the CTAP specification.

Table 5-1: Mapping threats and security objectives

T.ABUSE_DEBUG	T.ABUSE_FUNCTIONALITY	T.ASP_LEAK	T.ASP_MODIFICATION	T.BAD_KEY_OR_SIGNATURE_GENERATION	T.CLONE	T.IMPERSONATION (Optional)	T.KEY_LEAK	T.KEY_MODIFICATION	T.PIN_DATA_LEAK	T.PIN_DATA_MODIFICATION	T.PRIVACY_VIOLATION	T.RANDOM_NUMBER_PREDICTION	T.SIGNATURE_ALGORITHM_COMPROMISE	T.USER_PRESENCE_BYPASS	T.USER_VERIFICATION_BYPASS (Optional)	**Objectives**
				X									X			O.ALLOWED_CRYPTOGRAPHY
		X	X				X	X								O.ASP_PROTECTION
																O.ATTESTABLE_PROPERTIES
				X												O.CORRECT_KEY_AND_SIGNATURE_GENERATION
X																O.DISABLED_DEBUG
																O.ENTERPRISE_ATTESTATION_PROVISIONING (for Enterprise)
																O.ENTERPRISE_ATTESTATION (for Enterprise)
																O.FACTORY_RESET (Optional)
	X													X	X	O.FUNCTIONALITY_PROTECTION
					X	X										O.IMPERSONATION_RESILIENCE
		X			X		X									O.LEAKAGE_RESISTANCE
									X	X						O.PIN_DATA_PROTECTION
				X								X				O.RNG
																O.RP_IDENTIFIERS_LIST_MANAGEMENT
		X			X		X	X						X	X	O.TAMPER_RESISTANCE
						X										O.TRUSTWORTHY_DATA
											X					O.UNLINKABLE_LIMITED_PII
																O.USER_CONSENT
														X		O.USER_PRESENCE_CHECK
															X	O.USER_VERIFICATION (Optional)
																OE.ASSURANCE_LEVEL_LIFE
																OE.AUTHENTICATOR_CHECK
																OE.PROTECTION_AFTER_DELIVERY
X																OE.DISABLED_DEBUG
																OE.DISABLED_ENTERPRISE_ATTESTATION (for Consumer)
																OE.ENTERPRISE_ATTESTATION
											X					OE.LIMITED_PII
																OE.SECURE_VERIFIER_DATABASE
																OE.TRUSTWORTHY_CE
																OE.TRUSTWORTHY_RP
																OE.TRUSTWORTHY_RP_IDENTIFIERS_LIST (for Enterprise)
																OE.TRUSTWORTHY_SERVER_AUTHENTICATION
																OE.USER_DEVICE_SEPARATION_MECHANISM
																OE.ENTERPRISE_USERS (for Enterprise)

5.3.2 Organisational Security Policies

This section presents the rationale of coverage of OSPs by objectives for the TOE and the TOE operational environment. Table 5-2 provides the mapping between these elements.

OSP.ATTESTABLE_PROPERTIES: The OSP is directly covered by the objective O.ATTESTABLE_PROPERTIES.

OSP.CRED_PROTECTION_EXTENSION: The OSP is directly covered by the objective O.UNLINKABLE_LIMITED_PII, which enforces the implementation of CredProtection Level as stipulated in the CTAP specification.

OSP.DISABLED_ENTERPRISE_ATTESTATION (for Consumer): The OSP is directly covered by the objective OE.DISABLED_ENTERPRISE_ATTESTATION (for Consumer)

OSP.ENTERPRISE_ATTESTATION (for Enterprise): The OSP is covered by the objectives

- O.ENTERPRISE_ATTESTATION (for Enterprise), which ensures that only allowed RPs shall receive Enterprise Attestation, and
- OE.ENTERPRISE_ATTESTATION (for Enterprise), which ensures the use of unique private keys per identifier.

OSP.ENTERPRISE_ATTESTATION_PROVISIONING_RPLIST (for Enterprise): The OSP is covered by the objectives

- O.ENTERPRISE_ATTESTATION_PROVISIONING (for Enterprise), which ensures that only the Vendor can enable the Enterprise Attestation, and
- O.RP_IDENTIFIERS_LIST_MANAGEMENT (for Enterprise), which ensures that only the Vendor can modify the RP list (if it is supported).

OSP.FACTORY_RESET (Optional): The OSP is directly covered by the objective O.FACTORY_RESET.

OSP.LIMITED_PII: The OSP is directly covered by the objective OE.LIMITED_PII.

OSP.USER_CONSENT: The OSP is directly covered by the objective O.USER_CONSENT.

Table 5-2: Mapping OSPs and security objectives

OSP.ATTESTABLE_PROPERTIES	OSP.CRED_PROTECTION_EXTENSION	OSP.DISABLED_ENTERPRISE_ATTESTATION (for Consumer)	OSP.ENTERPRISE_ATTESTATION (for Enterprise)	OSP.ENTERPRISE_ATTESTATION_PROVISIONING_RPLIST (for Enterprise)	OSP.FACTORY_RESET (Optional)	OSP.LIMITED_PII	OSP.USER_CONSENT	Objectives
								O.ALLOWED_CRYPTOGRAPHY
								O.ASP_PROTECTION
X								O.ATTESTABLE_PROPERTIES
								O.CORRECT_KEY_AND_SIGNATURE_GENERATION
								O.DISABLED_DEBUG
				X				O.ENTERPRISE_ATTESTATION_PROVISIONING (for Enterprise)
			X					O.ENTERPRISE_ATTESTATION (for Enterprise)
					X			O.FACTORY_RESET (Optional)
								O.FUNCTIONALITY_PROTECTION
								O.IMPERSONATION_RESILIENCE
								O.LEAKAGE_RESISTANCE
								O.PIN_DATA_PROTECTION
								O.RNG
				X				O.RP_IDENTIFIERS_LIST_MANAGEMENT
								O.TAMPER_RESISTANCE
								O.TRUSTWORTHY_DATA
	X							O.UNLINKABLE_LIMITED_PII
							X	O.USER_CONSENT
								O.USER_PRESENCE_CHECK
								O.USER_VERIFICATION (Optional)
								OE.ASSURANCE_LEVEL_LIFE
								OE.AUTHENTICATOR_CHECK
								OE.PROTECTION_AFTER_DELIVERY
								OE.DISABLED_DEBUG
		X						OE.DISABLED_ENTERPRISE_ATTESTATION (For Consumer)

OSP.ATTESTABLE_PROPERTIES	OSP.CRED_PROTECTION_EXTENSION	OSP.DISABLED_ENTERPRISE_ATTESTATION (for Consumer)	OSP.ENTERPRISE_ATTESTATION (for Enterprise)	OSP.ENTERPRISE_ATTESTATION_PROVISIONING_RPLIST (for Enterprise)	OSP.FACTORY_RESET (Optional)	OSP.LIMITED_PII	OSP.USER_CONSENT	Objectives
			X					OE.ENTERPRISE_ATTESTATION
						X		OE.LIMITED_PII
								OE.SECURE_VERIFIER_DATABASE
								OE.TRUSTWORTHY_CE
								OE.TRUSTWORTHY_RP
								OE.TRUSTWORTHY_RP_IDENTIFIERS_LIST (for Enterprise)
								OE.TRUSTWORTHY_SERVER_AUTHENTICATION
								OE.USER_DEVICE_SEPARATION_MECHANISM
								OE.ENTERPRISE_USERS

5.3.3 Assumptions

This section presents the rationale of coverage of assumptions by objectives for the TOE operational environment. Table 5-3 provides the mapping between these elements.

A.ASSURANCE_LEVEL_LIFE: The assumption is directly covered by the objective OE. ASSURANCE_LEVEL_LIFE.

A.AUTHENTICATOR_CHECK: The assumption is directly covered by the objective OE. AUTHENTICATOR_CHECK.

A.PROTECTION_AFTER_DELIVERY: The assumption is directly covered by the objective OE PROTECTION_AFTER_DELIVERY.

A.SECURE_VERIFIER_DATABASE: The assumption is directly covered by the objective OE. SECURE_VERIFIER_DATABASE.

A.TRUSTWORTHY_CE: The assumption is directly covered by the objective OE. TRUSTWORTHY_CE.

A.TRUSTWORTHY_RP: The assumption is directly covered by the objective OE. TRUSTWORTHY_RP.

A.TRUSTWORTHY_RP_IDENTIFIERS_LIST: The assumption is directly covered by the objective OE. TRUSTWORTHY_RP_IDENTIFIERS_LIST.

A.TRUSTWORTHY_SERVER_AUTHENTICATION: The assumption is directly covered by the objective OE. TRUSTWORTHY_SERVER_AUTHENTICATION.

A.USER_DEVICE_SEPARATION_MECHANISM: The assumption is directly covered by the objective OE. USER_DEVICE_SEPARATION_MECHANISM.

A.ENTERPRISE_USERS: The assumption is directly covered by the objective OE. ENTERPRISE_USERS.

Table 5-3: Mapping assumptions and security objectives

A.ASSURANCE_LEVEL_LIFE	A.AUTHENTICATOR_CHECK	A.PROTECTION_AFTER_DELIVERY	A.SECURE_VERIFIER_DATABASE	A.TRUSTWORTHY_CE	A.TRUSTWORTHY_RP	A.TRUSTWORTHY_RP_IDENTIFIERS_LIST	A.TRUSTWORTHY_SERVER_AUTHENTICATION	A.USER_DEVICE_SEPARATION_MECHANISM	A.ENTERPRISE_USERS	FIDO2 PP objectives
X										OE.ASSURANCE_LEVEL_LIFE
	X									OE.AUTHENTICATOR_CHECK
		X								OE.PROTECTION_AFTER_DELIVERY
										OE.DISABLED_DEBUG
										OE.DISABLED_ENTERPRISE_ATTESTATION (For Consumer)
										OE.ENTERPRISE_ATTESTATION
										OE.LIMITED_PII
			X							OE.SECURE_VERIFIER_DATABASE
				X						OE.TRUSTWORTHY_CE
					X					OE.TRUSTWORTHY_RP
						X				OE.TRUSTWORTHY_RP_IDENTIFIERS_LIST (for Enterprise)

A.ASSURANCE_LEVEL_LIFE	A.AUTHENTICATOR_CHECK	A.PROTECTION_AFTER_DELIVERY	A.SECURE_VERIFIER_DATABASE	A.TRUSTWORTHY_CE	A.TRUSTWORTHY_RP	A.TRUSTWORTHY_RP_IDENTIFIERS_LIST	A.TRUSTWORTHY_SERVER_AUTHENTICATION	A.USER_DEVICE_SEPARATION_MECHANISM	A.ENTERPRISE_USERS	FIDO2 PP objectives
							X			OE.TRUSTWORTHY_SERVER_AUTHENTICATION
								X		OE.USER_DEVICE_SEPARATION_MECHANISM
									X	OE.ENTERPRISE_USERS

6 EXTENDED SECURITY REQUIREMENTS

6.1 Definition of FPT_EMS.1

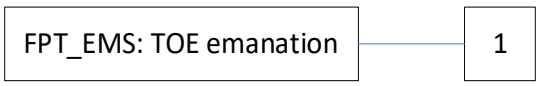
The family FPT_EMS (TOE Emanation) of the class FPT (Protection of the TSF) is defined here to describe the IT security functional requirements of the TOE. The TOE shall prevent attacks against secret data stored in and used by the TOE where the attack is based on external observable physical phenomena of the TOE. Examples of such attacks include evaluation of the TOE's electromagnetic radiation, simple power analysis (SPA), differential power analysis (DPA), timing attacks, etc. This family describes the functional requirements for the limitation of intelligible emanations that are not being directly addressed by any other component of [CC2].

The family TOE Emanation (FPT_EMS) Is specified as follows:

FPT_EMS TOE Emanation

This family defines requirements to mitigate intelligible emanations.

Component levelling:



FPT_EMS.1 TOE emanation has two constituents:

FPT_EMS.1.1 Limit of Emissions requires not emitting intelligible emissions that enable access to TSF data or user data.

FPT_EMS.1.2 Interface Emanation requires not emitting interface emanations that enable access to TSF data or user data.

Management: FPT_EMS.1

There are no management activities foreseen.

Audit: FPT_EMS.1

There are no actions defined to be auditable.

FPT_EMS.1 TOE Emanation

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_EMS.1.1 The TOE shall not emit [assignment: *types of emissions*] in excess of [assignment: *specified limits*] enabling access to [assignment: *list of types of TSF data*] and [assignment: *list of types of user data*].

FPT_EMS.1.2 The TSF shall ensure [assignment: *type of users*] are unable to use the following interface [assignment: *type of connection*] to gain access to [assignment: *list of types of TSF data*] and [assignment: *list of types of user data*].

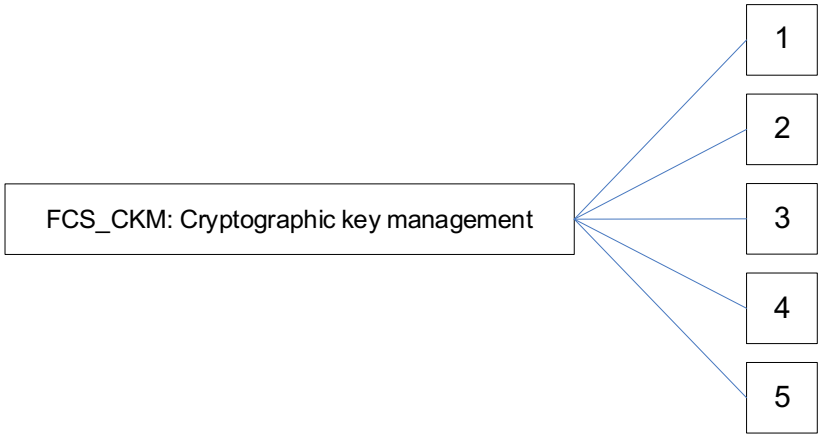
6.2 Definition of FCS_CKM.5

This component describes functional requirements for cryptographic key derivation and signature generation. Key derivation is the process by which one or more cryptographic keys are calculated either from a pre-shared key or based on a shared secret and other information. The component is part of the family FCS_CKM of the class FCS.

The component FCS_CKM.5 has been specified as follows:

FCS_CKM Cryptographic Key Management

Component levelling:



Management: FCS_CKM.5

There are no management activities foreseen.

Audit: FCS_CKM.5

There are no actions defined to be auditable.

FCS_CKM.5 Cryptographic key derivation and signature generation

Hierarchical to: No other components.

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

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.5.1 The TSF shall [selection: *derive, generate*] [selection: [assignment: *signature type*], [assignment: *cryptographic key type*]] from [assignment: *input parameters*] in accordance with [selection: [assignment: *signature generation algorithm*], [assignment: *cryptographic key derivation algorithm*]] and specified [selection: *signature sizes, cryptographic key sizes*] of [assignment: *sizes*] that meet the following: [assignment: *list of standards*].

6.3 Definition of FCS_RNG.1

Family behaviour

To define the IT security functional requirements of the TOE, an additional family (FCS_RNG) of the Class FCS (cryptographic support) is defined here. This family describes the functional requirements for random number generation used for cryptographic purposes.

This family defines quality requirements for the generation of random numbers which are intended to be used for cryptographic purposes.

Component levelling

There is only one level in this family.

FCS_RNG.1 requires that random numbers meet a defined quality metric.

Management: FCS_RNG.1

No management activities are foreseen.

Audit: FCS_RNG.1

No actions are defined to be auditable.

FCS_RNG.1 Random numbers generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a [selection: *physical, non-physical true, deterministic, hybrid, hybrid deterministic*] random number generator that implements: [assignment: *list of security capabilities*].

FCS_RNG.1.2 The TSF shall provide random numbers that meet [assignment: *a defined quality metric*].

7 SECURITY REQUIREMENTS

7.1 Security Functional Requirements

7.1.1 General

The sections 7.1.2 to 7.1.11 define the set of SFRs for the FIDO2 SE Authenticator.

Most of the SFRs are generic and apply to any FIDO2 SE Authenticator as defined in 2.1. Some SFRs apply only to Consumer or Enterprise modes.

The following conventions apply to the statement of the security functional requirements:

- *italics* are used for the parts of an SFR that must be completed/provided by the developer;
- **bold** is used for parts of an SFR attributed/instantiated in this PP;
- footnotes indicate the SFR as included in the [CC2] catalogue.

7.1.2 Cryptography

7.1.2.1 FCS_CKM.1 Cryptographic key generation

Hierarchical to: No other components.

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

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1[7] The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: *cryptographic key generation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: ***applicable standards compliant with [FCrypto]***].

Refinement:

1. Only an Allowed Random Number Generator shall be used for key generation resulting in an Authenticator Security Parameter
2. Only an Allowed Random Number Generator shall be used to generate the pinToken or pinUvAuthToken if used by the Authenticator.

Application Note: FCS_CKM.1.1 applies to the generation of all cryptographic keys, including ASPs that are cryptographic keys, including attestation keys if these are generated on-card.

[7] FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: *cryptographic key generation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].

Application Note:

- The SFR applies to all cryptographic keys.

7.1.2.2 FCS_CKM.4 Cryptographic key destruction

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes], or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1[8] The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [assignment: *cryptographic key destruction method*] that meets the following: [assignment: *list of standards*].

Application Note:

- The means by which the destruction of cryptographic keys is accomplished is implementation dependent. Examples of key destruction mechanisms include:
  - Physically overwriting the keys, for instance by using a method that meets the FIPS 140-2;
  - Deleting the keys;
  - Destroying the key material used to encrypt the keys.

7.1.2.3 FCS_CKM.5/KD Cryptographic key derivation and signature generation

Editor's note: This SFR could also be stated by using FCS_CKM.1. If this option is chosen, then the refinements must be merged with those already present in FCS_CKM.1.

Hierarchical to: No other components.

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

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.5.1/KD The TSF shall *derive cryptographic keys* from [assignment: *input parameters*] in accordance with a *specified cryptographic key derivation* algorithm and specified *cryptographic key sizes* of [assignment: *sizes*] that meet the following: [assignment: ***applicable standards compliant with [FCrypto]***].

Refinement:

1. Only an Allowed Key Derivation Function shall be used for key generation resulting in an Authenticator Security Parameter
2. Only an Allowed Key Derivation Function shall be used to generate the pinToken or pinUvAuthToken if used by the Authenticator.

[8] FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [assignment: *cryptographic key destruction method*] that meets the following: [assignment: *list of standards*].

3. If the Authenticator generates a key with an Allowed Key Derivation Function, then the security level of the Allowed Key Derivation Function SHALL be at least as large as the claimed cryptographic level of the key generated.

7.1.2.4 FCS_CKM.5/SG Cryptographic key derivation and signature generation

Hierarchical to: No other components.

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

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.5.1/SG The TSF shall *generate signatures* from [assignment: *input parameters*] in accordance with a *specified signature generation* algorithm and specified *signature sizes* of [assignment: *sizes*] that meet the following: [assignment: ***applicable standards compliant with [FCrypto]***].

7.1.2.5 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]

FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1[9] The TSF shall perform [assignment: *list of cryptographic operations*] in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: ***applicable standards compliant with [FCrypto]***].

Refinement:

The cryptographic operations shall comply with the following principles:

1. When responding to a “Register”, “Sign”, “Deregister”, "MakeCredentials", or "GetAssertion" command, the Authenticator shall use an Allowed Hashing or Data Authentication Cryptographic Function or an alternative equivalent method to bind keys to apps or RP ID.
2. If the Authenticator uses a key with parameters generated by an Allowed Key Derivation Function, then the security level of the Allowed Key Derivation Function shall be at least as large as the claimed cryptographic level of the key used.
3. If the Authenticator adds randomly generated nonces, then an *Allowed* Random Number Generator shall be used (cf. FCS_RNG.1).

Application Note:

- The ST author is expected to instantiate FCS_COP.1 for all the cryptographic operations performed by the Authenticator.

[9] FCS_COP.1.1 The TSF shall perform [assignment: *list of cryptographic operations*] in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].

- For functions such as software update, the cryptographic operations are performed by the SE and are then covered through the SE Security Target.
- For refinement 1, hashing collectedClientData and other relevant elements (e.g. discoverable/non-discoverable credentials) SHOULD be applicable.

7.1.3 Unlinkability and limited personal information

7.1.3.1 FPR_ANO.2 Anonymity without soliciting information

Hierarchical to: FPR_ANO.1 Anonymity

Dependencies: No dependencies.

FPR_ANO.2.1[10] The TSF shall ensure that the **RP instances** are unable to determine the real user name bound to **a FIDO2 SE Authenticator**.

FPR_ANO.2.2[11] The TSF shall provide

- **registration**
- **authentication**
- [assignment: *list of services*]

to **RP instances** without soliciting any reference to the real user name.

Application Note:

- The expression “real user name” denotes any personal identification information (PublicKeyCredentialUserEntity).

Application Note:

- Second-factor Authenticators shall not store PublicKeyCredentialUserEntity inside a Raw Key Handle that was not cryptographically wrapped.

[10] FPR_ANO.2.1 The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine the real user name bound to [assignment: *list of subjects and/or operations and/or objects*].

[11] FPR_ANO.2.2 The TSF shall provide [assignment: *list of services*] to [assignment: *list of subjects*] without soliciting any reference to the real user name.

7.1.3.2 FPR_UNL.1/NO_INFERENCE1 Unlinkability

Hierarchical to: No other components.

Dependencies: No dependencies.

FPR_UNL.1.1[12]/NO_INFERENCE1 The TSF shall ensure that **any two different RP instances** are unable to determine whether **registration or authentication operations were caused by the same user**.

Application Note (for the Consumer case):

- This means that there is no Correlation Handle that is visible across multiple RPs and traces back to the user (KeyId/Credential ID, Key Handles).

Application Note (for the Enterprise case):

- This means that there is no Correlation Handle that is visible across multiple RPs and traces back to the user (KeyID/Credential ID, Key Handles), except for the unique identifier that is present in the Enterprise Attestation Certificate.

7.1.3.3 FPR_UNL.1/NO_INFERENCE2 Unlinkability

Hierarchical to: No other components.

Dependencies: No dependencies.

FPR_UNL.1.1[13]/NO_INFERENCE2 The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine whether [assignment: *list of operations*] [selection: *were caused by the same user, are related as follows* [assignment: *list of relations*]].

Editorial refinement:

The TSF shall ensure that **an RP instance** is unable **to use the conveyed information to uniquely identify the Authenticator instance to a different RP**.

7.1.3.4 FPR_UNL.1/NO_INFERENCE3 Unlinkability

Hierarchical to: No other components.

Dependencies: No dependencies.

[12] FPR_UNL.1.1 The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine whether [assignment: *list of operations*] [selection: *were caused by the same user, are related as follows* [assignment: *list of relations*]].

[13] FPR_UNL.1.1 The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine whether [assignment: *list of operations*] [selection: *were caused by the same user, are related as follows* [assignment: *list of relations*]].

FPR_UNL.1.1[14]/NO_INFERENCE3 The TSF shall ensure that **RP instances** are unable to determine whether **the same FIDO2 SE Authenticator is used by two different user accounts**.

Application Note:

- It is assumed that the CTAP protocol ensures the property of unlinkability by-design. This SFR refers to verifying the correct and robust implementation of the CTAP protocol.

7.1.3.5 FPR_UNL.1/ID Unlinkability

Hierarchical to: No other components.

Dependencies: No dependencies.

FPR_UNL.1.1[15]/ID The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine whether [assignment: *list of operations*] [selection: *were caused by the same user, are related as follows* [assignment: *list of relations*]].

Editorial refinement:

The TSF shall ensure that **any third party with physical access to the FIDO2 SE Authenticator** are unable to determine **the AppIDs or RP IDs the FIDO2 SE Authenticator has been registered to without having user consent.**

Application Note:

- Note that all FIDO SE Authenticators are expected to meet this requirement. For FIDO2 SE Authenticators that store the Uauth key pair and do not implement user verification (or only implement user presence check) a response that cannot be distinguished from a valid authentication response must be provided.

7.1.3.6 FPR_UNL.1/Info Unlinkability

Hierarchical to: No other components.

Dependencies: No dependencies.

FPR_UNL.1.1[16]/Info The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine whether [assignment: *list of operations*] [selection: *were caused by the same user, are related as follows* [assignment: *list of relations*]].

[14] FPR_UNL.1.1 The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine whether [assignment: *list of operations*] [selection: *were caused by the same user, are related as follows* [assignment: *list of relations*]].

[15] FPR_UNL.1.1 The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine whether [assignment: *list of operations*] [selection: *were caused by the same user, are related as follows* [assignment: *list of relations*]].

[16] FPR_UNL.1.1 The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine whether [assignment: *list of operations*] [selection: *were caused by the same user, are related as follows* [assignment: *list of relations*]].

Editorial refinement:

> The TSF shall ensure that **any third party** are unable to determine **whether a key was registered or not for a given RP ID, user-specific information, or RP-specific information without verifying the user or requiring a Credential ID depending on the configuration of the Authenticator.**

Application Note:

- This SFR means that the FIDO2 SE Authenticator can reveal such information only under strict conditions as specified in CTAP.

7.1.4 Leakage Resistance

7.1.4.1 FDP_RIP.1 Subset residual information protection

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_RIP.1.1[17] The TSF shall ensure that any previous information content of a resource is made unavailable upon the **deallocation of the resource from** the following objects: **ASPs, ASPs persistent or transient related data, and any object containing user specific information**.

Application Note:

- This SFR covers all ASPs including keys and PIN. The goal is that if such objects are “destroyed” then they are permanently unavailable so they can never be read or used again.
- Deallocation covers the normal operation of the Authenticator and the optional factory reset operation which implies deleting all user specific information, including confidential ASPs.

7.1.4.2 FPT_EMS.1 TOE Emanation

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_EMS.1.1[18] The TOE shall not emit [assignment: *types of emissions*] in excess of [assignment: *specified limits*] enabling access to:

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

[18] FPT_EMS.1.1 The TOE shall not emit [assignment: *types of emissions*] in excess of [assignment: *specified limits*] enabling access to [assignment: *list of types of TSF data*] and [assignment: *list of types of user data*].

- **the seed**
- **the MACKey**
- **the Attestation Key**
- **the authentication private keys**
- **the PIN**
- **the secret ASPs**
- **[assignment: *list of types of TSF data or user data*].**

Refinement:

1. Secret ASPs and any other cryptographic private key shall not be leaked at a rate that would allow weakening the key below its claimed cryptographic strength, even after observing all allowed key uses.
2. The variations in the amount of time required to perform a cryptographic operation shall not allow reducing the security of secret ASPs or private cryptographic keys below their claimed cryptographic strength.
3. The length of time required to perform a cryptographic operation using a secret ASP (cryptographic key or secret) shall not depend on the value of that secret or cryptographic key.

FPT_EMS.1.2[19] The TSF shall ensure **all users** are unable to use the following interface: **any of the TOE's contactless/contact-based interfaces and circuit contacts** to gain access to:

- **the seed**
- **the MACKey**
- **the Attestation Key**
- **the authentication private keys**
- **the PIN**
- **the secret ASPs**
- **[assignment: *list of types of TSF data or user data*].**

7.1.5 User presence and user verification

7.1.5.1 (Optional) FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

[19] FPT_EMS.1.2 The TSF shall ensure [assignment: *type of users*] are unable to use the following interface [assignment: *type of connection*] to gain access to [assignment: *list of types of TSF data*] and [assignment: *list of types of user data*].

FIA_AFL.1.1[20] The TSF shall detect when [selection: [assignment: *positive integer number*, *an administrator configurable positive integer within [assignment: range of acceptable values] in a*] unsuccessful authentication attempts occur related to:

- **user verification**
- [assignment: *list of authentication events*].

FIA_AFL.1.2[21] When the defined number of unsuccessful authentication attempts has been [selection: **met, surpassed**], the TSF shall [assignment: *list of actions*].

Application Note:

- This SFRs applies to FIDO2 Authenticators implementing user verification other than user presence check.
- This SFR does not apply to user presence checks.
- The aim of this SFR is to rate-limit user verification attempts to prevent brute force attacks. Recommendations include:
  - Allowing up to 3 failed user verification attempts without any penalty and then imposing a delay of at least 30 seconds before the 4th one,
  - Increasing exponentially with each successive attempt (e.g., 1 minute before the 5th one, 2 minutes before the 6th one), or
  - Disable the biometric user authentication and offer another factor (e.g., a different biometric modality or a PIN/Passcode if it is not already a required factor) if such an alternative method is already available after the 5th failed user verification attempt.
  - Disabling the first user verification method and falling back to an alternative user verification method MAY take place at any time without imposing additional delays.

7.1.5.2 FIA_UAU.5 Multiple authentication mechanisms

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.5.1[22] The TSF shall provide

a) User presence check, i.e. “a mechanism to obtain a gesture or action from the user establishing that the user authorizes the given authentication action”

[20] FIA_AFL.1.1 The TSF shall detect when [selection: [assignment: *positive integer number*], *an administrator configurable positive integer within [assignment: range of acceptable values]*] unsuccessful authentication attempts occur related to [assignment: *list of authentication events*].

[21] FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [selection**: met, surpassed**], the TSF shall [assignment: *list of actions*].

[22] FIA_UAU.5.1 The TSF shall provide [assignment: *list of multiple authentication mechanisms*] to support user authentication.

b) Optionally, the following user verification methods without implicit user presence check [selection: *none*, [assignment: *list of user verification mechanisms without implicit user presence check, which may include PIN verification*]]

c) **Optionally, the following user verification methods with implicit user presence check [selection: *none*, [assignment: *list of user verification mechanisms without implicit user presence check, which may include PIN verification*]**

to support user authentication.

FIA_UAU.5.2[23] The TSF shall authenticate any user's claimed identity according to the **following rules**

a) [assignment: *rules describing how user presence check is provided and the effect in the TOE*]

b) Optionally, [assignment: *rules describing how any user verification mechanism without implicit user presence check provides authentication and the effect in the TOE*]

c) Optionally, [assignment: *rules describing how any user verification mechanism with implicit user presence check provides authentication and the effect in the TOE*].

Application Note:

- For an NFC SE, tapping with the NFC reader is considered a user gesture.

7.1.5.3 FIA_UAU.6 Re-authenticating

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.6.1[24] The TSF shall re-authenticate the user under the conditions

a) there is an authentication request for user presence or user verification, and

b) either the flag of the required authentication mechanism is unset

c) or the cached period associated with the required authentication mechanism has expired.

Application Note:

- The ‘flag’ stands for ‘user presence flag’ and to “user verification flag’ or ‘user verification with implicit user presence flag’ for TOEs supporting such mechanisms.

[23] FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [assignment: *rules describing how the multiple authentication mechanisms provide authentication*].

[24] FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions [assignment: list of conditions under which re-authentication is required].

7.1.6 Authenticator SFP

The FIDO2 Authenticator SFP used in the FDP-related SFRs is an abstraction of the CTAP 2.1 specification focused on the following aspects:

- User presence
- User consent
- Cryptographic rules
- Validity of cryptographic material
- Maximum number of (cryptographic) operations.

The rules defined in FDP_IFF.1 are extracted from the following table, based on CTAP 2.1 specification.

Table 7-1: UP/UV rules

Operation	**UP Mandatory**	**UP Required under specific conditions/ UP Optional**	**UV Mandatory**	**UV Required under specific conditions/ UV Optional**	**UV Not required**
authenticatorMakeCredential	X			if uv_protected* = true and makeCredUVNotRqd = false or absent or alwaysUV is enabled or the RP requires it	when makeCredUVNotRqd = true (and the Credential to be created is not expected to be discoverable, i.e. rk = false)
authenticatorGetAssertion		when RP requires it		when RP requires it or uv_protected* = true or alwaysUV is enabled	
authenticatorGetNextAssertion	Can only be performed in a certain time limit following an authenticatorGetAssertion command				
authenticatorGetInfo	X				X
authenticatorClientPIN		If userPresent = true	X		
authenticatorReset	X				

Operation	UP Mandatory	UP Required under specific conditions/ UP Optional	UV Mandatory	UV Required under specific conditions/ UV Optional	UV Not required
authenticatorBioEnrollment				Mandatory for/if subcommand is: - enrolling fingerprint - enumerate enrollments - rename fingerprint - - remove enrollment	Not required for: - cancel current enrollment - get fingerprint sensor info
authenticatorCredentialManagement			X		
authenticatorSelection	X				
authenticatorLargeBlobs				if uv_protected* = true or alwaysUV is enabled	
authenticatorConfig				if uv_protected* = true or alwaysUV is enabled	

7.1.6.1 FDP_ETC.2 Export of user data with security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_ETC.2.1[25] The TSF shall enforce the **Authenticator SFP** when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.2.2 The TSF shall export the user data with the user data's associated security attributes.

FDP_ETC.2.3 The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data.

[25] FDP_ETC.2.1 The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.2.4[26] The TSF shall enforce the following rules when user data is exported from the TOE: **Any ASP shall be signed, and any secret ASP shall be encrypted; more precisely:**

1. **Exportable FIDO user verification reference data shall be wrapped for the Authenticator only.**
2. **Exportable Authenticator User Private Key shall be wrapped for the Authenticator only.**
3. **Any ASP that is exported outside the TOE shall be protected against modification using an Allowed Data Authentication, Signature, or Key Protection Cryptographic Function.**
4. **Any secret ASP that is exported outside the TOE shall be encrypted using an Allowed Key Protection Cryptographic Function.**
5. **Any key used to protect a secret ASP that is exported outside the TOE shall have a claimed cryptographic strength greater than or equal to the claimed cryptographic strength of the key being wrapped.**

[assignment: *additional exportation control rules, if applicable*].

7.1.6.2 FDP_IFC.1 Subset information flow control

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1[27] The TSF shall enforce the **FIDO2 Authenticator SFP** on

- **Subjects:**
  - **the Authenticator application**
- **Information:**
  - **the input data conveyed by the RP's operation request, and**
  - **the output data conveyed by the Authenticator's answer**
- **Operations:**
  - **commands defined in CTAP 2.1, i.e.**
    - **authenticatorBioEnrollment for subcommands cancel enrollment, enrolling fingerprint, enumerate enrollments, get fingerprint sensor info, remove enrollment and rename fingerprint**
    - **authenticatorClientPIN**
    - **authenticatorConfig**
    - **authenticatorCredentialManagement**
    - **authenticatorGetAssertion**
    - **authenticatorGetInfo**

[26] FDP_ETC.2.4 The TSF shall enforce the following rules when user data is exported from the TOE: [assignment: *additional exportation control rules*].

[27] FDP_IFC.1.1 The TSF shall enforce the [assignment: *information flow control SFP*] on [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*].

- **authenticatorGetNextAssertion**
- **authenticatorLargeBlobs**
- **authenticatorMakeCredential**
- **authenticatorReset**
- **authenticatorSelection**

▪ [assignment: *list of operations*].

7.1.6.3 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[28] The TSF shall enforce the **FIDO2 Authenticator SFP** based on the following types of subject and information security attributes:

- Subject security attributes:
  - **up_cached_period, up_cached_period_limit**
  - **always_uv**
  - **for any supported uv mechanism, uv_enabled, uv_cached_period, uv_cached_period_limit, uv_try_counter, uv_try_limit**
  - **for any supported uv_up mechanism, uv_up_enabled, uv_up_cached_period, uv_up_cached_period_limit, uv_try_counter, uv_try_limit**
  - **for any key and certificate, key_validity, certificate_validity**
  - **(optional) signature_counter, signature_limit**
  - **(optional) operation_counters**
- Information security attributes:
  - **up_requested**
  - **uv_requested.**

FDP_IFF.1.2[29] The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

a) For authenticatorGetInfo:

i. user is present if the RP requires it

[28] FDP_IFF.1.1 The TSF shall enforce the [assignment: *information flow control SFP*] based on the following types of subject and information security attributes: [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*].

[29] FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*].

b) For authenticatorReset, authenticatorSelection:
   i. user is present
c) For authenticatorMakeCredential:
   i. user is present, and
   ii. user is verified if either the Authenticator supports UV and RP requires user verification, or the Authenticator always requires user verification
d) For authenticatorGetAssertion:
   i. user is present if the RP requires it, and
   ii. user is verified if either the Authenticator supports UV and RP requires user verification, or the Authenticator always requires user verification
e) For authenticatorLargeBlobs:
   i. writing serialized data case: user is verified if the Authenticator supports or always requires user verification
   ii. reading serialized data case: user is verified
f) For authenticatorConfig:
   i. user is verified if the Authenticator supports or always requires user verification
g) For authenticatorClientPIN and all subcommands:
   i. user is present if required by the RP, and
   ii. user is verified
h) For authenticatorBioEnrollment:
   i. user is verified for enrolling fingerprint, enumerate enrollments, rename fingerprint, and remove enrollment
i) For authenticatorCredentialManagement
   i. user is verified

where

- 'user is present' means either that user presence was checked previously and the up_cached_period (uv_up_cached_period) has not reached its limit, or that user presence is successfully checked in the context of the operation
- 'user is verified' means either that the user was verified previously and the uv_cached_period (uv_up_cached_period) has not reached its limit, or that user is successfully verified in the context of the operation in less than uv_try_counter.
- 'permit an information flow' means that the specified conditions are necessary, not sufficient, i.e. the complete set of conditions is given in the CTAP specification.

FDP_IFF.1.3[30] The TSF shall enforce **the following rules when performing an operation:**

a) **A proof of user consent is required before a relationship to a new Relying Party is established.**
b) **For each new registration an independent User Authentication Key shall be generated, as specified in [assignment: *instance of FCS_CKM.1*]**
c) **A key provided to the Authenticator in a Key Handle shall only be used for signing if the key was generated by the Authenticator.**
d) **A key provided to the Authenticator in a Key Handle shall only be used for signing if the Key Handle is associated with the RP ID.**
e) **The Attestation Private Key shall only be used to sign well-formed FIDO attestation objects.**
f) **All Authenticator User Private Keys shall only be usable for generating well-formed FIDO signature assertions.**
g) **(For Enterprise) The Authenticator shall only return an Enterprise Attestation for RP IDs that are in the RP ID list (if supported).**

FDP_IFF.1.4[31] The TSF shall explicitly authorise an information flow based on the following rules:

- **For authenticatorGetNextAssertion:**
  - **The operation is performed within the set time limit following an authenticatorGetAssertion**

FDP_IFF.1.5[32] The TSF shall explicitly deny an information flow based on the following rules:

- **The operation is not supported by the Authenticator (cf. FDP_IFC.1).**
- **The operation requires user verification, and no user verification mechanism is supported and enabled.**
- **The operation requires the use of a cryptographic key or certificate which has expired.**
- **(Optional, for supported signature counter(s)) The operation is intended to perform a signature and the signature counter(s) have reached their limit.**
- **(Optional, for supported the counter is supported) The operations counter(s) have reached their limit.**

[30] FDP_IFF.1.3 The TSF shall enforce the [assignment: *additional information flow control SFP rules*].

[31] FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly authorise information flows*].

[32] FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly deny information flows*].

- **[assignment: *list of complementary denial rules*].**

7.1.6.4 FDP_ITC.1 Import of user data without security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_MSA.3 Static attribute initialisation

FDP_ITC.1.1[33] The TSF shall enforce the **Authenticator SFP** when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3[34] The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE:

- **data input is validated**
- [assignment: *additional importation control rules, if applicable*].

Application Note:

- Data input is validated to protect against invalid input-based attack vectors, e.g. buffer overruns, stack overflows and integer under/overflow.

7.1.6.5 FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1[35] The TSF shall enforce the **FIDO2 Authenticator SFP** to restrict the ability to **modify** the security attributes

[33] FDP_ITC.1.1 The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] when importing user data, controlled under the SFP, from outside of the TOE.

[34] FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [assignment: *additional importation control rules*].

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

- **alwaysUV**
- **up_cached_period**
- **for any supported uv mechanism: uv_support, uv_enabled, uv_cached_period, uv_try_counter**
- **for any supported uv_up mechanism: uv_support, uv_up_enabled, uv_up_cached_period, uv_try_counter**
- **for any key and certificate, key_validity, certificate_validity**
- **(optional) signature_counter**
- **(optional) operation_counters**

to the **TSF itself during the performance of an operation controlled by the SFP**.

7.1.6.6 FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

FMT_MSA.3.1[36] The TSF shall enforce the **FIDO2 Authenticator SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2[37] The TSF shall allow **no role** to specify alternative initial values to override the default values when an object or information is created.

7.1.7 Functionality Protection

7.1.7.1 FMT_MOF.1 Management of security functions behavior

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MOF.1.1[38] The TSF shall restrict the ability to [selection: *determine the behaviour of, disable, enable, modify the behaviour of*] the functions [assignment: *list of functions*] to [assignment: *the authorised identified roles*].

Editorial refinement:

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

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

[38] FMT_MOF.1.1 The TSF shall restrict the ability to [selection: determine the behaviour of, disable, enable, modify the behaviour of] the functions [assignment: list of functions] to [assignment: the authorised identified roles].

The TSF shall restrict the ability to **enable the Enterprise Attestation functionality** to the **Vendor**.

7.1.7.2 FMT_MTD.1/CONFIG

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1[39]/CONFIG The TSF shall restrict the ability to ***modify*** the **TOE's security characteristics** to **the Vendor (or its delegates)**.

Application Note:

- *"Security characteristics" is defined as an answer to a security requirement, including data confidentiality, data integrity, authentication, non-repudiation, access control and availability.*

7.1.7.3 FMT_MTD.1/DEBUG 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[40]/DEBUG The TSF shall restrict the ability to **query, modify, or retrieve the FIDO2 SE Authenticator data for debug purposes** to **no role**.

7.1.7.4 FMT_MTD.1/PROPERTIES Management of TSF data

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

[39] FMT_MTD.1.1 The TSF shall restrict the ability to [selection: *change_default, query ,modify, delete, clear, [assignment: other operations]*] the [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].

[40] FMT_MTD.1.1 The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*] the [assignment: list of TSF data] to [assignment: *the authorised identified roles*].

FMT_MTD.1.1[41]/PROPERTIES	The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*] the [assignment: list of TSF data] to [assignment: *the authorised identified roles*].
Refinement:	The TSF shall allow and restrict the ability to **query the FIDO2 SE Authenticator model/type** to **Relying Parties**.

7.1.7.5 (Optional) (For Enterprise) FMT_MTD.1/RP_LIST 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[42]/RP_LIST	The TSF shall restrict the ability to **modify and delete** the **configured RP identifiers list** to **Vendor**.

Application Note:

- This SFR applies to FIDO2 SE Authenticators supporting Enterprise Attestation.

7.1.7.6 FMT_MTD.3/KH Secure TSF data

Hierarchical to: No other components.

Dependencies: FMT_MTD.1 Management of TSF data

FMT_MTD.3.1[43]/KH	The TSF shall ensure that only secure values are accepted for **Key Handles containing a signature key provided to the Authenticator.**

Application Note:

- Secure value means that the Key Handle contains a key that was generated by the Authenticator and is associated with the RP ID.

7.1.7.7 FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

[41] FMT_MTD.1.1 The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*] the [assignment: list of TSF data] to [assignment: *the authorised identified roles*].

[42] FMT_MTD.1.1 The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*] the [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].

[43] FMT_MTD.3.1 The TSF shall ensure that only secure values are accepted for [assignment: *list of TSF data*].

Dependencies: No dependencies.

FMT_SMF.1.1[44] The TSF shall be capable of performing the following management functions:

- **(Optional) Factory reset**
- [assignment: *list of management functions to be provided by the TSF*].

7.1.7.8 FMT_SMR.1 Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1[45] The TSF shall maintain the roles **Vendor (or its delegates), Relying Parties** [assignment: *the authorised identified roles*].

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

Application Note:

- The ST author will specify FIA_UID.1 if necessary to identify the Vendor.

7.1.7.9 FPT_FLS.1 Failure with preservation of secure state

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_FLS.1.1[46] The TSF shall preserve a secure state when the following types of failures occur:

- **Failure of data input validation as specified in FDP_ITC.1.3**
- **Failure of key binding as specified in FMT_MTD.3/KH**
- **Failures related to abuse of functionality, including user presence/user verification**
- [assignment: *list of types of failures in the TSF*].

Application Note: The types of failure include:

- Data input validation errors

[44] FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [assignment: *list of management functions to be provided by the TSF*].

[45] FMT_SMR.1.1 The TSF shall maintain the roles [assignment: *the authorised identified roles*].

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

- Key Handles which do not satisfy FMT_MTD.3/KH.

Application Note:

- Some types of failures are handled by the SE, e.g. software update failure.

7.1.8 Tamper Protection

7.1.8.1 FPT_PHP.3 Resistance to physical attack

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_PHP.3.1[47] The TSF shall resist **physical tampering including physical manipulation and probing, induced fault analysis** [assignment: *additional physical tampering scenarios*] to the **Authenticator** by responding automatically such that the SFRs are always enforced.

7.1.9 ASP and PIN Data Protection

7.1.9.1 FDP_SDI.1 Stored data integrity monitoring

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_SDI.1.1[48] The TSF shall monitor user data stored in containers controlled by the TSF for **integrity errors (*modification or substitution)*** on all objects, based on the following attributes: [assignment: *user data attributes*].

Application Note:

- User data stands for ASPs including PIN data.

7.1.9.2 FMT_MTD.1/ASP_PIN Management of TSF data

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

[47] FPT_PHP.3.1 The TSF shall resist [assignment: *physical tampering scenarios*] to the [assignment: *list of TSF devices/elements*] by responding automatically such that the SFRs are always enforced.

[48] FDP_SDI.1.1 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*].

FMT_MTD.1.1[49]/ASP_PIN The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*] the **TSF data covered in FMT_MTD.2/ASP_PIN, FMT_MTD.3/KH**, [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].

Refinement:

- If the Authenticator implements signature counter(s), then the TSF shall ensure they are strictly monotonic.

7.1.9.3 FMT_MTD.2/ASP_PIN Management of limits on TSF data

Hierarchical to: No other components.

Dependencies: FMT_MTD.1 Management of TSF data

FMT_SMR.1 Security roles

FMT_MTD.2.1[50]/ASP_PIN The TSF shall restrict the specification of the limits for

- **the PIN counter,**
- **the signature counter(s), if the Authenticator implements such counters,**
- **the number of times a private cryptographic keys or ASPs that are cryptographic keys can be used,**
- [assignment: *list of TSF data*]

to [assignment: *the authorised identified roles*].

FMT_MTD.2.2[51]/ASP_PIN The TSF shall take the following actions, if the TSF data are at, or exceed, the indicated limits: [assignment: *actions to be taken*].

7.1.9.4 FTP_TRP.1 Trusted path

Hierarchical to: No other components.

Dependencies: No dependencies.

[49] FMT_MTD.1.1 The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear*, [assignment: *other operations*]] the [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].

[50] FMT_MTD.2.1 The TSF shall restrict the specification of the limits for [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].

[51] FMT_MTD.2.2 The TSF shall take the following actions, if the TSF data are at, or exceed, the indicated limits: [assignment: *actions to be taken*].

FTP_TRP.1.1[52] The TSF shall provide a communication path between itself and [selection: *remote, local*] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from **modification, disclosure, substitution, injection** [assignment: *other types of integrity or confidentiality violation*].

FTP_TRP.1.2[53] The TSF shall permit **the TSF and local users** to initiate communication via the trusted path.

FTP_TRP.1.3[54] The TSF shall require the use of the trusted path for:

- **any operation or service requiring direct input from the user to the TSF**
- **any operation or service of the TSF providing direct output to the user**
- *[assignment: other services for which trusted path is required].*

7.1.10 Random numbers generation

7.1.10.1 FCS_RNG.1 Random numbers generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a [selection: *physical, non-physical true, deterministic, hybrid, hybrid deterministic*] random number generator that implements: [assignment: ***list of security capabilities compliant with [FCRYPTO]***].

Refinement:

1. If the Authenticator implements a Deterministic Random Number Generator, then an Allowed Physical True Random Number Generator SHALL always be used for seeding (seed, re-seed, seed update).

FCS_RNG.1.2 The TSF shall provide random numbers that meet [assignment: ***a defined quality metric compliant with [FCRYPTO]***].

Refinement:

2. The security strength of any Authenticator's Allowed Deterministic Random Number Generator shall be at least as large as the largest claimed cryptographic strength of any key generated or used.

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

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

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

Application Note:

- The RNG shall be used for generating the seed, the MACKey, the nonces.

7.1.11 Replay detection

7.1.11.1 FPT_RPL.1 Replay detection

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_RPL.1.1[55] The TSF shall detect replay for the following entities:

- **Any user verification data**
- **Any user presence check signal**
- **Any ASP**
- [assignment: *list of identified entities*].

FPT_RPL.1.2[56] The TSF shall perform [assignment: *list of specific actions*] when replay is detected.

7.2 Security Assurance Requirements

The Evaluation Assurance Level is EAL4 augmented with ALC_DVS.2 and AVA_VAN.5. The refinements applicable to the assurance components are presented in Table 7-2 and Table 7-3.

Table 7-2: Refinement of ASE components

SARs	Refinements
ASE_CCL.1	NA
ASE_ECD.1	NA
ASE_INT.1	The actual boundary of the TOE, i.e. all hardware and software used for user presence check, user verification, key generation, signature generation, etc. must be described. If the Transaction Confirmation Display is supported, where and how this is implemented must be described.

[55] FPT_RPL.1.1 The TSF shall detect replay for the following entities: [assignment: *list of identified entities*].

[56] FPT_RPL.1.2 The TSF shall perform [assignment: *list of specific actions*] when replay is detected.

SARs	Refinements
	It shall be documented whether the TOE is a first-factor or a second-factor Authenticator.
	The overall cryptographic strength, which shall be less than or equal to the claimed cryptographic strength of all the Authenticator Security Parameters that are cryptographic keys.
	It shall be documented whether Signature Counters are supported and if they are supported, it shall be documented whether one Signature Counter per authentication key is implemented or one (global) Signature Counter for all authentication keys (i.e. at least one counter covering multiple keys).
ASE_OBJ.2	NA
ASE_REQ.2	NA
ASE_SPD.1	NA

Table 7-3: Refinement of ADV, AGD, ALC, ATE and AVA components

SARs	Refinements
ADV_ARC.1	If the Authenticator uses an alternative method to bind keys to apps/RP ID (cf. FCS_COP.1 refinement 1), then this method shall provide equivalent security to the standard method. It shall be documented how the following is enforced (1) prevent other apps (not originating from the same RP) from using the key and (2) in the case of bound Authenticators, prevent other FIDO Clients of triggering the use of that key, and (3) to what extend the method relies on the underlying HLOS platform.
	The operating environment (the Java Card platform) shall prevent non-Authenticator processes from reading, writing and modifying running or stored Authenticator Application and its associated memory. (Linked to ADV_COMP)
	The operating environment (the Java Card platform) shall not be able to be modified in a way that undermines the security of the Authenticator. (Linked to ADV_COMP)
ADV_FSP.4	The Authenticator shall not provide information to one Relying Party that can be used to uniquely identify that Authenticator instance to a different Relying Party.
ADV_IMP.1	NA
ADV_TDS.3	For each Authentication Private Key the following information shall be specified: - Private key storage location; - Structure; • Relation to/Interaction with Key Handles and Key IDs used by the FIDO2 SE Authenticator.

SARs	Refinements
(Remark: such information can alternatively be provided as part of ADV_ARC.1 or ADV_FSP.4.)	All ASPs shall be documented (recall that at a minimum ASPs include all configuration data and settings, user verification reference data, user verification tokens, key handle access tokens, signature or registration operation counters, privacy sensitive data, cryptographic keys used for PIN management.)
	For each ASP the following information shall be specified: - Measures implemented to protect the ASP; - ASP storage location and storage protection mechanisms; - Whether the ASP is an input or an output and how it is used; - Whether or not the ASP is destroyed; - All secret ASPs shall be specified as confidential.
	For each ASP that is a cryptographic key the following information shall be specified: - How the key was generated; - Whether the key is unique or shared across multiple FIDO2 SE Authenticators; - The key's cryptographic strength. The key's cryptographic strength shall fulfil the requirements specified in the *Allowed Cryptography List* [FCrypto].
AGD_OPE.1	The security configuration of the operating environment (Java Card platform) shall be fully under control of the Authenticator vendor or its delegates such that the security configuration present at commercial shipment cannot be changed except for in-the-field updates that are also fully under control of the Authenticator device vendor or its delegates. (Linked to AGD_COMP)
AGD_PRE.1	The model of the FIDO2 SE Authenticator (i.e. all mandatory metadata fields) shall be accurately specified.
	It shall be documented whether the attestation root certificate is shared across multiple Authenticator models.
	It shall be documented whether the attestation certificate includes the Authenticator model (e.g. AAID or AAGUID). This is mandatory if the root certificate is shared across multiple Authenticator models.
ALC_CMC.4	NA
ALC_CMS.4	The revision control system SHALL, at minimum, track changes to all software or hardware specifications, implementation files, and all tool chains used in the production of the final Authenticator
ALC_DEL.1	NA
ALC_DVS.2	NA

SARs	Refinements
ALC_LCD.1	(For Consumer only) There is no Correlation Handle (KeyID/CredentialID, Key Handle) that is visible across multiple Relying Parties. If the same exact Attestation Key is put into a group of FIDO2 SE Authenticators, then the group of FIDO2 SE Authenticators must be at least 100.000 in number. If less than 100.000 FIDO2 SE Authenticators are made, then they all must have the exact same Attestation Key.
	(For Enterprise only) There is no Correlation Handle (KeyID/CredentialID, Key Handle) that is visible across multiple Relying Parties, except the unique identifier present in the Enterprise Attestation Certificate or the Enterprise Attestation Certificate itself.
	(For Consumer only) A FIDO2 SE Authenticator in Consumer configuration shall not provide any functionality related with Enterprise Attestation in the end-user phase. That is, all firmware supporting Enterprise Attestation shall be disabled in the end-user phase for FIDO2 SE Authenticators in Consumer configuration. Remark: Enterprise Attestation functionality can only be provisioned/configured by the Vendor or its delegates in an Enterprise Authenticator.
	The Attestation Certificate / ECDAA Issuer public keys shall be dedicated to a single Authenticator model.
	The FIDO2 SE Authenticator shall fulfil at least the specific security characteristics stated for its model.
	A FIDO2 SE Authenticator that supports Enterprise Attestation and that inserts a unique identifier in its Enterprise Attestation Certificate shall use a unique private key per identifier.
	(For Enterprise) If the Vendor configures the RPIDs list before delivery, the Vendor must check that the provided list of RPIDs is owned by the Customer or the Customer's Data Processors as defined by the GPDR.
	Authenticator Security Parameters that are cryptographic keys generated during manufacturing shall be generated as required by the standard referenced in the "Allowed Cryptography List" [FIDOAllowedCrypto1-2] for that algorithm using an Allowed Random Number Generator.
ALC_TAT.1	NA
ATE_COV.2	NA
ATE_DPT.1	NA
ATE_FUN.1	NA
ATE_IND.2	NA

SARs	Refinements
AVA_VAN.5	The Authenticator shall resist brute-force attacks of its user verification methods other than user presence check. The evaluator shall assess the robustness of the rate-limit mechanism. The evaluator shall assume that an attacker can try all possible input combinations (e.g. passwords, PINs, patterns, biometrics...) in order to pass the user verification. In the case of biometric user verification, the attacker can bring a potentially unlimited number of "friends" that can try whether their biometric characteristic is accepted (as false accept). In all cases the number of trials per time is limited by the verification speed of the authenticator. Remark: The violation of the integrity of the authenticator (e.g. decapping of chips, malware, ...) is covered in tamper-resistance tests.

7.3 Security Requirements Rationale

7.3.1 Rationale for the SFRs

The security objectives for the TOE are covered by the SFRs as shown in Table 7-4 and Table 7-5.

Table 7-4: Mapping security objectives for the TOE and SFRs – Part 1

Objectives	FCS_CKM.1	FCS_CKM.4	FCS_CKM.5/KD	FCS_CKM.5/SG	FCS_COP.1	FCS_RNG.1	FDP_ETC.2	FDP_ITC.1	FDP_RIP.1	FDP_SDI.1	FIA_AFL.1 (Optional)	FIA_UAU.5	FIA_UAU.6	FDP_IFC.1	FDP_IFF.1	FMT_MOF.1	FMT_MSA.1	FMT_MSA.3
O.ALLOWED_CRYPTOGRAPHY	X	X	X	X	X													
O.ASP_PROTECTION							X			X								
O.ATTESTABLE_PROPERTIES																		
O.CORRECT_KEY_AND_SIGNATURE_GENERATION	X		X	X														
O.DISABLED_DEBUG																		
O.ENTERPRISE_ATTESTATION (for Enterprise)														X	X			
O.ENTERPRISE_ATTESTATION_PROVISIONING (for Enterprise)																X		
O.FACTORY_RESET (Optional)									X									
O.FUNCTIONALITY_PROTECTION								X										
O.IMPERSONATION_RESILIENCE																		
O.LEAKAGE_RESISTANCE									X									
O.PIN_DATA_PROTECTION							X			X								
O.RNG						X												
O.RP_IDENTIFIERS_LIST_MANAGEMENT																		
O.TAMPER_RESISTANCE																		
O.TRUSTWORTHY_DATA																		

O.UNLINKABLE_LIMITED_PII																		
O.USER_CONSENT														X	X		X	X
O.USER_PRESENCE_CHECK												X	X	X	X		X	X
O.USER_VERIFICATION (Optional)											X	X	X	X	X		X	X

Table 7-5: Mapping security objectives for the TOE and SFRs – Part 2

Objectives	FMT_MTD.1/ASP_PIN	FMT_MTD.1/CONFIG	FMT_MTD.1/DEBUG	FMT_MTD.1/PROPERTIES	FMT_MTD.1/RP_LIST	FMT_MTD.2/ASP_PIN	FMT_MTD.3/KH	FMT_SMF.1	FMT_SMR.1	FPR_ANO.2	FPR_UNL.1/NO_INFERENCE1	FPR_UNL.1/NO_INFERENCE2	FPR_UNL.1/NO_INFERENCE3	FPR_UNL.1/ID	FPR_UNL.1/Info	FPT_EMS.1	FPT_FLS.1	FPT_PHP.3	FPT_RPL.1	FTP_TRP.1
O.ALLOWED_CRYPTOGRAPHY																				
O.ASP_PROTECTION	X					X														X
O.ATTESTABLE_PROPERTIES				X					X											
O.CORRECT_KEY_AND_SIGNATURE_GENERATION							X										X			
O.DISABLED_DEBUG			X																	
O.ENTERPRISE_ATTESTATION (for Enterprise)																				
O.ENTERPRISE_ATTESTATION_PROVISIONING (for Enterprise)					X				X											
O.FACTORY_RESET (Optional)								X												
O.FUNCTIONALITY_PROTECTION		X							X								X			
O.IMPERSONATION_RESILIENCE	X					X	X												X	
O.LEAKAGE_RESISTANCE																X				

O.PIN_DATA_PROTECTION	X					X													X
O.RNG																			
O.RP_IDENTIFIERS_LIST_MANAGEMENT					X				X										
O.TAMPER_RESISTANCE																	X		
O.TRUSTWORTHY_DATA							X									X			
O.UNLINKABLE_LIMITED_PII										X	X	X	X	X	X				
O.USER_CONSENT																			
O.USER_PRESENCE_CHECK																			
O.USER_VERIFICATION (Optional)																			

Note that FPT_EMS.1 and FPT_PHP.3 are mentioned in relationship with O.LEAKAGE_RESISTANCE and O.TAMPER_RESISTANCE respectively but contribute to most of the objectives.

O.ALLOWED_CRYPTOGRAPHY: The objective is covered by the conjunction of the following security functional requirements:

- FCS_CKM.1, which enforces the conformance with [FCrypto] for the generation of keys
- FCS_CKM.4, which enforces the use of standard means of key destruction
- FCS_CKM.5/KD, which enforces the conformance with [FCrypto] for the derivation of keys
- FCS_CKM.5/SG, which enforces the conformance with [FCrypto] for the generation of signatures
- FCS_COP.1, which enforces the conformance with [FCrypto] for any cryptographic operations.

O.ASP_PROTECTION: The objective is covered by the conjunction of the following security functional requirements:

- FDP_ETC.2, which enforces the confidentiality protection of exported ASPs
- FDP_SDI.1, which enforces the detection of ASPs integrity loss
- FMT_MTD.1/ASP_PIN and FMT_MTD.2/ASP_PIN, which control the operations on specific ASPs including PIN limit and signature counters
- FTP_TRP.1, which enforces the integrity and confidentiality of ASPs that are communicated to external entities.

O.ATTESTABLE_PROPERTIES: The objective is directly covered by the conjunction of the following security functional requirements: FMT_MTD.1/PROPERTIES and FMT_SMR.1.

O.CORRECT_KEY_AND_SIGNATURE_GENERATION: The objective is covered by the conjunction of the following security functional requirements:

- FCS_CKM.1, FCS_CKM.5/KD and FCS_CKM.5/SG, which enforce the conformance with [FCrypto] for key generation, key derivation and signature generation
- FMT_MTD.3/KH and FPT_FLS.1, which ensure that signature keys are bound to the Authenticator and fails securely otherwise.

O.DISABLED_DEBUG: The objective is directly covered by the security functional requirement FMT_MTD.1/DEBUG.

O.ENTERPRISE_ATTESTATION (for Enterprise): The objective is directly covered by the conjunction of the following security functional requirements: FDP_IFC.1 and FDP_IFF.1.3 item g).

O.ENTERPRISE_ATTESTATION_PROVISIONING (for Enterprise): The objective is covered by the conjunction of the following security functional requirements:

- FMT_MOF.1, which ensures that only the Vendor can enable the Enterprise Attestation functionality
- FMT_MTD.1/RP_LIST, which ensures that only the Vendor can manage the RP_List
- FMT_SMR.1, which enforces the Vendor role.

O.FACTORY_RESET (Optional): The objective is covered by the conjunction of the following security functional requirements:

- FDP_RIP.1, which enforces the unavailability of sensitive information upon factory reset
- FMT_SMF.1, which directly supports factory reset.

O.FUNCTIONALITY_PROTECTION: The objective is covered by the conjunction of the following security functional requirements:

- FDP_ITC.1, which enforces input validation
- FMT_MTD.1/CONFIG, which enforces modification of the security characteristics by the Vendor only
- FMT_SMR.1, which enforces the Vendor role
- FPT_FLS.1, which enforces secure failure in the presence of abuse of functionality and data input verification.

O.IMPERSONATION_RESILIENCE: The objective is covered by the conjunction of the following security functional requirements:

- FMT_MTD.1/ASP_PIN and FMT_MTD.2/ASP_PIN, which enforces the use of monotonic signature counters
- FMT_MTD.3/KH, which enforces the use of keys bound to the Authenticator and the RP
- FPT_RPL.1, which enforces replay detection.

O.LEAKAGE_RESISTANCE: The objective is covered by the conjunction of the following security functional requirements:

- FDP_RIP.1, which enforces the unavailability of sensitive information upon deallocation (including factory reset)
- FPT_EMS.1, which enforces protection against disclosure of confidential information through emanations.

O.PIN_DATA_PROTECTION: The objective is covered by the conjunction of the following security functional requirements:

- FDP_ETC.2, which enforces the confidentiality protection of exported PIN and PIN related data
- FDP_SDI.1, which enforces the detection of PIN and PIN related data integrity loss
- FMT_MTD.1/ASP_PIN and FMT_MTD.2/ASP_PIN, which control the operations on PIN limit and signature counters
- FTP_TRP.1, which enforces the integrity and confidentiality of PIN and PIN related data that are communicated to external entities.

O.RNG: The objective is directly covered by the security functional requirement FCS_RNG.1.

O.RP_IDENTIFIERS_LIST_MANAGEMENT (for Enterprise): The objective is directly covered by the following security functional requirements: FMT_MTD.1/RP_LIST and FMT_SMR.1.

O.TAMPER_RESISTANCE: The objective is directly covered by the security functional requirement FPT_PHP.3.

O.TRUSTWORTHY_DATA: The objective is directly covered by the conjunction of the following security functional requirements: FMT_MTD.3/KH and FPT_FLS.1.

O.UNLINKABLE_LIMITED_PII: The objective is covered by the conjunction of the following security functional requirements:

- FPR_ANO.2, which protects the personal information that is conveyed in a communication with an RP
- FPR_UNL.1/NO_INFERENCE1, which prevents from establishing a link between separate conversations with a user
- FPR_UNL.1/NO_INFERENCE2, which ensures that the same information cannot be used for communicating with different RPs
- FPR_UNL.1/NO_INFERENCE3, which prevents from determining if the same Authenticator is used by different accounts
- FPR_UNL.1/ID and FPR_UNL.1/Info, which enforces the protection of the information revealed to third parties through user consent, user verification or credential ID.

O.USER_CONSENT: The objective is covered by the conjunction of the following security functional requirements:

- FDP_IFC.1 and FDP_IFF.1, which directly enforce the objective (cf. FDPIFF.1.3a))
- FMT_MSA.1 and FMT_MSA.3, which enforce the management of the security attributes of the policy FIDO2 Authenticator SFP.

O.USER_PRESENCE_CHECK: The objective is covered by the conjunction of the following security functional requirements:

- FIA_UAU.5, which enforces the support of user presence check
- FIA_UAU.6, which enforces the conditions for the renewal of user presence check
- FDP_IFC.1 and FDP_IFF.1, which enforce the rules of the CTAP specification
- FMT_MSA.1 and FMT_MSA.3, which support the management of the security attributes of the policy FIDO2 Authenticator SFP.

O.USER_VERIFICATION (Optional): The objective is covered by the conjunction of the following security functional requirements:

- FIA_AFL.1, which sets a limit to user verification failures
- FIA_UAU.5, which enforces the support of user verification
- FIA_UAU.6, which enforces the conditions for the renewal of user verification
- FDP_IFC.1 and FDP_IFF.1, which enforce the rules of the CTAP specification
- FMT_MSA.1 and FMT_MSA.3, which support the management of the security attributes of the policy FIDO2 Authenticator SFP.

7.3.2 Rationale for the SARs

EAL4 augmented with AVA_VAN.5 and ALC_DVS.2 is required for this type of TOE to protect against sophisticated attacks, and to resist attackers with high attack potential.

7.3.2.1 AVA_VAN.5 Advanced Methodical Vulnerability Analysis

The TOE is intended to operate in hostile environments. AVA_VAN.5 “Advanced methodical vulnerability analysis” is considered as the expected level for Java Card/GlobalPlatform technology-based products hosting sensitive applications, such as FIDO2 Authenticator Application. AVA_VAN.5 has dependencies on ADV_ARC.1, ADV_FSP.1, ADV_TDS.3, ADV_IMP.1, AGD_PRE.1, and AGD_OPE.1. All these assurance requirements are met by EAL4.

7.3.2.2 ALC_DVS.2 Sufficiency of Security Measures

Development security is concerned with physical, procedural, personnel, and other technical measures of the development environment. Due to the sensitivity of the TOE, it is necessary to justify the sufficiency of the development environment measures to protect the integrity and confidentiality of the TOE up to the delivery. This is achieved through ALC_DVS.2, which has no dependencies.

7.3.3 Dependencies

7.3.3.1 SFRs Dependencies

Table 7-6 presents the dependencies of the SFRs defined in this PP that are satisfied.

Table 7-6: SFRs Dependencies

Requirements	CC Dependencies	Satisfied Dependencies
FCS_CKM.1	No dependencies	
FCS_CKM.4	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1)	FCS_CKM.1
FCS_CKM.5	(FCS_CKM.2 or FCS_COP.1) and (FCS_CKM.4)	FCS_COP.1 and FCS_CKM.4
FCS_COP.1	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and (FCS_CKM.4)	FCS_CKM.1 and FCS_CKM.4
FCS_RNG.1	No dependencies	
FDP_ETC.2	(FDP_ACC.1 or FDP_IFC.1)	FDP_IFC.1
FDP_IFC.1	FDP_IFF.1	FDP_IFF.1
FDP_IFF.1	(FDP_IFC.1) and (FMT_MSA.3)	FDP_IFC.1 and FMT_MSA.3
FDP_ITC.1	(FDP_ACC.1 or FDP_IFC.1) and (FMT_MSA.3)	FDP_IFC.1 and FMT_MSA.3
FDP_RIP.1	No dependencies	
FDP_SDI.1	No dependencies	
FIA_AFL.1	FIA_UAU.1	The dependency FIA_UAU.1 is fulfilled through FDP_IFC.1 and FDP_IFF.1 where the user verification rules are specified.
FIA_UAU.6	No dependencies	
FMT_MOF.1	(FMT_SMR.1) and (FMT_SMF.1)	FMT_SMR.1
FMT_MSA.1	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_IFC.1
FMT_MSA.3	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1
FMT_MTD.1	(FMT_SMR.1) and (FMT_SMF.1)	FMT_SMR.1
FMT_MTD.2	(FMT_MTD.1) and (FMT_SMR.1)	FMT_MTD.1 and FMT_SMR.1
FMT_SMF.1	No dependencies	

Requirements	CC Dependencies	Satisfied Dependencies
FMT_SMR.1	FIA_UID.1	
FPR_ANO.2	No dependencies	
FPR_UNL.1	No dependencies	
FPR_UNO.1	No dependencies	
FPT_RPL.1	No dependencies	
FPT_EMS.1	No dependencies	
FPT_FLS.1	No dependencies	
FPT_PHP.3	No dependencies	
FTP_TRP.1	No dependencies	

The dependencies FMT_SMF.1 and FMT_SMR.1 of FMT_MSA.1 are discarded since no special security management function or role are necessary.

The dependency FMT_SMF.1 of FMT_MOF.1 is discarded since no special security management function is necessary.

The dependency FMT_SMF.1 of FMT_MTD.1 is discarded since no special security management function is necessary.

The dependency FMT_SMR.1 of FMT_MSA.3 is discarded since no special role is necessary.

The dependency FIA_UID.1 of FMT_SMR.1 is left to the ST author.

7.3.3.2 SARs Dependencies

Table 7-7 presents the dependencies of the SARs defined in this PP and how they are satisfied.

Table 7-7: SARs Dependencies

SARs	CC Dependencies	Satisfied Dependencies
ADV_ARC.1	(ADV_FSP.1) and (ADV_TDS.1)	ADV_FSP.4, ADV_TDS.3
ADV_FSP.4	(ADV_TDS.1)	ADV_TDS.3
ADV_IMP.1	(ADV_TDS.3) and (ALC_TAT.1)	ADV_TDS.3, ALC_TAT.1
ADV_TDS.3	(ADV_FSP.4)	ADV_FSP.4
AGD_OPE.1	(ADV_FSP.1)	ADV_FSP.4
AGD_PRE.1	No Dependencies	
ALC_CMC.4	(ALC_CMS.1) and (ALC_DVS.1) and (ALC_LCD.1)	ALC_CMS.4, ALC_DVS.2, ALC_LCD.1
ALC_CMS.4	No Dependencies	
ALC_DEL.1	No Dependencies	
ALC_DVS.2	No Dependencies	
ALC_FLR.2	No Dependencies	
ALC_LCD.1	No Dependencies	

SARs	CC Dependencies	Satisfied Dependencies
ALC_TAT.1	(ADV_IMP.1)	ADV_IMP.1
ASE_CCL.1	(ASE_ECD.1) and (ASE_INT.1) and (ASE_REQ.1)	ASE_ECD.1, ASE_INT.1, ASE_REQ.2
ASE_ECD.1	No Dependencies	
ASE_INT.1	No Dependencies	
ASE_OBJ.2	(ASE_SPD.1)	ASE_SPD.1
ASE_REQ.2	(ASE_ECD.1) and (ASE_OBJ.2)	ASE_ECD.1, ASE_OBJ.2
ASE_SPD.1	No Dependencies	
ASE_TSS.1	(ADV_FSP.1) and (ASE_INT.1) and (ASE_REQ.1)	ADV_FSP.4, ASE_INT.1, ASE_REQ.2
ATE_COV.2	(ADV_FSP.2) and (ATE_FUN.1)	ADV_FSP.4, ATE_FUN.1
ATE_DPT.1	(ADV_ARC.1) and (ADV_TDS.2) and (ATE_FUN.1)	ADV_ARC.1, ADV_TDS.3, ATE_FUN.1
ATE_FUN.1	(ATE_COV.1)	ATE_COV.2
ATE_IND.2	(ADV_FSP.2) and (AGD_OPE.1) and (AGD_PRE.1) and (ATE_COV.1) and (ATE_FUN.1)	ADV_FSP.4, AGD_OPE.1, AGD_PRE.1, ATE_COV.2, ATE_FUN.1
AVA_VAN.5	(ADV_ARC.1) and (ADV_FSP.4) and (ADV_IMP.1) and (ADV_TDS.3) and (AGD_OPE.1) and (AGD_PRE.1) and (ATE_DPT.1)	ADV_ARC.1, ADV_FSP.4, ADV_IMP.1, ADV_TDS.3, AGD_OPE.1, AGD_PRE.1, ATE_DPT.1