Federal Office
for Information Security

Common Criteria Security Target de.fac2 – FIDO U2F Authenticator Applet, v1.34

v1.25 (Jun 14th, 2022)

Federal Office for Information Security
Post Box 20 03 63
D-53133 Bonn

Internet: https://www.bsi.bund.de

Table of Contents

1 ST Introduction

This document defines the security functionality of the target of evaluation (TOE) “de.fac2 - FIDO U2F Authenticator Applet, v1.34” as a security target (ST) that is conformant to common criteria and the protection profile [FIDOPP].

Changes of the [FIDOPP] made by the ST author are marked in blue fonts.

1.1 ST Reference

Title
Common Criteria Security Target de.fac2 - FIDO U2F Authenticator Applet, v1.34

Version Number
Version 1.25 (Jun 14th, 2022)

CC Version
3.1 (Revision 5)

General Status
final

TOE
de.fac2 - FIDO U2F Authenticator Applet, v1.34

Javacard OS Platform
Sm@rtCafé® Expert 7.0 C3, BSI-DSZ-CC-1028-2017 [SCST] (certification date 08.09.2017), configuration 1
Assurance Continuity Maintenance Report, BSI-DSZ-CC-1028-2017-MA-01 [MA-SCST] (certification date 04.10.2018)

Security Controller
IFX M5073 G11, BSI-DSZ-CC-0951-2015-RA-01 [ICST] (certification date 31.05.2017)

TOE Documentation
Administration and User Guide [GUIDANCE]

Assurance Level
Minimum assurance level for this ST is EAL4 augmented.

Registration
BSI-DSZ-CC-1060

Sponsor and Developer
Bundesamt für Sicherheit in der Informationstechnik (BSI)

Keywords
FIDO, U2F, Applet

1.2 TOE Overview

1.2.1 TOE Definition and Operational Use

The TOE (de.fac2 [deɪˈfæktəʊ]) addressed by the current Security Target (ST) is a FIDO Authenticator intended for FIDO Universal Second Factor (U2F) authentication [FIDOSpec]. In this ST a specific implementation of the U2F token is considered. The authenticator is physically implemented as a security chip with an application and is used to securely access online services. The authenticator, also referred to as a „U2F token“ (or just „token“), communicates with an external server controlled by a relying party (RP) that supports the standardised FIDO U2F protocol.

The main goal of U2F based authentication is to provide a strong second-factor authentication mechanism for web-users while preserving the user's privacy. The second factor is the U2F authenticator, carried by the user. A U2F authenticator thus augments the security of the commonly used user name and password mechanism.

To authenticate himself towards the RP, the user has to prove his presence by interacting with the authenticator (e.g. by pressing some button or placing the token into the proximity range of an NFC-enabled device) or by entering a PIN code[1]. The TOE then uses cryptographic material, namely a private key from an asymmetric key pair that is generated and used by the token, to log in. Key pairs are unique for each tuple of relying party, user account and U2F authenticator, thus preventing to trace the user over different relying parties and accounts.

1.2.2 TOE Major Security Features

The following security goals are met by a certified device and thus describe the essential security features of the TOE:

- **Strong Authentication:** The TOE authenticates a user and/or a device 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 U2F protocol, that could be potentially used to link two accounts to the same user are 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 authenticator does not store or require to associate any personal information with the identity of the user.
- **User Presence:** The U2F device has a physical test of user presence to ensure that the relying party can trust in that the authentication process is actively triggered by the user himself.

Note: Unlinkability and Privacy are only ensured, if not any other application are operated on the same authenticator.

Note: These security goals stem from FIDO U2F. Within the design of FIDO U2F, several assumptions are made regarding the security chracteristics of the operating environment components on which a FIDO implementation depends. The reader is referred to [FIDOSecRef] for a detailed discussion on the implications of the FIDO U2F design and these security goals.

1 In case of the TOE, the proof of presence is demonstrated disconnecting the TOE from power, for example by placing the token into the proximity range of an NFC-enabled device, by (re)-inserting the card into the card reader, or by programmatically sending a reset command to the card.

1.2.3 TOE Type

The TOE is a dual-interface secure chip including all IC dedicated software, embedded in an arbitrary housing with embedded software including the operating system and an application for FIDO U2F authentication.

The TOE requires a client application (e.g. a web-browser) to interact with the relying party. Neither the client application nor the relying party are part of the TOE.

1.2.4 TOE Life Cycle

This Security Target describes security objectives and requirements for an authenticator supporting the FIDO U2F protocol. The life-cycle and its order is viewed from a logical perspective on product development. As this is a composite evaluation, the exhaustive guidance of [ICPP] was taken into account when defining the TOE life-cycle.

Stage 1: Development

Step 1 - IC Development (ICPP Phase2)

The IC developer develops the TOE in phase 1. This includes the IC design, the IC dedicated software and the guidance documentation associated with these TOE components.

Step 2a - Security IC Embedded Software Development (ICPP Phase 1)

The OS developer uses the guidance documentation for the integrated circuit and the guidance documentation for relevant parts of the IC dedicated software, and develops the IC embedded software (operating system), and the guidance documentation associated with this TOE component. The manufacturing documentation of the IC including the IC dedicated software and the embedded software to be stored in the non-volatile non-programmable memories is securely delivered to the IC manufacturer. The IC embedded software to be stored in the non-volatile programmable memories, the application(s), and the guidance documentation is securely delivered to the chip manufacturer.

Step 2b - Applet Development (ICPP Phase 1)

The application developer uses the Guidance of the Java Card platform to develop the FIDO U2F application and the guidance documentation associated with this TOE component.

Stage 2: Manufacturing

Step 3 – IC Manufacturing (ICPP Phase 3)

This step includes the integration and the IC production.

Step 4 – IC Packaging (ICPP Phase 4)

The IC is packaged and combined with hardware for the dual interface.

Step 5 - Composite product integration (ICPP Phase 5)

The software developer delivers the FIDO U2F application, the attestation certificate and the private key of the certificate to the composite product integrator. The Java Card is then combined with the FIDO U2F application into a composite product by the composite product integrator.

The FIDO U2F applet is uploaded to the card via the GlobalPlatform card manager. During the upload and installation process the private key for the FIDO attestation certificate will be transferred and stored into the card. The FIDO U2F applet is then in the initalisiation state and only accepts proprietary APDUs to store the attestation certificate in the card. After the attestation certificate upload is completed, the card generates card individual keys for the seed and the MACKey and switches to the operational state. In the operational state the card refuses the proprietary APDU for uploading the attestation certificate and only accept APDUs which are used for the FIDO U2F functionality specified in [FIDOSpec] and the APDU for the reset process.

This is the finishing process and the composite product integrator delivers the TOE for operational use. The TOE is now in state ready_for_use.

Step 6 – Personalisation (ICPP Phase 6)

The life-cycle of the composite TOE includes no personalisation steps.

Stage 3: Operational Use

Step 7 – Operational Use (ICPP Phase 7)

The end-user may command the token to (re-)seed and to generate the MACKey that is stored in the TOE during the initialisation process. For each registration a new nonce has to be generated. The nonce is then used to compute the private key:

nonce = RNG()
PrivK = KDF(seed, AppID, nonce)

If the end-user resets the authenticator, the TOE is in delivery_state. Because subsequently new cryptographic key material has to be generated, the de.fac2 Applet triggers immediately the initialisation step for seed and MACKey generation. So all previously-done registrations are permanently unusable for future authentication. A reset can be performed at any time by the end-user.

“TOE Delivery“ in this ST indicates delivery after Step 5 if the TOE is delivered as a composite product (cf. [ICPP] Figure 2: Definition of “TOE Delivery” and responsible Parties). TOE Delivery in this case means, the FIDO U2F Authenticator as a Composite Product is already delivered in state ready_for_use after step 5 (see Figure 1).

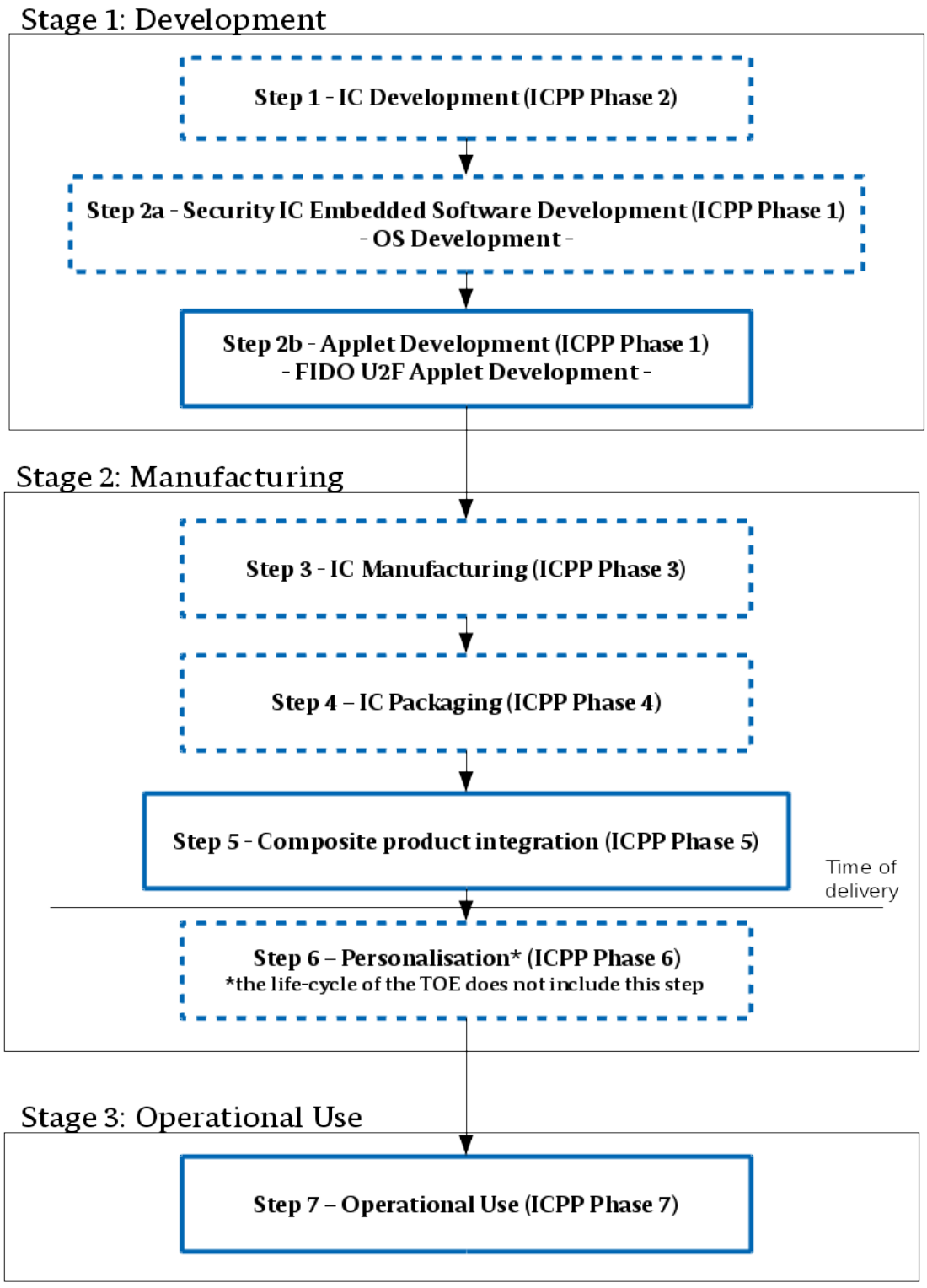


Figure 1: TOE Life-Cycle[2]

2 The steps that are not evaluated again but included through certificates are indicated by a dashed line.

1.3 TOE Description

1.3.1 Physical Scope

The TOE and delivery scope consists of the following parts:

- The hardware platform IFX M5073 G11 (Certificate: BSI-DSZ-CC-0951-2015, including re-assessment BSI-DSZ-CC-0951-2015-RA-01, [ICCR-RA]). with the following configurations according to [ICST]:
  - FLASH: up to 628 kByte
  - ROM: not available
  - RAM for the user: 1-12 kByte
  - SCP (Symmetric Crypto Co-processor for DES and AES Standards): accessible
  - Crypto2304T (Crypto Co-processor for asymmetric algorithms like RSA and EC): accessible
  - Interfaces: ISO/IEC 7816 and/or ISO/IEC 14443
- The Sm@rtCafé® Expert 7.0 C3 OS (Certificate: BSI-DSZ-CC-1028-2017 [SCCR], BSI-DSZ-CC-1028-2017-MA-01 [MA-SCST]) with the following configurations according to [SCST]):
  - Java Card Runtime Environment (JCRE)
  - Java Card Virtual Machine (JCVM)
  - Java Card API
  - On-card Installer
  - Applet Deletion Manager
  - Card Manager
  - Smart Card OS including the G&D crypto library
  - Not supported: Java Card Remote Method Invocation (JCRMI)
- The applet for FIDO U2F authentication and it's documentation [GUIDANCE].

The smart card with the loaded applet is delivered to the end user by the Composite Product Integrator via post. The digital user guidance [GUIDANCE] is to be downloaded in PDF format from the BSI website. The user has to verify the SHA-512 checksum as given in the certification report for this TOE, to validate the originality and integrity of the guidance documentation.

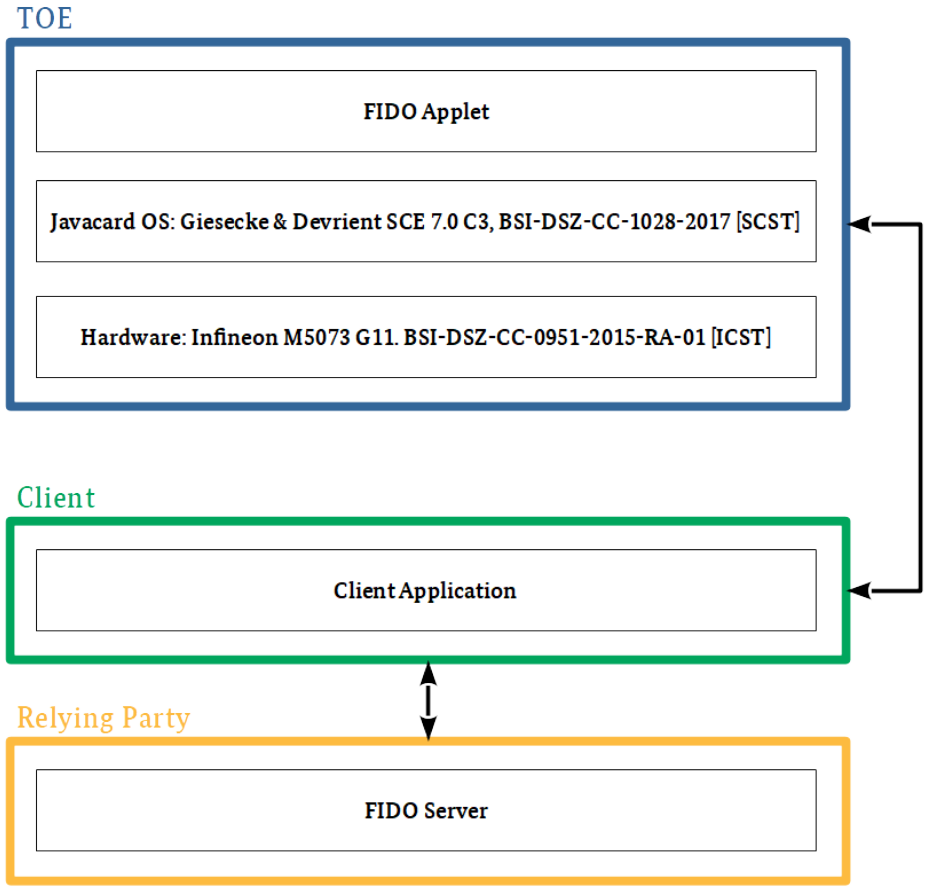


Figure 2: TOE Boundary and the Operational Environment

1.3.2 Logical Scope

As a FIDO Authenticator token intended for FIDO Universal Second Factor (U2F) authentication, the TOE communicates with an external server controlled by a relying party (RP).

The two main steps that have to be performed as part of this communication are the registration process and the authentication process. In the following, we assume the existence of a *seed* and a *secret key for MAC computation (MACKey)* on the token, whereupon seed and MACKey have to be different. The MACKey and the seed are generated in the initialisation process either during the first usage of the authenticator by the end-user or during the initialisation process at the manufacturer. These two options are the only permitted options to initialize the key material. The life cycle of this ST provides the generation of the key material at the manufacturer. Seed and MACKey are stored securely on the device. The user can also afterwards reset the device and thus destruct all existing key material. After that, the initialisation process has to be repeated before the next usage.

During manufacturing, the manufacturer loads an externally generated attestation key pair and an attestation certificate onto the device. The attestation certificate refers to the public attestation key and is signed usually by a certification authority selected by the manufacturer. The Attestation key serves as a trust anchor for the authenticity of the FIDO Authenticator to the relying party.

Apart from its use in the Registration process, all other security mechanisms and requirements applying to the attestation key pair and the attestation certificate according to [FIDOSpec] are completely outside the scope of the present version of this PP. It is planned to cover them in a later PP version.

During the registration process, a site/application/account-specific asymmetric key pair is generated. First, the relying party submits the AppID (a 32 byte value) to the authenticator. The authenticator then generates a random nonce. Then a private key is generated using a key derivation function (KDF). The input to the KDF is the seed, the AppID as well as the previously generated fresh nonce. The public asymmetric key pair is generated from the private key using the underlying asymmetric cryptographic algorithm. The authenticator proceeds to generate a *key handle* that allows the authenticator later on to recover the generated key pair in the authentication phase. The key handle consists of the generated nonce, as well as a message authentication code (MAC) over the AppID and nonce using the MACKey. The key handle, the AppID, 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 is then transmitted to and stored at the relying party.

During authentication, the authenticator receives (via the user device) the AppID, the previously registered key handle as well as a challenge. The authenticator verifies the message authentication code and thus checks whether the supplied key handle contains a nonce generated by the authenticator that fits to the supplied AppID. After successful verification, the KDF is activated with the seed together with the AppID and nonce to re-generate the private key. Then, the challenge as well as the AppID are signed with the private key, and the challenge together with the signature is sent to the relying party. After successful verification of the signature with the public key of the authenticator (which was stored during registration at the relying party), the relying party can conclude that indeed the user authenticated himself to the service.

The four steps that can be handled by the authenticator (initialisation, reset, registration and authentication) are visualised in figure 3 to 5. These pseudo-code descriptions are purely informative. Relevant details for the technical implementation can be found in the security requirements chapter 5.1.

This description omits some parts from the specification, e.g. the user-device in between the relying party and the token, as well as some implementation-specific details. For a full specification of the FIDO standard we refer to [FIDOSpec].

During the life-cycle of the TOE, an attestation key is loaded on the device. Attestation keys serve as trust anchor for the authenticity of FIDO Authenticators to the relying party. The attestation certificates are not in the scope of the TOE.

```
          Initialisation Process
U2F Authenticator                 Relying Party

if (user_presence == true) and (security_state == delivery_state) {
 seed = RNG()
 MACKey = RNG()
 security_state = ready_for_use
}
manage_user_presence()
```

Figure 3: Process description of the initialisation step

```
          Reset Process
U2F Authenticator                 Relying Party

if (user_presence == true) and (security_state == ready_for_use) {
 security_state = delivery_state
 secure_erase(seed)
 secure_erase(MACKey)
}
manage_user_presence()
```

Figure 4: Process description of the reset step

```
          Registration Process
U2F Authenticator                 Relying Party
                    <-------- App ID, challenge ---------
if (user_presence == true) and (security_state == ready_for_use){
 nonce = RNG()
 PrivK = KDF(seed, AppID, nonce)
 PubK = genKeyPair(PrivK)
 KeyHandle = nonce | MAC(MACKey, AppID | nonce)
 sig = sign(AttestationPrivK, KeyHandle | challenge | PubK | AppID)
 secure_erase(PrivK)
           ------ Key Handle, PubK, AttestationCert, sig ----->
 delete(PubK)
                                      verify(AttestationCert.AttestationPubK,
                                                    KeyHandle | challenge |
                                                          PubK | AppID, sig)
}
manage_user_presence()
```

Figure 5: Process description of the registration step

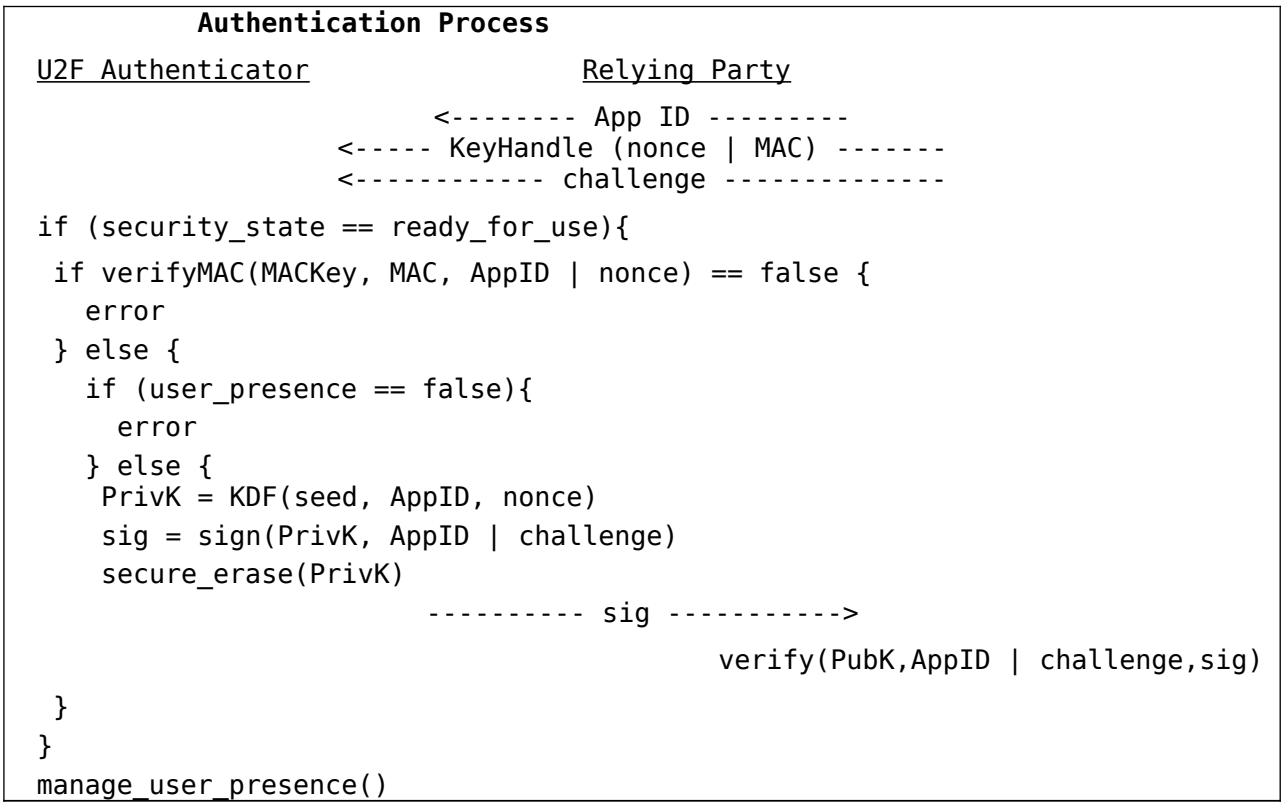


Figure 6: Process description of the authentication step

2 Conformance Claims

2.1 CC Conformance Claim

This Security Target claims conformance to

Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model; CCMB-2017-04-001, Version 3.1, Revision 5, April 2017 [CC1]

Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components; CCMB-2017-04-002, Version 3.1, Revision 5, April 2017 [CC2]

Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components; CCMB-2017-04-003, Version 3.1, Revision 5, April 2017 [CC3]

as follows

Part 2 extended,

Part 3 conformant.

The

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

has to be taken into account.

2.2 PP Claim

This ST claims strict conformance to [FIDOPP].

2.3 Package Claim

The ST is conformant to the following packages:

Assurance package EAL4 augmented with AVA_VAN.5 as defined in [CC3].

2.4 Conformance Rationale

This ST claims strict conformance to only one PP, the PP [FIDOPP].

The TOE Type of this ST claims conformance to the TOE type of the PP [FIDOPP] by being identical.

The Security Problem Definition, Security Objectives and Security Requirements are identical to the PP [FIDOPP].

The roles in Chapter 3.1.2 are identical to the PP [FIDOPP].

The following Table 2.1 demonstrates the realization of the Application Notes of the PP [FIDOPP] in this ST.

No.	SFR	Application Note	Implemenation
1	FCS_CKM.1	The SFR above applies to all cryptographic key generation methods implemented on the TOE.	Kept as is.
2	FCS_COP.1	The above SFR applies to all cryptographic operations implemented on the TOE.	Operations performed for signature creation

No.	SFR	Application Note	Implemenation
			and application note removed.
3	FCS_RNG.1	The above SFR requires the TOE to generate random numbers used for seed, nonce and MACKey generation.	Kept as is.
4	FCS_CKM.5/ U2F-Private	The SFR implements the process of generating random bits in the FIDO authentication process. The generated random bits of the FIDO compliant KDF are used to derive FIDO private keys from the global seed.	Kept as is.
5	FDP_IFF.1	The function manage_user_presence() (re)sets the user presence value according to the requirements defined in FIA_UAU.2 and FIA_UAU.6.	Kept as is.
6	FDP_IFF.1	FDP_IFF.1.4 is left open here and in subsequent SFRs FDP_IFF.* and is intended to be used when there is a need to cater to specific implementational details. Assignments here must not circumvent the explicit rules stated in FDP_IFF.1.3 and must not weaken the security requirements. The ST-Writer should assign “none” here if no such specific implementational detail has to be considered.	“None” assigned. Application note removed.
7	FDP_SDI.1	Assignment in the above SFR should be filled by the ST-writer and very much depend on the specific implementation. For example, integrity errors could be manipulated errors in RAM during execution.	Operations performed and application note removed.
8	FIA_UAU.2	Proof of presence means e.g. pressing some button or placing the token into the proximity range of an NFC-enabled device.	Added footnote to clarify TOE-specific implementation..
9	FIA_UAU.6	"Possible conditions are: 1) the elapsed time since the previous proof of presence demonstration has surpassed a limit (specified in the condition) 2) an event (specified in the condition) has occurred. The list of conditions must not be empty."	Listed conditions that require re-authentication. Application note removed.
10	FMT_SMF.1	The list of other security management functions to be provided by the TSF may be empty (i.e. assignment “none”).	“None” assigned. Application note removed.
11	FMT_MSA.3/ Initialisation	Restrictive means: user_presence = false and security_state = delivery_state	Kept as is.
12	FMT_MSA.3/ Reset	Restrictive means: user_presence = false and security_state = ready_for_use	Kept as is.
13	FMT_MSA.3/ Registration	Restrictive means: user_presence = false	Kept as is.
14	FMT_MSA.3/ Authentication	Restrictive means: user_presence = false and MAC verification status = false	Kept as is.
15	FPT_TST.1	Authorised user means user has demonstrated his proof of presence.	Kept as is.

Table 2.1: Realization of the application notes.

3 Security Problem Definition

3.1 Assets, subjects and threat agents

3.1.1 Assets

Object No.	Asset	Definition
1	seed	The seed is once generated by the TOEs RNG. For this purpose the end-user has to execute a command to start the generation during the initialisation process. The seed is used to compute the private key and must never leave the TOE. In case of a reset, initiated by the end-user, the seed will be deleted. A new seed is generated by the RNG of the token if the end-user starts the initialisation process again.
2	MACKey	The MACKey is once generated by the TOEs RNG. For this purpose the end-user has to execute a command to start the generation during the initialisation process. The MACKey is used for MAC generation; the MAC itself is part of the key handle. The key must never leave the TOE. In case of a reset, initiated by the end-user, the MACKey will be deleted. A new MACKey is generated by the RNG of the token if the end-user starts the initialisation process again.
3	private keys	For each combination of relying party and account, one asymmetric key pair exists. A private key from such a key pair is generated temporarily on the TOE during the authentication and registration process for that account. Private keys must never leave the TOE.

Table 3.1: Assets to be protected by the TOE.

3.1.2 Subjects

This Security Target considers the following external entities and subjects:

Authenticator: FIDO Authenticator intended for FIDO Universal Second Factor (U2F) authentication. The authenticator, also referred to as a „U2F token“ (or just „token“), communicates with an external server controlled by a relying party (RP) that supports the standardised FIDO U2F protocol.

User: An *end-user* is the owner of the authenticator (TOE) who is legitimated to use the token.

Manufacturer: The *manufacturer* refers to the FIDO token manufacturer (cf. Life-Cycle)

Relying Party: A web site or other entity that uses a FIDO protocol to directly authenticate users (i.e., performs peer-entity authentication).

User Agent: A client application running on the user device that is acting on behalf of a user in a client-server system. Examples of user agents include web browsers and mobile applications.

Threat Agents: The attacker is a human or a process acting on his behalf located outside the TOE. The main goal of the attacker is to access the token in such a way that allows him to circumvent the authentication mechanism. He does so by e.g. trying to gain knowledge of secret information stored on the U2F authenticator. The attacker has high attack potential.

3.2 Threats

This section describes the threats to be averted by the TOE independently or in collaboration with its IT environment. These threats result from the assets protected by the TOE and the method of the TOE's use in the operational environment.

All attacks are executed through the threat agent described in the section above.

If the seed or private keys are compromised, an attacker is able to violate all security goals of FIDO. The attacker could impersonate the user with a cloned authenticator and has unauthorized access to the relying party. Similar, the compromise of the MACKey can lead to similar violations of the security goals of FIDO.

In the following the threats are listed in detail. For reference, identifiers in brackets link to threats from the “FIDO Security Reference” document [FIDOSpec].

T.InformationLeakage

Adverse action: An attacker may exploit information leaking from the TOE during its usage in order to disclose confidential keys stored on the TOE token or/and exchanged between the TOE and the relying party. The information leakage may be inherent in the normal operation or caused by the attacker.

Information leakage can occur through covered channels (side channels). Typical side channel attacks include measuring the power consumption (differential power/electromagnetic analysis) during operational use or to actively enforce leakage by fault injection (differential fault analysis).

Asset: seed, private key, MACKey

Threat agent: Attacker

T.PhysTamper

Adverse action: An attacker may perform physical probing of the TOE in order to disclose/reconstruct the keys. An attacker may physically modify the authenticator in order to alter its security functionality (hardware and software part).

Physical tampering may be focused directly on the disclosure or manipulation of key material Techniques commonly employed in IC failure analysis and IC reverse engineering efforts may be used. Before that, hardware security mechanisms and layout characteristics need to be identified. The modification may result in the deactivation of a security function. Changes of circuitry or data can be permanent or temporary.

Asset: seed, private key, MACKey

Threat agent: Attacker

T.KeyCompromise (T-1.4.2)

Adverse action: An attacker succeeds in extracting a user's MACKey, private signing keys, or the seed.

Asset: seed, private key, MACKey

Threat agent: Attacker

T.UserVerificationBy-Pass (T-1.4.3)

Adverse action: An attacker uses an authenticator or a private signing key either with or without being noticed by the legitimate user.

Asset: seed, private key, MACKey

Threat agent: Attacker

T.SignatureAlgorithmAttack (T-1.4.8)

Adverse action: A cryptographic attack is discovered against the public key cryptography system used to sign data by the FIDO authenticator.

Asset: seed, private key

Threat agent: Attacker

T.AbuseFunctionality (T-1.4.9)

Adverse action: It might be possible for an attacker to abuse the authenticator functionality by sending commands with invalid parameters or invalid commands to the authenticator

Asset: seed, private key, MACKey

Threat agent: Attacker

T.RandomNumberPrediction (T-1.4.10)

Adverse action: It might be possible for an attacker to get access to information allowing the prediction of RNG data.

Asset: seed, private key, MACKey

Threat agent: Attacker

T.FirmwareRollback (T-1.4.11)

Adverse action: An attacker might be able to install a previous and potentially buggy version of the firmware.

Asset: seed, private key, MACKey

Threat agent: Attacker

T.Forgery (SG-11)

Adverse action: An attacker may attempt to modify intercepted communications in order to masquerade as the legitimate user and login to the relying party.

Asset: seed, private key, MACKey

Threat agent: Attacker

T.Clone

Adverse action: An attacker clones a U2F authenticator and uses the cloned authenticator to login at the relying party as the legitimate user.

Asset: seed, private key, MACKey

Threat agent: Attacker

T.PrivacyViolation

Adverse action: An attacker is able to trace logins to two different accounts by information exchanged between the token and (one or more) relying parties to the same FIDO U2F authenticator, thus violating the user's privacy.

Asset: seed, private key, MACKey

Threat agent: Attacker

3.3 Organizational Security Policies

The TOE shall comply with the following Organizational Security Policies (OSP) as security rules, procedures, practices, or guidelines imposed by an organization upon its operations (see CC part 1, sec. 3.2 [CC1]).

P.UserConsent (SG-7)

The user is notified before a relationship to a new relying party is being established (requiring explicit consent).

P.Attestation

The relying party must be able to verify the signature of the registration step as well as the FIDO Authenticator model/type in order to calculate the associated risk.

3.4 Assumptions

The assumptions describe the security aspects of the environment in which the TOE will be used or is intended to be used.

A.SeparationMechanism (SA-2)

Operating system privilege separation mechanisms relied up on by the software modules involved in a FIDO operation on the user device perform as advertised, e.g. boundaries between user and kernel mode, between user accounts, and between applications (where applicable) are securely enforced and security principals can be mutually, securely identifiable.

A.TrustworthAppServerChannel (SA-3)

Applications on user devices are able to establish secure channels that provide trustworthy server authentication, and confidentiality and integrity for messages (TLS).

A.TrustworthCE (SA-5)

The computing environment (CE) on the FIDO user device and the applications involved in a FIDO operation act as trustworthy agents of the user.

A.TrustworthRP (SA-7)

The computing resources at the relying party (RP) involved in processing a FIDO operation act as trustworthy agents of the relying party.

3.5 Security Objectives for the TOE

This chapter describes the security objectives for the TOE and for the TOE environment.

OT.LeakageResistance

The TOE must provide protection against disclosure of TSF-data stored by the token (passive side channel attacks), including disclosure

- by measurement and analysis of the shape and amplitude of signals or the time between events found by measuring signals on the electromagnetic field, power consumption, clock, or I/O lines
- by physical manipulation of the TOE
- by exploiting software bugs and vulnerabilities

OT.TamperResistance

The TOE must provide protection of confidentiality and integrity of secret keys, TSF and the TOEs embedded software active or semi-active side channel attacks). The TOE must in particular implement counter-measures that prevent information extraction by

- power and emission analysis using the TOE's contact-based, contactless or other interfaces combined with active probing of the IC
- other types of side channel attacks, including those using tools employed in solid-state physics research and IC failure analysis,

- manipulation of the hardware and its security functionality, e.g. fault analysis
- reverse-engineering to understand the design and its properties and functionality.

OT.Prot_Abuse-Func (T-1.4.9)

The TOE must prevent that functions of the TOE, which may not be used in the TOE's operational phase, can be abused in order to manipulate or disclose TSF-data stored in the TOE or to manipulate (bypass, deactivate or modify) soft-coded security functionality of the TOE.

OT.HighLevelOfAssurance (SG-1)

Contribute to a FIDO authentication with a high level of assurance by employing a cryptographic protocol with high (cryptographic) strength.

OT.Unlinkability (SG-4)

Protect the protocol conversation such that any two relying parties cannot link the separate conversation to one user (i.e. be unlinkable).

OT.TrustworthyData

The FIDO Authenticator only accepts and processes sensitive data that are generated by itself and are linkable to the AppID of the relying party.

OT.AuthenticatorLeakResilience (SG-6)

Be resilient to leaks from other FIDO Authenticators. I.e., nothing that a particular FIDO Authenticator could possibly leak can help an attacker to impersonate any other user to any relying party.

OT.LimitedPII (SG-8)

Limit the amount of personal identifiable information (PII) exposed to the relying party to the absolute minimum.

OT.UserConsentForAllProcesses

For all processes the user invokes with the TOE, successful execution requires an immediately-prior demonstration of user presence.

3.6 Security Objectives for the Operational Environment

OE.RespectSecurityBoundaries (SG-15)

Ensure that registrations and key material as a shared system resource is appropriately protected according to the operating environment privilege boundaries in place on the FIDO user agent.

OE.Attestation

The Attestation certificate as well as the attestation key material is generated with high cryptographic security in a secure environment. The attestation keys are securely imported by the manufacturer and stored in the TOE. For privacy reasons a large amount of FIDO Token shares the same attestation key. Furthermore the manufacturer imports the attestation certificate, with indicate model and type of the authenticator.

3.7 Security Objective Rationale

The following table 3.2 provides an overview for security objectives coverage (TOE and its environment). It shows that all threats and OSPs are addressed by the security objectives. It also shows that all assumptions are addressed by the security objectives for the TOE environment.

3.8 Rationale for Objectives for the TOE and the Operational Environment

	OT.LeakageResistance	OT.TamperResistance	OT.Prot_Abuse-Func	OT.HighLevelOfAssurance	OT.Unlinkability	OT.TrustworthyData	OT.AuthenticatorLeakResilince	OT.LimitedPII	OT.UserConsentForAllProcesss	OE.RespectSecurityBoundaries	OE.Attestation
T.InformationLeakage	x	x	x	x							
T.PhysTampering	x	x	x	x							
T.KeyCompromise	x	x		x			x				
T.UserVerificationByPass	x	x		x			x				
T.SignatureAlgorithmAttack				x							
T.AbuseFunctionality			x				x				
T.RandomNumberPrediction				x	x		x				
T.FirmwareRollback			x								
T.Forgery				x		x	x				
T.Clone	x	x	x				x				
T.PrivacyViolation					x			x			x
P.UserConsent									x	x	
P.Attestation											x
A.TrustworthCE										x	
A.SeparationMechanism										x	
A.TrustworthAppServerChannel										x	
A.TrustworthRP										x	

Table 3.2: Security Objective Rationale

3.9 Security objective sufficiency

Countering of threats by security objectives:

The threats **T.InformationLeakage** and **T.PhysTampering** are protected by the directly related security objectives *OT.LeakageResistance* and *OT.TamperResistance* as well as *OT.Prot_Abuse-Func* and *OT.HighLevelOfAssurance*.

T.KeyCompromise via publicly known data produced by the TOE could lead to impersonating the user with a cloned authenticator. *OT.HighLevelOfAssurance* counters this threat by implementing cryptographically secure generation of the key pair. *OT.LeakageResistance* and *OT.TamperResistance* prevent leakage of the key. *OT.AuthenticatorLeakResilience* prevents that the authenticator himself leaks.

T.UserVerificationByPass: Impersonating the user is prevented by *OT.HighLevelOfAssurance*. Readout of the MACKey and/or private keys is prevented by *OT.LeakageResistance* and *OT.TamperResistance*. Leakage of the MACkey and/or private keys from the authenticator is prevented by *OT.AuthenticatorLeakResilience.*

T.SignatureAlgorithmAttack: *OT.HighLevelOfAssurance* aims for the use of proper cryptographic security and prevents the use of insecure signature algorithms.

T.AbuseFunctionality: The security objective *OT.Prot_Abuse-Func* and *OT.AuthenticatorLeakResilience* ensures that the usage of functions having not to be used in the operational phase is effectively prevented.

T.RandomNumberPrediction is covered by *OT.HighLevelOfAssurance. OT.AuthenticatorLeakResilience, and OT.Unlinkability* prevent to gain any information w.r.t. generated random numbers.

T.FirmwareRollback is directly addressed by *OT.Prot_Abuse-Func.*

T.Forgery is prevented by secure cryptography (*OT. HighLevelOfAssurance),* leak resistance (*OT.AuthenticatorLeakResilience)* and is also covered by *OT.TrustworthyData.*

T.Clone: To prevent this threat any kind of leakage w.r.t information stored on the TOE must be minimized. *OT.LeakageResistance, OT.TamperResistance, OT.AuthenticatorLeakResilience,* and *OT.Prot_Abuse-Func* covers this threat.

T.PrivacyViolation is prevented by *OT.Unlinkability, OT.LimitedPII* and *OE.Attestation (group keys).*

Enforcement of Organizational Security Policies by security objectives:

P.UserContent *OE.RespectSecurityBoundaries and OT. UserConsentForAllProcesses* provides for the guarantee that the user is informed before establishing a relationship between user and relying party. *OE.RespectSecurityBoundaries e*nsures that key material (e.g. TLS keys) and registrations handled by the operating environment cannot be easily manipulated. If the privileged boundaries of the operating environment cannot be maintained a user could be fooled into registering at a new relying party while thinking she is merely providing user presence to authenticate to a known relying party. *OT.UserConsentForAllProcesses* ensures that the authenticator only act if user presence has been verified prior to execution.

P.Attestation *OE.Attestation* provides the guarantee, that the attestation certificate and the attestation private key are generated and imported in a secure environment.

Upholding assumptions by environment objectives:

A.TrustworthCE, A.SepartionMechanism, A.TrustworthAppServerChannel and **A.TrustworthRP** are covered by *OE.RespectSecurityBoundaries.*

4 Extended Components Definition

This Security Target uses components defined as extensions to CC part 2 [CC2] as specified in [FIDOPP].

4.1 Definition of the Family FCS_RNG

To define the IT security functional requirements of the TOE a 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. The component FCS_RNG.1 is not limited to generation of cryptographic keys unlike the component FCS_CKM.1. The similar component FIA_SOS.2 is intended for non-cryptographic use.

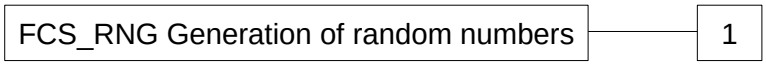
The family 'Generation of random numbers (FCS_RNG)' is specified as follows:

FCS_RNG Generation of random numbers

Family behaviour

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

Component levelling:



FCS_RNG.1 Generation of random numbers requires that the random number generator implements defined security capabilities and that the random numbers meet a defined quality metric.

Management: FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no actions defined to be auditable.

FCS_RNG.1 Random number 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 physical, 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*].

4.2 Definition of the Family FPT_EMS

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 are evaluation of 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 being not directly addressed by any other component of CC part 2 [CC2].

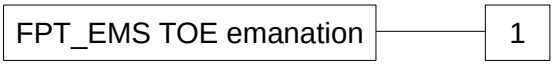
The family 'TOE Emanation (FPT_EMS)' is specified as follows:

FPT_EMS TOE emanation

Family behaviour

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 to not emit intelligible emissions enabling access to TSF data or user data.

FPT_EMS.1.2 Interface Emanation requires to not emit interface emanation enabling 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*].

4.3 Definition of the Component FCS_CKM.5

This chapter describes functional requirements for key derivation as process by which one or more keys are calculated from either a pre-shared key or 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

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 derive cryptographic keys [assignment: *key type*] from [assignment: *input parameters*] in accordance with a specified cryptographic key derivation algorithm [assignment: *cryptographic key derivation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].

5 Security Requirements

This part defines detailed security requirements that shall be satisfied by the TOE. The statement of TOE security requirements shall define the functional and assurance security requirements that the TOE must satisfy in order to meet the security objectives for the TOE.

Common Criteria allows several operations to be performed on security requirements on the component level: *refinement, selection, assignment* and *iteration*, cf. sec. 8.1 of [CC1]. Each of these operations is used in this ST.

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinements of security requirements are denoted in such a way that added words are in **bold text** and removed words are ~~crossed out~~.

The **selection** operation is used to select one or more options provided by CC in stating a requirement. Selections that have been made by the PP author are denoted as underlined text. A footnote will show the original text from [CC2]. Selections filled in by the ST author are italicized. A footnote will show the original text from [FIDOPP].

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments that have been made by the PP author are denoted as underlined text. A footnote will show the original text from [CC2]. Assignments filled in by the ST author are italicized. A footnote will show the original text from [FIDOPP].

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing a slash "/", and the iteration indicator after the component identifier. For the sake of better readability, the iteration operation may also be applied to a non-repeated single component in order to indicate that such component belongs to a certain functional cluster. In such a case, the iteration operation is applied to only one single component.

5.1 Security Functional Requirements

5.1.1 Class FCS Cryptographic Support

FCS_CKM.1 **Cryptographic key generation**

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]: Fulfilled by *FCS_COP.1/MAC*
	FCS_CKM.4 Cryptographic key destruction: Fulfilled by FCS_CKM.4
FCS_CKM.1.1	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm: *DRNG instantiate by JavaCard RandomData object with constant ALG_SECURE_RANDOM (see FCS_RNG.1) stored in JavaCard AESKey*[3] *object* and specified cryptographic key sizes *128 bit*[4] that meet the following:*[AIS 20]*[5]*. Key generation as specified in [SP800-133] (chapter 7.1).*

Application note 1: The SFR above applies to all cryptographic key generation methods implemented on the TOE.

3 [assignment: *cryptographic key generation algorithm*]
4 [assignment: *cryptographic key sizes*]
5 *[assignment: list of standards compliant to [FIDOCrypt]]*

Application note 2: The keys generated in this SFR will be used as input for the KDF used in SFR FCS_COP.1 and FCS_COP.1/MAC. Both cryptographic operation use the same KDF. In FCS_COP.1 the KDF is used to generate the ECDSA signature key, in FCS_COP.1/MAC the KDF is used to generate the MAC.

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]: Fulfilled by FCS_CKM.1
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *Key.clearKey() of the Java Card platform*[6] that meets the following: *[JCAPI304], class javacard.security.Key*[7].

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]: Fulfilled by *FCS_CKM.5/U2F-Private. The SFR fulfills the dependency because the KDF as defined in FCS_CKM.5/U2F-Private is used to generate the U2F key, that is used for the ECDSA signature creation.* FCS_CKM.4 Cryptographic key destruction: Fulfilled by FCS_CKM.4
FCS_COP.1.1	The TSF shall perform *signature creation*[8] in accordance with a specified cryptographic algorithm *ECDSA using NIST P-256*[9] and cryptographic key sizes *256 bit*[10] that meet the following: *[FIPS186-4]*[11]*. Data is first hashed with SHA 256 according to [FIPS 180-4].*

Application note 3: The above SFR applies to all cryptographic operations implemented on the TOE.

FCS_COP.1/MAC Cryptographic operation – MAC

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]: Fulfilled by FCS_CKM.1 FCS_CKM.4 Cryptographic key destruction: Fulfilled by FCS_CKM.4
FCS_COP.1.1	The TSF shall perform message authentication code[12] in accordance with a specified cryptographic algorithm *AES CMAC*[13] and cryptographic key sizes *128 bit*[14]. that meet the

6 [assignment: *cryptographic key destruction method*]
7 [assignment: *list of standards*]
8 [assignment: *list of cryptographic operations*]
9 [assignment: *cryptographic algorithm*]
10 [assignment: *cryptographic key sizes*]
11 [assignment: *list of standards compliant to [FIDOCrypt]*]
12 [assignment: *list of cryptographic operations*]
13 [assignment: *cryptographic algorithm*]
14 [assignment: *cryptographic key sizes*]

following: *[SP800-38B][15] and uses AES according to [FIPS 197]. The CMAC output is afterwards hashed with SHA 256 according to [FIPS 180-4].*

FCS_RNG.1 Random number generation – RNG for seed/nonce/MACKey generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a *hybrid deterministic*[16] random number generator that implements[17]:

- (DRG.4.1) The internal state of the RNG uses a PTRNG of class PTG.2 as random source.
- (DRG.4.2) The RNG provides forward secrecy.
- (DRG.4.3) The RNG provides backward secrecy, even if the current internal state is known.
- (DRG.4.4) The RNG provides enhanced forward secrecy for every call.
- (DRG.4.5) The internal state of the RNG is seeded by a PTRNG of class PTG.2.

FCS_RNG.1.2 The TSF shall provide random numbers that meet[18]:

- (DRG.4.6) The RNG generates output for which two strings of bit length 128 are mutually different with probability 1-2^-128.
- (DRG.4.7) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A as defined in [AIS 20].

Application note 4: The above SFR requires the TOE to generate random numbers used for seed, nonce and MACKey generation.

FCS_CKM.5/U2F-Private Cryptographic key derivation for U2F private keys

Hierarchical to: No other components.

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

FCS_CKM.4 Cryptographic key destruction: Fulfilled by FCS_CKM.4

FCS_CKM.5.1 The TSF shall derive cryptographic keys U2F private keys[19] from seed as the confidential key, and FIDO AppID and nonce[20] in accordance with a specified cryptographic key derivation *KDF in counter mode that meet the following [SP800-108][21] using AES CMAC as*

15 [assignment: *list of standards compliant to [FIDOCrypt]*]
16 [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*]
17 [assignment: *list of security capabilities*]
18 [assignment: *a defined quality metric*]
19 [assignment: key type]
20 [assignment: input parameters]
21 [assignment: *cryptographic key derivation algorithm*]

PRF and specified cryptographic key sizes *128 bit*[22] that meet the following: *[SP800-38B]*[23] *using AES according to [FIPS 197].*

Application note 5: The SFR implements the process of generating random bits in the FIDO authentication process. The generated random bits of the FIDO compliant KDF are used to derive FIDO private keys from the global seed.

Application note 6: The generated keys are used for the SFR FCS_COP.1.

Application note 7: The generated key are generated according to [FIPS186-4]

FCS_CKM.5/U2F-Public Cryptographic key derivation for U2F public keys

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]: The dependency is not resolved, because the TOE does not use the U2F public key for cryptographic operations
	FCS_CKM.4 Cryptographic key destruction: Fulfilled by FCS_CKM.4
FCS_CKM.5.1	The TSF shall derive cryptographic keys U2F public keys[24] from the corresponding U2F private key[25] in accordance with a specified cryptographic key derivation *ECC Key Pair Generation algorithm*[26] and specified cryptographic key sizes *256 bit*[27] that meet the following: *[FIPS186-4]*[28].

5.1.2 Class FDP User data protection

FDP_IFC.1/Initialisation Subset information flow control - Initialisation

Hierarchical to:	No other components.
Dependencies:	FDP_IFF.1 Simple security attributes: Fulfilled by FDP_IFF.1/ Initialisation
FDP_IFC.1.1	The TSF shall enforce the information flow control FDP_IFF.1/Initialisation SFP[29] on[30]:

- Subjects: end-user, authenticator and *none*[31];
- Information: seed, MACKey and *none*[32];
- Operation: generate seed and MACKey for initiating the authenticator and *none*[33].

FDP_IFC.1/Reset Subset information flow control - Reset

22 [assignment: *cryptographic key sizes*]
23 [assignment: *list of standards compliant to [FIDOCrypt]*]
24 [assignment: key type]
25 [assignment: input parameters]
26 [assignment: *cryptographic key derivation algorithm*]
27 [assignment: *cryptographic key sizes*]
28 [assignment: *list of standards compliant to [FIDOCrypt]*]
29 [assignment: *information flow control SFP*]
30 [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*]
31 [assignment: *other subjects*]
32 [assignment: *other information*]
33 [assignment: *other operations*]

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes: Fulfilled by FDP_IFF.1/Reset

FDP_IFC.1.1 The TSF shall enforce the information flow control FDP_IFF.1/Reset SFP[34] on[35]:

- Subjects: end-user, authenticator and *none*[36];
- Information: seed, MACKey and *none*[37];
- Operation: secure_erase seed and MACKey to reset the authenticator and *none*[38].

FDP_IFC.1/Registration Subset information flow control - Registration

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes: Fulfilled by FDP_IFF.1/Registration

FDP_IFC.1.1 The TSF shall enforce the information flow control FDP_IFF.1/Registration SFP[39] on[40]:

- Subjects: end-user, authenticator, relying party and *none*[41];
- Information: AppID, nonce, MACKey , MAC, private key, public key, seed, challenge and *none*[42];
- Operation: requesting AppID, challenge from relying party, generate_private_key, generate public_key, generate_mac, signature generation, sending out nonce, MAC, public key and signature to relying party and *none*[43].

FDP_IFC.1/Authentication Subset information flow control - Authentication

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes: Fulfilled by FDP_IFF.1/Authentication

FDP_IFC.1.1 The TSF shall enforce the information flow control FDP_IFF.1/Authentication SFP[44] on[45]:

- Subjects: end-user, authenticator, relying party and *none*[46];
- Information: AppID, nonce, MACKey , MAC, private key, seed, challenge and *none*[47];
- Operation: requesting AppID, nonce, challenge from relying party, private key generation, signature generation, sending signature to relying party and *none*[48].

34 [assignment: *information flow control SFP*]
35 [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*]
36 [assignment: *other subjects*]
37 [assignment: *other information*]
38 [assignment: *other operations*]
39 [assignment: *information flow control SFP*]
40 [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*]
41 [assignment: *other subjects*]
42 [assignment: *other information*]
43 [assignment: *other operations*]
44 [assignment: *information flow control SFP*]
45 [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*]
46 [assignment: *other subjects*]
47 [assignment: *other information*]
48 [assignment: *other operations*]

FDP_IFF.1/Initialisation Simple Security Attributes - Initialisation

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control: Fulfilled by FDP_IFC.1/Inititialisation

FMT_MSA.3 Static attribute initialisation : Fulfilled by FMT_MSA.3/Initialisation

FDP_IFF.1.1 The TSF shall enforce the initialisation process SFP[49] based on the following types of subject and information security attributes:

1) Subjects:

a) end-user

b) authenticator

2) information:

a) seed

b) MACKey

3) security attributes:

a) user_presence: true, false

b) security_state of the authenticator: delivery_state, ready_for_use[50].

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:

1) The end-user initialises the authenticator.

2) After initialisation the authenticator indicates its security status to the end-user[51].

FDP_IFF.1.3 The TSF shall enforce the following rules:

1) user_presence = check_user_presence()

a) user_presence == false → continue with 6)

b) user_presence == true → continue with 2)

2) security_state = check_security_state()

a) security_state == ready_for_use → continue with 6)

b) security_state == delivery_state → continue with 3)

3) seed = RNG() (cf. FCS_RNG.1)

4) MACKey = RNG() (cf. FCS_RNG.1)

5) set security_state = ready_for_use

6) manage_user_presence() [cf. FIA_UAU.2 resp. FIA_UAU.6][52]

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: none[53].

49 [assignment: *information flow control SFP*]

50 [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]

51 [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*]

52 [assignment: *additional information flow control SFP rules*]

53 [assignment: *rules, based on security attributes, that explicitly authorise information flows*]

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

1) The MACKey never leaves the TOE

2) The seed never leaves the TOE[54].

Application note 8: The function manage_user_presence() (re)sets the user presence value according to the requirements defined in FIA_UAU.2 and FIA_UAU.6

Application note 9: The TSF shall enforce the rules in the specific order given in FDP_IFF.1.3. If deviations are necessary, the developer has to take special care in the choice of the order due to avoid vulnerabilities. Deviations must not contradict the [FIDOSpec].

FDP_IFF.1/Reset Simple Security Attributes - Reset

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control: Fulfilled by FDP_IFC.1/Reset

FMT_MSA.3 Static attribute initialisation : Fulfilled by FMT_MSA.3/Reset

FDP_IFF.1.1 The TSF shall enforce the reset process SFP[55] based on the following types of subject and information security attributes:

1) Subjects:

a) end-user

b) authenticator

2) information:

a) seed

b) MACKey

3) security attributes:

a) user_presence: true, false

b) security_state: delivery_state, ready_for_use[56].

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:

1) The end-user resets the authenticator

2) After the reset process the authenticator indicates its security_state to the end-user[57].

FDP_IFF.1.3 The TSF shall enforce the following rules:

1) user_presence = check_user_presence()

a) user_presence == false → continue with 6)

b) user_presence == true → continue with 2)

2) security_state = check_security_state()

54 [assignment: *rules, based on security attributes, that explicitly deny information flows*]
55 [assignment: *information flow control SFP*]
56 [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]
57 [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes]*

a) security_state = delivery_state → continue with 6)

b) security_state = ready_for_use → continue with 3)

3) set security_state = delivery_state

4) secure_erase(seed)

5) secure_erase(MACKey)

6) manage_user_presence() [cf. FIA_UAU.2 resp. FIA_UAU.6][58].

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: none[59].

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

1) The MACKey never leaves the TOE

2) The seed never leaves the TOE[60].

Application note 10: The TSF shall enforce the rules in the specific order given in FDP_IFF.1.3. If deviations are necessary, the developer has to take special care in the choice of the order due to avoid vulnerabilities. Deviations must not contradict the [FIDOSpec].

FDP_IFF.1/Registration Simple Security Attributes - Registration

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control: Fulfilled by FDP_IFC.1/Registration

FMT_MSA.3 Static attribute initialisation : Fulfilled by FMT_MSA.3/Registration

FDP_IFF.1.1 The TSF shall enforce the registration process SFP[61] based on the following types of subject and information security attributes:

1) Subjects:

a) authenticator

b) relying party

c) end-user

2) information:

a) AppID

b) nonce

c) MACKey

d) MAC(MACKey, [nonce|AppID])

e) private key

f) public key

g) seed

h) challenge

58 [assignment: *additional information flow control SFP rules*]
59 [assignment: *rules, based on security attributes, that explicitly authorise information flows*]
60 [assignment: *rules, based on security attributes, that explicitly deny information flows*]
61 [assignment: *information flow control SFP*]

3) security attributes:

a) user_presence: true, false[62].

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:

1) The end-user starts to register the authenticator at the relying party

2) The authenticator requests AppID and challenge from RP

3) The authenticator sends nonce, MAC(MACKey, [nonce|AppID]), signature and public key[63].

FDP_IFF.1.3 The TSF shall enforce the following rules:

1) relying party supplies AppID

2) user_presence = check_user_presence()

a) user_presence == false → continue with 10)

b) user_presence == true → continue with 3)

3) security_state = check_security_state()

a) security_state = delivery_state → continue with 10)

b) security_state = ready_for_use → continue with 4)

4) nonce = RNG() (cf. FCS_RNG.1)

5) PrivK = KDF(seed, AppID, nonce) (cf. FCS_CKM.5/U2F-Private)

6) PubK = genKeyPair(PrivK)(cf. FCS_CKM.5/U2F-Public)

7) KeyHandle = nonce | MAC(MACKey, AppID | nonce)

8) sig = sign(AttestationPrivK, KeyHandle| AppID | challenge, PubK)

9) delete(PubK) and secure_erase(PrivK)

10) manage_user_presence() [cf. FIA_UAU.2 resp. FIA_UAU.6][64].

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: none[65].

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

1) The MACKey never leaves the TOE

2) The seed never leaves the TOE

3) Private keys never leave the TOE[66].

Application note 11: The TSF shall enforce the rules in the specific order given in FDP_IFF.1.3. If deviations are necessary, the developer has to take special care in the choice of the order due to avoid vulnerabilities. Deviations must not contradict the [FIDOSpec].

62 [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]

63 [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes]*

64 [assignment: *additional information flow control SFP rules*]

65 [assignment: *rules, based on security attributes, that explicitly authorise information flows*]

66 [assignment: *rules, based on security attributes, that explicitly deny information flows*]

FDP_IFF.1/Authentication Simple Security Attributes - Authentication

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control: Fulfilled by FDP_IFC.1/Authentication

FMT_MSA.3 Static attribute initialisation: Fulfilled by FMT_MSA.3/Authentication

FDP_IFF.1.1 The TSF shall enforce the authentication process SFP[67] based on the following types of subject and information security attributes:

1) Subjects:

a) authenticator

b) relying party

c) end-user

2) information:

a) AppID

b) nonce

c) MACKey

d) MAC(MACKey, [nonce|AppID])

e) private key

f) seed

g) challenge

3) security attributes:

a) user_presence: true, false

b) MAC verification status: true, false[68].

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:

1) The end-user starts authentication at the RP

2) The authenticator requests AppID, nonce, MAC from RP

3) The authenticator sends the signature to the RP[69].

FDP_IFF.1.3 The TSF shall enforce the following rules:

1) security_state = check_security_state()

a) security_state = delivery_state → continue with 7)

b) security_state = ready_for_use → continue with 2)

2) verify MAC

a) verify(MAC) == false → continue with 7)

b) verify(MAC) == true → continue with 3)

67 [assignment: *information flow control SFP*]

68 [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]

69 [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*]

3) user_presence = check_user_presence()

a) user_presence == false → continue with 7)

b) user_presence == true → continue with 4)

4) PrivK = KDF(seed, AppID, nonce) (cf. FCS_CKM.5/U2F-Private)

5) sig = sign(PrivK, AppID | challenge)

6) secure_erase(PrivK)

7) manage_user_presence() [cf. FIA_UAU.2 resp. FIA_UAU.6][70].

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: *none*[71].

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

1) The MACKey never leaves the TOE

2) The seed never leaves the TOE

3) Private keys never leave the TOE[72].

Application note 12: The TSF shall enforce the rules in the specific order given in FDP_IFF.1.3. If deviations are necessary, the developer has to take special care in the choice of the order due to avoid vulnerabilities. Deviations must not contradict the [FIDOSpec].

FDP_SDI.1 Stored data integrity monitoring

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_SDI.1.1 The TSF shall monitor ~~user data~~ **all assets as defined in Table 3.1** stored in containers controlled by the TSF for *integrity errors as described in chapter 9.1.4 SF.INTEGRITY([SCST])*[73] on all objects, based on the following attributes: *checksum integrity of cryptographic keys and their associated security attributes*[74].

5.1.3 Class FIA Identification and Authentication

FIA_UAU.2 User authentication before any action

Hierarchical to: FIA_UAU.1 Timing of authentication

Dependencies: FIA_UID.1 Timing of identification: not fulfilled, but justified: Identification is not needed for user presence

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated **by the user demonstrating proof of presence** before allowing any other TSF-mediated actions on behalf of that user.

70 [assignment: *additional information flow control SFP rules*]
71 [assignment: *rules, based on security attributes, that explicitly authorise information flows that model the "check-only" command 0x07 [FIDOSpec]*]
72 [assignment: *rules, based on security attributes, that explicitly deny information flows*]
73 [assignment: *integrity errors*]
74 [assignment: *user data attributes*]

Application note 13: Proof of presence means e.g. pressing some button or placing the token into the proximity range of an NFC-enabled device[75]. Note: After a single action, the presence check flag on the card is disabled. Malware on the host PC / smartphone could send a reset command to the reader programmatically, thereby circumventing the presence check: It is not possible for the card to distinguish if the reader sent a reset command or if it was physically removed from the reader. So the user must keep his system secure, i.e. that the host PC / smartphone is free of malware and under full control of the user.

FIA_UAU.6 Re-authenticating

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.6.1 The TSF shall **require to** re-authenticate the user **by the user demonstrating proof of presence again** ~~under the conditions~~ **when one of the following events occurres**[76]:

- *Initialisation*
- *Reset*
- *Registration*
- *Authentication.*

5.1.4 Class FMT Security Management

FMT_SMF.1 Security management functions

Hierarchical to: No other components.

Dependencies: No dependencies.

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

(1) creation and destruction of the MACKey/seed,

(2) reset the authenticator,

(3) *none*[77].

FMT_SMR.1 Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification: not fulfilled, but justified: No user is identified, only user presence is checked

FMT_SMR.1.1 The TSF shall maintain the roles *user and authenticator*[78].

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

FMT_MTD.1 Management of TSF data

75 In case of the TOE, the proof of presence is demonstrated by placing the token into the proximity range of an NFC-enabled device or by inserting the card into a card reader

76 [assignment: *list of conditions under which re-authentication is required*]

77 [assignment: *list of other security management functions to be provided by the TSF*]

78 [assignment: *the authorised identified roles*]

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles: Fulfilled by FMT_SMR.1
	FMT_SMF.1 Specification of Management Functions: Fulfilled by FMT_SMF.1
FMT_MTD.1.1	The TSF shall restrict the ability to *change_default, modify, and clear*[79] the *MACKey, seed and private keys*[80] to the *user*[81].

FMT_MSA.1/Initialisation Management of security attributes -Initialisation

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]: Fulfilled by FDP_IFC.1/Initialisation
	FMT_SMR.1 Security roles: Fulfilled by FMT_SMR.1
	FMT_SMF.1 Specification of Management Functions: Fulfilled by FMT_SMF.1
FMT_MSA.1.1	The TSF shall enforce the information flow control FDP_IFC.1/Initialisation SFP[82] to restrict the ability to query[83] the security attribute security_state[84] to *authenticator and user*[85].

FMT_MSA.3/Initialisation Static attribute initialisation - Initialisation

Hierarchical to:	other components.
Dependencies:	FMT_MSA.1 Management of security attributes: Fulfilled by FMT_MSA.1/Initialisation
	FMT_SMR.1 Security roles: Fulfilled by FMT_SMR.1
FMT_MSA.3.1	The TSF shall enforce the information flow control FDP_IFF.1/Initialisation SFP[86] to provide restrictive[87] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow the *authenticator*[88] to specify alternative initial values to override the default values when an object or information is created.

Application note 14: Restrictive means: user_presence = false and security_state = delivery_state

FMT_MSA.1/Reset Management of security attributes - Reset

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]: fulfilled by FDP_IFC.1/Reset
	FMT_SMR.1 Security roles: Fulfilled by FMT_SMR.1

79 [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
80 [assignment: *list of TSF data*]
81 [assignment: *the authorised identified roles*]
82 [assignment: *access control SFP(s), information flow control SFP(s)*]
83 [selection: *change_default, query, modify, delete*, [assignment: *other operations*]]
84 [assignment: *list of security attributes*]
85 [assignment: *the authorised identified roles*]
86 [assignment: *access control SFP, information flow control SFP*]
87 [selection, choose one of: *restrictive, permissive,* [assignment: *other property*]]
88 [assignment: *the authorised identified roles*]

FMT_SMF.1 Specification of Management Functions: Fulfilled by FMT_SMF.1

FMT_MSA.1.1 The TSF shall enforce the information flow control FDP_IFC.1/Reset SFP[89] to restrict the ability to query[90] the security attribute user_presence, security_state[91] to *authenticator and user*[92].

FMT_MSA.3/Reset Static attribute initialisation - Reset

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes: Fulfilled by FMT_MSA.1/Reset

FMT_SMR.1 Security roles: Fulfilled by FMT_SMR.1

FMT_MSA.3.1 The TSF shall enforce the information flow control FDP_IFF.1/Reset SFP[93] to provide restrictive[94] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the *authenticator*[95] to specify alternative initial values to override the default values when an object or information is created.

*Application note 15: R*estrictive means: user_presence = false and security_state = ready_for_use

FMT_MSA.1/Registration Management of security attributes - Registration

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]: fulfilled by FDP_IFC.1/Registration

FMT_SMR.1 Security roles: Fulfilled by FMT_SMR.1

FMT_SMF.1 Specification of Management Functions: Fulfilled by FMT_SMF.1

FMT_MSA.1.1 The TSF shall enforce the information flow control FDP_IFC.1/Registration SFP[96] to restrict the ability to query[97] the security attribute user_presence/security_state[98] to *authenticator and user*[99].

FMT_MSA.3/Registration Static attribute initialisation - Registration

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes: Fulfilled by FMT_MSA.1/Registration

FMT_SMR.1 Security roles: Fulfilled by FMT_SMR.1

89 [assignment: *access control SFP(s), information flow control SFP(s)*]
90 [selection: *change_default, query, modify, delete*, [assignment: *other operations*]]
91 [assignment: *list of security attributes*]
92 [assignment: *the authorised identified roles*]
93 [assignment: *access control SFP, information flow control SFP*]
94 [selection, choose one of: *restrictive, permissive,* [assignment: *other property*]]
95 [assignment: *the authorised identified roles*]
96 [assignment: *access control SFP(s), information flow control SFP(s)*]
97 [selection: *change_default, query, modify, delete*, [assignment: *other operations*]]
98 [assignment: *list of security attributes*]
99 [assignment: *the authorised identified roles*]

FMT_MSA.3.1 The TSF shall enforce the information flow control FDP_IFF.1/Registration SFP[100] to provide restrictive[101] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the *authenticator*[102] to specify alternative initial values to override the default values when an object or information is created.

Application note 16: Restrictive means: user_presence = false

FMT_MSA.1/Authentication Management of security attributes - Authentication

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]: fulfilled by FDP_IFC.1/Authentication

FMT_SMR.1 Security roles: Fulfilled by FMT_SMR.1

FMT_SMF.1 Specification of Management Functions: Fulfilled by FMT_SMF.1

FMT_MSA.1.1 The TSF shall enforce the information flow control FDP_IFC.1/Authentication SFP[103] to restrict the ability to query[104] the security attribute user_presence/MAC verification status[105] to *authenticator and user*[106].

FMT_MSA.3/ Authentication Static attribute initialisation - Authentication

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes: Fulfilled by FMT_MSA.1/Authentication

FMT_SMR.1 Security roles: Fulfilled by FMT_SMR.1

FMT_MSA.3.1 The TSF shall enforce the information flow control FDP_IFF.1/Authentication SFP[107] to provide restrictive[108] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the *authenticator*[109] to specify alternative initial values to override the default values when an object or information is created.

Application note 17: Restrictive means: user_presence = false and MAC verification status = false

5.1.5 Class FPR Privacy

FPR_ANO.1 Anonymity

100[assignment: *access control SFP, information flow control SFP*]
101[selection, choose one of: *restrictive, permissive,* [assignment: *other property*]]
102[assignment: *the authorised identified roles*]
103[assignment: *access control SFP(s), information flow control SFP(s)*]
104[selection: *change_default, query, modify, delete*, [assignment: *other operations*]]
105[assignment: *list of security attributes*]
106[assignment: *the authorised identified roles*]
107[assignment: *access control SFP, information flow control SFP*]
108[selection, choose one of: *restrictive, permissive*, [assignment: *other property*]]
109[assignment: *the authorised identified roles*]

Hierarchical to: No other components.

Dependencies: No dependencies.

FPR_ANO.1.1 The TSF shall ensure that *all subjects*[110] are unable to determine the real user name bound to *the user*[111].

5.1.6 Class FPT Protection of the TSF

FPT_EMS.1 TOE Emanation

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_EMS.1.1 The TOE shall not emit *power variations , timing variations during command execution and electromagnetic emanation*[112] in excess of *non-useful information*[113] enabling access to

1. the seed
2. the MACKey
3. private keys[114]

and *none*[115].

FPT_EMS.1.2 The TSF shall ensure any users[116] are unable to use the following interface authenticator's contactless/contact-based interface and circuit contacts[117] to gain access to

1. the seed
2. the MACKey
3. private keys[118]

and *none*[119].

FPT_PHP.3 Resistance to physical attack

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_PHP.3.1 The TSF shall resist *physical manipulation and physical probing*[120] to the *TSF*[121] by responding automatically such that the SFRs are always enforced.

FPT_TST.1 TSF testing

Hierarchical to: No other components.

110[assignment: *set of users and/or subjects*]
111[assignment: *list of subjects and/or operations and/or objects*]
112[assignment: *types of emissions*]
113[assignment: *specified limits*]
114[assignment: *list of types of TSF data*]
115[assignment: *list of types of user data*]
116[assignment: *type of users*]
117[assignment: *type of connection*]
118[assignment: *list of types of TSF data*]
119[assignment: *list of types of user data*]
120[assignment: *physical tampering scenarios*]
121[assignment: *list of TSF devices/elements*]

Dependencies:	No dependencies.
FPT_TST.1.1	The TSF shall run a suite of self tests *during initial start-up*[122] to demonstrate the correct operation of *the TSF*[123].
FPT_TST.1.2	The TSF shall provide authorised users with the capability to verify the integrity of *TSF data*[124].
FPT_TST.1.3	The TSF shall provide authorised users with the capability to verify the integrity of *TSF*[125].

Application note 18: Authorised user means user has demonstrated his proof of presence.

122[selection: *during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self test should occur]*]
123[selection: *[assignment: parts of TSF], the TSF*]
124[selection: *[assignment: parts of TSF data], TSF data*]
125[selection: *[assignment: parts of TSF], TSF*]

5.2 Security Assurance Requirements

The assurance requirements for the evaluation of the TOE, its development and operating environment are to choose as the predefined assurance package EAL4 augmented by the following components:

- AVA_VAN.5 (Advanced methodical vulnerability analysis).

5.3 Security Requirements Rationale

5.3.1 Security Functional Requirements Rationale

The following table 5.1 provides an overview for the coverage of the security functional requirements, and also gives evidence for sufficiency and necessity of the chosen SFRs.

OT.LeakageResistance: The SFR's FPT_TST.1, FPT_EMS.1, FMT_SMF.1 and FMT_MTD.1 supports OT.LeakageResistance.

OT.TamperResistance: FPT_PHP.3, FPT_TST.1 supports OT.TamperResistance. The SFR FDP_SDI.1 is used to detect integrity errors, so this SFR also supports the objective OT.TamperResistance.

OT.HighLevelOfAssurance: FCS_CKM.1, FCS_CKM.4, FCS_COP.1, FCS_COP.1/MAC, FCS_RNG.1, FCS_CKM.5/U2F-Private, FCS_CKM.5/U2F-Public are concerned with cryptographic operations and key generation. They support the objective OT.HighLevelOfAssurance as well as the managment SFR's FMT_MTD.1, FMT_SMF.1.

OT.LimitedPII, OT.Unlinkability: Because for each relying party and AppID a new nonce and key pair is generated it is not possible to link between two relying parties or user accounts and limit the amount of of personal identifiable information. OT.LimitedPII and OT.Unlinkability is covered by FCS_COP.1/MAC, FCS_RNG.1, FCS_CKM.5/U2F-Private, FCS_CKM.5/U2F-Public and FPR_ANO.1. The management SFR's FMT_SMF.1, FMT_MTD.1 supports OT.Unlinkability.

OT.TrustworthyData: FCS_COP.1/MAC, FCS_RNG.1, FCS_CKM.5/U2F-Private, FCS_CKM.5/U2F-Public supports OT.TrustworthyData.

OT.AuthenticatorLeakResilience: FCS_RNG.1, all instances of FDP_IFC.1, FDP_IFF.1, the SFRs FIA_UAU.2, FIA_UAU.6, FMT_SMR.1, FMT_SMF.1, all instances of FMT_MSA.1, all instances of FMT_MSA.3, and FMT_MTD.1 supports OT.AuthenticatorLeakResilience.

OT.Prot_Abuse-Func: The TSF self test SFR FPT_TST.1 supports OT.Prot_Abuse-Func.

	OT.LeakageResistance	OT.TamperResistance	OT.HighLevelOfAssurance	OT.Unlinkability	OT.TrustworthyData	OT.AuthenticatorLeakResilience	OT.Prot_Abuse-Func	OT.LimitedPII
Class FCS								
FCS_CKM.1			x					
FCS_CKM.4			x					
FCS_COP.1			x					
FCS_COP.1/MAC			x	x	x			x
FCS_RNG.1			x	x	x	x		x
FCS_CKM.5/U2F-Private			x	x	x			x
FCS_CKM.5/U2F-Public			x	x	x			x
Class FDP								
FDP_IFC.1/Initialisation						x		
FDP_IFC.1/Reset						x		
FDP_IFC.1/Registration						x		
FDP_IFC.1/Authentication						x		
FDP_IFF.1/Initialisation						x		
FDP_IFF.1/Reset						x		
FDP_IFF.1/Registration						x		
FDP_IFF.1/Authentication						x		
FDP_SDI.1		x						
Class FIA								
FIA_UAU.2						x		
FIA_UAU.6						x		
Class FMT								
FMT_SMR.1						x		
FMT_SMF.1	x		x	x		x		

	OT.LeakageResistance	OT.TamperResistance	OT.HighLevelOfAssurance	OT.Unlinkability	OT.TrustworthyData	OT.AuthenticatorLeakResilience	OT.Prot_Abuse-Func	OT.LimitedPII
FMT_MSA.1/Initialisation						x		
FMT_MSA.1/Reset						x		
FMT_MSA.1/Registration						x		
FMT_MSA.1/Authentication						x		
FMT_MSA.3/Initialisation						x		
FMT_MSA.3/Reset						x		
FMT_MSA.3/Registration						x		
FMT_MSA.3/Authentication						x		
FMT_MTD.1	x		x	x		x		
Class FPR								
FPR_ANO.1				x				x
Class FPT								
FPT_EMS.1	x							
FPT_PHP.3		x						
FPT_TST.1	x	x					x	

Table 5.1: Security Functional Requirements Rationale

5.3.2 Rationale for SFR's Dependencies

The dependency analysis for the security functional requirements shows that the basis for mutual support and internal consistency between all defined functional requirements is satisfied. All dependencies between the chosen functional components are analyzed, and non-dissolved dependencies are appropriately explained.

The dependency analysis has directly been made within the description of each SFR in the sections above. All dependencies being expected are either fulfilled, or their non-fulfillment is justified.

5.3.3 Security Assurance Requirements Rationale

The current assurance package was chosen based on the predefined assurance package EAL4. This package permits a developer to gain maximum assurance from positive security engineering based on good

commercial development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources. EAL4 is the highest level, at which it is likely to retrofit to an existing product line in an economically feasible way. EAL4 is applicable in those circumstances where developers or users require a moderate to high level of independently assured security in conventional commodity TOEs and are prepared to incur additional security specific engineering costs.

The selection of the component AVA_VAN.5 provides a higher assurance than the predefined EAL4 package, namely requiring a vulnerability analysis to assess the resistance to penetration attacks performed by an attacker possessing a high attack potential.

The set of assurance requirements being part of EAL4 fulfills all dependencies a priori. The augmentation of EAL4 chosen comprises the following assurance components: AVA_VAN.5.

For these additional assurance component, all dependencies are met or exceeded in the EAL4 assurance package. Below we list only those assurance requirements that are additional to EAL4.

AVA_VAN.5
Dependencies: ADV_ARC.1, ADV_FSP.4, ADV_TDS.3, ADV_IMP.1, AGD_OPE.1, AGD_PRE.1, ATE_DPT.1
fulfilled by ADV_ARC.1, ADV_FSP.4, ADV_TDS.3, ADV_IMP.1, AGD_OPE.1, AGD_PRE.1, ATE_DPT.1

5.3.4 Security Requirements – Internal Consistency

The following part of the security requirements rationale shows that the set of security requirements for the TOE consisting of the security functional requirements (SFRs) and the security assurance requirements (SARs) are internally consistent. The analysis of the TOE´s security requirements with regard to their mutual support and internal consistency demonstrates:

The dependency analysis above for the security functional requirements shows that the basis for internal consistency between all defined functional requirements is satisfied. All dependencies between the chosen functional components are analyzed and non-satisfied dependencies are appropriately justified.

All subjects and objects addressed by more than one SFR are also treated in a consistent way: the SFRs impacting them do not require any contradictory property or behavior of these ‘shared’ items.

The assurance package EAL4 is a predefined set of internally consistent assurance requirements. The dependency analysis for the sensitive assurance components in Section 5.3.3 shows that the assurance requirements are internally consistent as all (additional) dependencies are satisfied and no inconsistency appears.

Inconsistency between functional and assurance requirements can only arise due to functional-assurance dependencies not being met. As shown above, the chosen assurance components are adequate for the functionality of the TOE. Hence, there are no inconsistencies between the goals of these two groups of security requirements.

6 TOE Summary Specifications

6.1 Security Functionality

The TOE offers the following security functions to realize the security requirements.

6.1.1 SF_StrongAuthentication

FIDO Processes

The TOE provides strong authentication by defining and control the four processes

- initialization (FDP_IFC.1/Initialisation, FDP_IFF.1/Initialisation),
- registration (FDP_IFC.1/Registration, FDP_IFF.1/Registration),
- authentication (FDP_IFC.1/Authentication, FDP_IFF.1/Authentication) and
- reset (FDP_IFC.1/Reset, FDP_IFF.1/Reset).

The corresponding source code functions display the informal process description of chapter 1.2.1.

The TOE executes any process based on the subjects and information security attributes indicated separately for each process (FDP_IFF .1.1). The TOE permits an information flow between a controlled subject and a controlled information via a controlled operation for the respective rules (FDP_IFF .1.2). The TOE enforces the particular rules in their specific order (FDP_IFF .1.3). Other rules for authorized information flows are explicitly not foreseen (FDP_IFF .1.4). The TOE prohibits that MACKey, seed or private keys leave the TOE at any time (FDP_IFF .1.5).

To ensure the secure and authorized use of these functions, security management is needed:

The TOE defines the functions creation and destruction of the MACKey and the seed as well as resetting the authenticator to delivery state. The covered security functional requirements is FMT_SMF.1.

The TOE defines the role users and authenticator with the security functional requirement FMT_SMR.1

The TOE restricts the ability to change_default, modify and clear the MACKey, seed and private keys to the role user. This is covered with the security functional requirement FMT_MTD.1.

Further, the ability to query the security attributes user_presence and security_state during the four different processes is restricted to the roles authenticator and user. The covered functional requirements are FMT_MSA.1/Initialisation, FMT_MSA.1/Reset, FMT_MSA.1/Registration, FMT_MSA.1/Authentication.

The TOE enforces the information flow control instances to provide restrictive default values for security attributes. Only the authenticator is allowed to override the default values. The covered functional requirements are FMT_MSA.3/Initialisation, FMT_MSA.3/Reset, FMT_MSA.3/Registration, FMT_MSA.3/Authentication.

Crypto

The cryptographic support within the TOE is covered by several functions.

The TOE generates a MAC within U2F Registration Process over the registration data AppID and nonce. At Authentication Process the TOE verifies the MAC with the authentication data which also contains AppID and nonce. Only if the MAC is correct the TOE starts the KDF function and signs the incoming authentication data.

The generation of the MACKey and the seed is implemented by using the AESKey object of the platform filled with the function RandomData (instance of ALG_SECURE_RANDOM) of the platform. The covered security functional requirement is FCS_CKM.1.

The TOE supports the destruction of cryptographic keys by using the Java Card API method Key.clearKey(). The covered security functional requirement is FCS_CKM.4.

The TOE performs signature creation with ECDSA. The signature is used for registration and authentication. During the registration process , the key handle, AppID, public key and the challenge are signed with the private key. The key handle together with the public key and the signature is then transmitted to and stored at the relying party. In the authentication process a challenge is signed and send to the relying party. This allows the relying party to check the authenticity of the TOE. The covered security functional requirement is FCS_COP.1.

The TOE performs MAC calculation by using the AES CMAC Algorithm The CMAC output is afterwards hashed with SHA 256. The MAC is used to check integrity of the data send from the relying party to the TOE. The covered security functional requirement is FCS_COP.1/MAC.

The TOE provides a deterministic random number generator by using a RandomData instance with ALG_SECURE_RANDOM parameter which is used to generate random numbers for seed, nonce and MACKey. The covered security functional requirement is FCS_RNG.1.

The TOEs TSF data (seed and MACKey) are stored as AES Key objects provided by the platform. The TOE monitors TSF data stored in containers controlled by the TSF for integrity errors on all objects, based on checksum integrity of cryptographic keys and their associated security attributes. The covered security functional requirement is FDP_SDI.1. The platform itself protect sensitive TSF data used by the OS (AES-Keys) with various internal mechanisms.

The TOE derives the cryptographic private U2F (EC-)key from FIDO AppID and nonce by using the cryptographic key derivation function in counter mode using AES CMAC as pseudo random function (PRF) supported from the underlying platform. The seed is stored and used as AES Key. The covered security functional requirement is FCS_CKM.5/U2F-Private. The public key is derived from the private key according to ECC Key Pair Generation algorithm (FCS_CKM.5/U2F-Public). The public key is sent to the relying party during the registration process. Afterwards the public key will be deleted on the TOE.

6.1.2 SF_Unlinkability

The TOE provides unlinkability by generating unique asymmetric key pairs for each relying party and account. The covered security functional requirement is FCS_CKM.5/U2F-Private, FCS_CKM.5/U2F-Public. Unlinkability is also supported by the crypto requirements and the user management functions listed in chapter 6.1.1 (FCS_COP.1, FCS_RNG.1, FMT_SMF.1, FMT_MTD.1).

6.1.3 SF_Privacy

There is no personalization step which creates or stores data with the identity of the user. The TOE does not store any personal information and ensures that all subjects are unable to determine the real user name bound to the user. The covered security functional requirement is FPR_ANO.1.

6.1.4 SF_UserPresence

The TOE provides physical test of user presence by placing the TOE into the proximity range of an NFC-enabled device or by inserting the TOE into a card reader. The TOE requires demonstrating user presence each time when one of the four operation initialisation, registration, authentication or reset occurs (FIA_UAU.2). After each operation the user is required to remove the TOE from the proximity range of the NFC-enabled device or the card reader before placing it back to re-authenticate the user in order to perform a new operation (FIA_UAU.6).

6.1.5 SF_TSF-Protection

The TOE provides TSF-Protection. This functionality is covered by the Hardware or the underlying platform (see chapter 9.1.5 SF.SECURITY [SCST]). The TSF ensures resistance to physical tampering using features against probing and an active shield detecting integrity violation.

1. The TOE detects physical tampering of the TSF with sensors for operating voltage, clock frequency, temperature and electromagnetic radiation. It is resistant to physical tampering of the TSF. If the TOE detects with the above mentioned sensors that it is not supplied within the specified limits, a security reset is initiated and the TOE is not operable until the supply is back in the specified limits. The design of the hardware protects it against analysing and physical tampering.

2. The TOE hides information about IC power consumptions and command execution time, to ensure that no confidential information can be derived from this data.

The covered security functional requirement is FPT_PHP.3. (see chapter 9.1.5 SF.SECURITY 4./5. [SCST]).

The TOE prevents emission of usable information on key material over side-channels like power or timing variations and electromagnetic emanations (FPT_EMS.1).

The TOE runs packages checksum integrity tests during initial start-up at each power on of the TOE to demonstrate the correct operation of the TSF (FPT_TST.1). (see chapter 9.1.4 SF.INTEGRITY 4. [SCST]).

6.2 TOE Summary Specification Rationale

	SF_StrongAuthentication	SF_Unlinkability	SF_Privacy	SF_UserPresence	SF_TSF-Protection
FDP_IFC.1/Initialisation	x				
FDP_IFF.1/Initialisation	x				
FDP_IFC.1/Reset	x				
FDP_IFF.1/Reset	x				
FDP_IFC.1/Registration	x				
FDP_IFF.1/Registration	x				
FDP_IFC.1/Authentication	x				
FDP_IFF.1/Authentication	x				
FMT_SMF.1	x	x			
FMT_SMR.1	x				
FMT_MTD.1	x	x			
FMT_MSA.1/Initialisation	x				
FMT_MSA.3/Initialisation	x				

FMT_MSA.1/Reset	x				
FMT_MSA.3/Reset	x				
FMT_MSA.1/Registration	x				
FMT_MSA.3/Registration	x				
FMT_MSA.1/Authentication	x				
FMT_MSA.3/ Authentication	x				
FCS_CKM.1	x				
FCS_CKM.4	x				
FCS_COP.1	x				
FCS_COP.1/MAC	x	x			
FCS_RNG.1	x	x			
FDP_SDI.1	x				
FCS_CKM.5/U2F-Private	x	x			
FCS_CKM.5/U2F-Public	x	x			
FPR_ANO.1			x		
FIA_UAU.2				x	
FIA_UAU.6				x	
FPT_EMS.1					x
FPT_PHP.3					x
FPT_TST.1					x

Table 6.1: TSS Rationale

6.3 Cryptographic Mechanisms Implemented in the TOE

No.	Purpose	Cryptographic Mechanism	Implementation Standard	Key Size in Bits	Comments
1	Key generation	AES	AIS 20/DRG.4 [AIS 20] NIST [SP800-133](chapter 7.1)	128	Key generation seed and MACKey
2	Authentication	ECDSA SHA 256	NIST Fips 186 -4 Curve NIST P-256 [FIPS186-4] SHA 256 Hash algorithm [FIPS 180-4]	256	Signature creation
3	Random Number generation	DRNG	AIS 20/DRG.4 [AIS 20]		Input for AES Keys seed and MACKey
4	Key Derivation Function	KDF in counter mode with AES-CMAC as PRF	[SP800-38B] (CMAC), [SP800-108](KDF), [FIPS	128	Private Key generation

			197] (AES)		
5	Private Key generation	ECDSA using KDF	NIST Fips 186 -4 Curve NIST P-256 [FIPS186-4], KDF (no.4), [ADV_ARC]	256	Private Key generation
6	Public Key generation	ECC Key Pair Generation	Key Pair Generation [FIPS 180-4]	256	Public Key generation
7	Authentication	AES-CMAC SHA 256	[SP800-38B] (CMAC), [FIPS 197] (AES), [FIPS 180-4] (SHA 256 Hash algorithm), [ADV_ARC]	128	MAC generation

Table 6.2: Cryptographic Mechanisms

6.4 Statement of Compatibility

This is a statement of compatibility between this Composite Security Target (Composite-ST) and the Platform Security Target (Platform-ST) [SCST]. This statement is compliant to the requirements of [SUPP].

6.4.1 Assessment of the Platform TSFs

The following table lists all Security Functionalities of the underlying Platform ST and shows, which Security Functionalities of the Platform ST are relevant for this Composite ST and which are irrelevant. The first column addresses specific Security Functionality of the underlying platform, which is assigned to Security Functionalities of the Composite ST in the second column. The last column provides additional information on the correspondence if necessary.

Platform TSF-group	Correspondence in this ST	References/Remarks
SF.TRANSACTION	No correspondence, internal Java Card mechanisms.	This security function provides atomic transactions according to the Java Card Transaction and Atomicity mechanism with commit and rollback capability for updating persistent data in FLASH memory.
SF.ACCESS_CONTROL	No correspondence, internal Java Card mechanisms.	This security function provides control for the TOE. It is in charge of the FIREWALL access control SFP and the JCVM information flow control SFP.
SF.CRYPTO	SF.StrongAuthentication (section Crypto)	This security function controls all the operations related to the cryptographic key management and cryptographic operations.
SF.INTEGRITY	SF.StrongAuthentication (section Crypto)	This security function provides a means to check the integrity of checksummed data stored in FLASH memory.
SF.SECURITY	SF.TSF_Protection	This security function ensures a secure state of information, the non-observability of operations on it and the unavailability of previous information content upon deallocation.
SF.APPLET	No correspondence, internal Java Card mechanisms.	This security function ensures the secure loading of a package or installing of an applet and the secure deletion of applets and/or packages.
SF.CARRIER	No correspondence, internal Java Card mechanisms.	This security function ensures secure downloading of applications on the card.

Table 6.3: Assessment of platform TSFs

6.4.2 Assessment of the Platform Security Requirements

The following table provides an assessment of all Platform SFRs and their correspondence in this ST.

Platform SFR	Correspondence in this ST	References/Remarks
FDP_ACC.2/FIREWALL	No correspondence	Out of scope (internal Java Card Firewall). No contradiction to this ST.
FDP_ACF.1/FIREWALL	No correspondence	Out of scope (internal Java Card Firewall). No contradiction to this ST.
FDP_IFC.1/JCVM	No correspondence	Out of scope. No contradiction to this ST.
FDP_IFF.1/JCVM	No correspondence	Out of scope. No contradiction to

		this ST.
FDP_RIP.1/OBJECTS	No correspondence	Out of scope. No contradiction to this ST.
FMT_MSA.1/JCRE	No correspondence	Out of scope. No contradiction to this ST.
FMT_MSA.1/JCVM	No correspondence	Out of scope. No contradiction to this ST.
FMT_MSA.2/FIREWALL_JCVM	No correspondence	Out of scope (internal Java Card Firewall). No contradiction to this ST.
FMT_MSA.3/FIREWALL	No correspondence	Out of scope (internal Java Card Firewall). No contradiction to this ST.
FMT_MSA.3/JCVM	No correspondence	Out of scope. No contradiction to this ST.
FMT_SMF.1	No correspondence	Out of scope. The defined management functions are Javacard internal). No contradiction to this ST.
FMT_SMR.1	No correspondence	Out of scope. The defined security roles are Javacard internal roles. No contradiction to this ST.
FCS_CKM.1/RSA	No correspondence	Out of scope. RSA is not used in this TOE. No contradiction to this ST.
FCS_CKM.1/ECC	No correspondence	Out of scope. ECC is not used in this TOE. No contradiction to this ST.
FCS_CKM.1/3DES	No correspondence	Out of scope. 3DES is not used in this TOE. No contradiction to this ST.
FCS_CKM.1/AES	FCS_CKM.1	The requirement FCS_CKM.1 of this ST targets the cryptographic key generation (seed and MACKey) and is fulfilled by the platform SFR FCS_CKM.1/AES
FCS_CKM.2	No correspondence	Out of scope. Key distribution is not used in this TOE. No contradiction to this ST.
FCS_CKM.3	No correspondence	Out of scope. No key access is needed in this TOE. No contradiction to this ST.
FCS_CKM.4	FCS_CKM.4	The requirements are compatible (physically overwriting the keys)
FCS_COP.1.1/RSA-CRT-SIGN	No correspondence	Out of scope. No contradiction to this ST.

FCS_COP.1. 1/RSA-SIGN	No correspondence	Out of scope. No contradiction to this ST.
FCS_COP.1.1/RSA-VERI	No correspondence	Out of scope. No contradiction to this ST.
FCS_COP.1.1/MAC-DES	No correspondence	Out of scope. No contradiction to this ST.
FCS_COP.1.1/MAC-AES	No correspondence	Out of scope. No contradiction to this ST.
FCS_COP.1.1/CMAC-AES	FCS_COP.1/MAC	MAC generation requirements are compatible
FCS_COP.1.1/3DES	No correspondence	Out of scope. No contradiction to this ST.
FCS_COP.1.1/AES	No correspondence	Out of scope. No contradiction to this ST.
FCS_COP.1.1/RSA-DEC	No correspondence	Out of scope. No contradiction to this ST.
FCS_COP.1.1/RSA-CRT-DEC	No correspondence	Out of scope. No contradiction to this ST.
FCS_COP.1.1/RSA-ENC	No correspondence	Out of scope. No contradiction to this ST.
FCS_COP.1.1/ECDSA-SIGN	FCS_COP.1	Signature creation requirements by using ECDSA is compatible
FCS_COP.1.1/ ECDSA-VERI	No correspondence	Out of scope. No contradiction to this ST.
FCS_COP.1.1/HASH	FCS_COP.1/MAC	The requirement of the ST uses the the hash function of the platform.
FCS_RNG.1.1	FCS_RNG.1.1	This ST uses the Deterministic Random Number Generator of the Chip according to AIS 20 in the FCS_RNG.1.1 SFR
FDP_RIP.1/ABORT	No correspondence	Out of scope. No contradiction to this ST.
FDP_RIP.1/APDU	No correspondence	Out of scope. No contradiction to this ST.
FDP_RIP.1/bArray	No correspondence	Out of scope. No contradiction to this ST.
FDP_RIP.1/KEYS	No correspondence	Out of scope. No contradiction to this ST.
FDP_RIP.1/TRANSIENT	No correspondence	Out of scope. No contradiction to this ST.
FDP_ROL.1/FIREWALL	No correspondence	Out of scope (internal Java Card Firewall). No contradiction to this ST.
FAU_ARP.1	No correspondence	Out of scope. No contradiction to

		this ST.
FDP_SDI.2	FDP_SDI.1	The requirement of this ST uses the integrity monitoring of the platform.
FPR_UNO.1	FPT_EMS.1	Not directly corresponding, but relevant for the fulfillment of FPT_EMS.1. No contradiction to this ST.
FPT_FLS.1	No correspondence	Out of scope. No contradiction to this ST.
FPT_TDC.1	No correspondence	Out of scope. No contradiction to this ST.
FIA_ATD.1/AID	No correspondence	Out of scope. No contradiction to this ST.
FIA_UID.2/AID	No correspondence	Out of scope. No contradiction to this ST.
FIA_USB.1/AID	No correspondence	Out of scope. No contradiction to this ST.
FMT_MTD.1/JCRE	No correspondence	Out of scope. No contradiction to this ST.
FMT_MTD.3/JCRE	No correspondence	Out of scope. No contradiction to this ST.
FDP_ITC.2/Installer	No correspondence	Out of scope. No contradiction to this ST.
FMT_SMR.1/Installer	No correspondence	Out of scope. No contradiction to this ST.
FPT_FLS.1/Installer	No correspondence	Out of scope. No contradiction to this ST.
FPT_RCV.3/Installer	No correspondence	Out of scope. No contradiction to this ST.
FDP_ACC.2/ADEL	No correspondence	Out of scope. No contradiction to this ST.
FDP_ACF.1/ADEL	No correspondence	Out of scope. No contradiction to this ST.
FDP_RIP.1/ADEL	No correspondence	Out of scope. No contradiction to this ST.
FMT_MSA.1/ADEL	No correspondence	Out of scope. No contradiction to this ST.
FMT_MSA.3/ADEL	No correspondence	Out of scope. No contradiction to this ST.
FMT_SMF.1/ADEL	No correspondence	Out of scope. No contradiction to this ST.
FMT_SMR.1/ADEL	No correspondence	Out of scope. No contradiction to this ST.

FPT_FLS.1/ADEL	No correspondence	Out of scope. No contradiction to this ST.
FDP_RIP.1/ODEL	No correspondence	Out of scope. No contradiction to this ST.
FPT_FLS.1/ODEL	No correspondence	Out of scope. No contradiction to this ST.
FCO_NRO.2/CM	No correspondence	Out of scope. No contradiction to this ST.
FDP_IFC.2/CM	No correspondence	Out of scope. No contradiction to this ST.
FDP_IFF.1/CM	No correspondence	Out of scope. No contradiction to this ST.
FDP_UIT.1/CM	No correspondence	Out of scope. No contradiction to this ST.
FIA_UID.1/CM	No correspondence	Out of scope. No contradiction to this ST.
FMT_MSA.1/CM	No correspondence	Out of scope. No contradiction to this ST.
FMT_MSA.3/CM	No correspondence	Out of scope. No contradiction to this ST.
FMT_SMF.1/CM	No correspondence	Out of scope. No contradiction to this ST.
FMT_SMR.1/CM	No correspondence	Out of scope. No contradiction to this ST.
FTP_ITC.1/CM	No correspondence	Out of scope. No contradiction to this ST.
FTP_ITC.1/CMGR	No correspondence	Out of scope. No contradiction to this ST.
FPT_PHP.3	FPT_PHP.3 FPT_EMS.1	The fulfillment of the SFR in this ST is based on the platform SFR (together with additional countermeasures).
FPT_TST.1	FPT_TST.1	Self-testing is provided by the Java Card platform during initial start-up.

Table 6.4: Assessment of Platform SFRs

6.4.3 Assessment of the Platform Assurance Requirements

The Composite-ST requires EAL 4 augmented by: AVA_VAN.5.

The Platform-ST requires EAL 5 augmented by: ALC_DVS.2 and AVA_VAN.5.

Therefore, the assurance requirements for the composite TOE are a subset of the assurance requirements of the underlying platform.

6.4.4 Assessment of the Platform Objectives

The following table provides an assessment of all relevant Platform objectives.

Platform Objectives	Correspondence in this ST	References/Remarks
O.SID	No correspondence	Out of scope. No contradiction to this ST.
O.FIREWALL	No correspondence	Out of scope. No contradiction to this ST.
O.GLOBAL_ARRAYS_CONFID	No correspondence	Out of scope. No contradiction to this ST.
O.GLOBAL_ARRAYS_INTEG	No correspondence	Out of scope. No contradiction to this ST.
O.NATIVE	No correspondence	Out of scope. No contradiction to this ST.
O.OPERATE	Relevant for all objectives. But no direct correspondence.	No contradiction to this ST.
O.REALLOCATION	No correspondence	Out of scope. No contradiction to this ST.
O.RESOURCES	No correspondence	Out of scope. No contradiction to this ST.
O.ALARM	OT.TamperResistance, OT.Prot_Abuse-Func	Objectives are related. No contradiction to this ST.
O.CIPHER	OT.HighLevelOfAssurance OT.Unlinkability OT.TrustworthyData OT.AuthenticatorLeakResilience OT.LimitedPII	Objectives are related. No contradictions to this ST.
O.KEY-MNGT	OT.HighLevelOfAssurance OT.Unlinkability	Objectives are related. No contradictions to this ST.
O.PIN-MNGT	No correspondence	Out of scope. No contradiction to this ST.
O.TRANSACTION	No correspondence	Out of scope. No contradiction to this ST.
O.OBJ-DELETION	No correspondence	Out of scope. No contradiction to this ST.
O.DELETION	No correspondence	Out of scope. No contradiction to this ST.
O.LOAD	No correspondence	Out of scope. No contradiction to this ST.
O.INSTALL	No correspondence	Out of scope. No contradiction to this ST.
O.CARD-MANAGEMENT	No correspondence	Out of scope. No contradiction to this ST.

O.SCP.IC	OT.LeakageResistance, OT.TamperResistance, OT.Prot_Abuse-Func	Objectives are related. No contradiction to this ST.
O.SCP.RECOVERY	No correspondence	Out of scope. No contradiction to this ST.
O.SCP.SUPPORT	No correspondence	Out of scope. No contradiction to this ST.

Table 6.5: Assessment of Platform Objectives

6.4.5 Assessment of the Platform Threats

The following table provides an assessment of all relevant Platform threats.

Platform Threats	Correspondence in this ST	References/Remarks
T.CONFID-APPLI-DATA	No correspondence	Out of scope. No contradiction to this ST.
T.CONFID-JCS-CODE	No correspondence	Out of scope. No contradiction to this ST.
T.CONFID-JCS-DATA	No correspondence	Out of scope. No contradiction to this ST.
T.INTEG-APPLI-CODE	No correspondence	Out of scope. No contradiction to this ST.
T.INTEG-APPLI-CODE.LOAD	No correspondence	Out of scope. No contradiction to this ST.
T.INTEG-APPLI-DATA	No correspondence	Out of scope. No contradiction to this ST.
T.INTEG-APPLI-DATA.LOAD	No correspondence	Out of scope. No contradiction to this ST.
T.INTEG-JCS-CODE	No correspondence	Out of scope. No contradiction to this ST.
T.INTEG-JCS-DATA	No correspondence	Out of scope. No contradiction to this ST.
T.SID.1	No correspondence	Out of scope. No contradiction to this ST.
T.SID.2	No correspondence	Out of scope. No contradiction to this ST.
T.EXE-CODE.1	No correspondence	Out of scope. No contradiction to this ST.
T.EXE-CODE.2	No correspondence	Out of scope. No contradiction to this ST.
T.NATIVE	No correspondence	Out of scope. No contradiction to this ST.
T.RESOURCES	No correspondence	Out of scope. No contradiction to this ST.

T.DELETION	No correspondence	Out of scope. No contradiction to this ST.
T.SECURE_DELETION	No correspondence	Out of scope. No contradiction to this ST.
T.INSTALL	No correspondence	Out of scope. No contradiction to this ST.
T.OBJDELETION	No correspondence	Out of scope. No contradiction to this ST.
T.PHYSICAL	T.InformationLeakage T.PhysTamper T.KeyCompromise	No contradiction to this ST.

Table 6.6: Assessment of Platform Threats

6.4.6 Assessment of the Platform Organisational Security Policies

The Organisational Security Policy “OSP.VERIFICATION” focuses on the integrity of loaded applets, which is fulfilled by the TOE of this ST since the applet is loaded secured by platform security measures into the flash memory. This policy does not contradict to the policies of this ST.

6.4.7 Assessment of the Platform Assumptions

The following table provides an assessment of all relevant Platform assumptions.

Platform Assumptions	References/Remarks
A.APPLET	A.APPLET states that applets loaded post-issuance do not contain native methods. Assumption is not relevant for the TOE, as no applets can be loaded post-issuance.
A.VERIFICATION	This assumption targets the applet code verification. No third party applets can be loaded post-issuance.

Table 6.7: Assessment of Platform Assumptions

6.4.8 Assessment of the Platform Objectives for the Operational Environment

The following table provides an assessment of all relevant Platform objectives for the operational environment.

Platform Objectives for the Operational Environment	References/Remarks
OE.APPLET	The platform objective for the environment states that applets loaded post-issuance do not contain native methods. Because no third party applets can be installed on the TOE of this ST, this objective is fulfilled.
OE.VERIFICATION	The platform objective for the environment targets the applet code verification. This is fulfilled by the TOE of this ST; no third-party-code can be installed on the TOE
OE.CODE-EVIDENCE	The platform objective for the environment focusses on application code loaded pre-issuance or post-issuance. After code verification code cannot be changed. The development environment of this TOE assures this

	objective. Changes post-issuance are not possible because no third-party applets can be loaded on the TOE.

Table 6.8: Assessment of Platform Objectives for the Operational Environment

Glossary

FIDO User Device The computing device where the FIDO Client operates, and from which the user initiates an action that utilizes FIDO.

Computing Environment The user's computer that is interacting with the FIDO Token.

For further FIDO related terms see also the „FIDO Technical Glossary“ of [FIDOSpec].

Reference Documentation

FIDOPP	Bundesamt für Sicherheit in der Informationstechnik, Common Criteria Protection Profile - FIDO Universal Second Factor (U2F), BSI-CC-PP-0096-V3-2018, 5. November 2018, v3.0
SCST	Giesecke+Devrient Mobile Security GmbH, Security Target Lite Sm@rtCafé® Expert 7.0 C3, 16. August 2017, v2.9
MA-SCST	Giesecke+Devrient Mobile Security GmbH, Assurance Continuity Maintenance Report BSI-DSZ-CC-1028-2017-MA-01Sm@rtCafé® Expert 7.0 C3, 4.10.2018,
ICST	Infineon Technologies AG, Security Target Lite for M5073 G11 Common Criteria CCv3.1 EAL6 augmented (EAL6+), BSI-DSZ-CC-0951-2015-RA-01, 10. Mai 2017, v1.2
GUIDANCE	Bundesamt für Sicherheit in der Informationstechnik, Administration and User Guide, 14.06.2022, v1.4
FIDOSpec	FIDO Alliance, Fido Alliance Universal 2nd Factor 1.2 Specifications, 11. July 2017, https://fidoalliance.org/specifications/download/
FIDOSecRef	FIDO Alliance, FIDO Security Reference, 27 February 2018, https://fidoalliance.org/specs/fido-v2.0-id-20180227/fido-security-ref-v2.0-id-20180227.html#attack-classification
ICPP	Inside Secure, Infineon Technologies AG, NXP Semiconductors Germany GmbH, STMicroelectronics, Common Criteria Protection Profile - Security IC Platform Protection Profile with Augmentation Packages, BSI-CC-PP-0084-2014, 13.01.2014, v1.0
ICCR-RA	Federal Office for Information Security, Assurance Continuity Reassessment Report BSI-DSZ-CC-0951-2015-RA-01, Infineon Security Controller M5073 G11 with optional RSA2048/4096 v2.03.008, EC v2.03.008,SHA-2 v1.01 and Toolbox v2.03.008 libraries and with specific IC dedicated software (firmware) from Infineon Technologies AG, 31. Mai 2017,
SCCR	Federal Office for Information Security, Certification report BSI-DSZ-CC-1028-2017 for Sm@rtCafé® Expert 7.0 C3 from Veridos GmbH - Identity Solutions by G&D BDR, 08. September 2017,
CC1	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, April 2017, Version 3.1, Revision 5, CCMB-2017-04-001
CC2	Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, April 2017, Version 3.1, Revision 5, CCMB-2017-04-002
CC3	Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, April 2017, Version 3.1, Revision 5, CCMB-2017-04-003
CEM	Common Methodology for Information Technology Security Evaluation, Evaluation methodology, April 2017, Version 3.1, Revision 5, CCMB-2017-04-004
CC2	Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components; CCMB-2012-09-002, September 2012, Version 3.1, Revision 4
CC1	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2012-09-001, September 2012, Version 3.1, Revision 4
FIDOCrypt	FIDO Alliance, FIDO Authenticator Allowed Cryptography List, 24. May 2017, https://fidoalliance.org/wp-content/uploads/fido-authenticator-allowed-cryptography-list-2.pdf
AIS 20	Bundesamt fuer Sicherheit in der Informationstechnik, Funktionsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren, Version 3.0, 2013-05-15,
SP800-133	NIST, Recommendation for Cryptographic Key Generation, December 2012,

JCAPI304	Oracle, Java Card API, Classic Edition, 2011, v3.0.4
FIPS186-4	National Institute of Standards and Technology, FIPS PUB 186-4: Digital Signature Standard (DSS), July 2013, http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
FIPS 180-4	National Institute of Standards and Technology, FIPS PUB 180-4: Secure Hash Standard (SHS), August 2015,
SP800-38B	M. Dworkin, NIST Special Publication 800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, October 2016, https://doi.org/10.6028/NIST.SP.800-38B
FIPS 197	National Institute of Standards andTechnology, Federal Information Processing Standards Publication 197, Advanced Encryption Standard, November 26, 2001,
SP800-108	Lily Chen, NIST Special Publication 800-108: Recommendation for Key Derivation Using Pseudorandom Functions, October 2009, http://doi.org/10.6028/NIST.SP.800-108
ADV_ARC	Bundesamt fur Sicherheit in der Informationstechnik, Developer Documentation ARC - Security Architecture, 2019, v1.1
SUPP	Supporting Document, Mandatory Technical Document, Composite product evaluation for Smart Cards and similar devices, December 2015, v1.4, CCDB-2015-12-001