Federal Office
for Information Security

Developer Documentation TDS - TOE Design Description

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

1.1 (3. Apr. 2019)

Federal Office for Information Security
Post Box 20 03 63
D-53133 Bonn
Internet: https://www.bsi.bund.de
© Federal Office for Information Security 2020

Table of Contents

Figures

Tables

1 Subsystem and Module description

The subsystems and modules described herein all map directly to methods in the TOE source code and are documented in the source code documentation ([de.fac2 JavaDoc]) or are provided by the platforms API [JCAPI304]. Detailed information such as purpose, method of use, parameters, parameter description and description of the interface's actions are described in the source code documentation. Additional information like module interaction and relation to SFR are described in this chapter.

The only external interface of the TOE is the APDU interface which is controlled by the Subsystem: JavaCard platform. If the de.fac2 applet is selected and JavaCard platform receives a APDU the Subsystem: JavaCard platform will call the applets Module: U2FApplet method in Subsystem: Control & Communication subsystem. The process method will dispatch the incoming Command APDU to the matching module in Subsytem 1. Most modules in Subsystem: Control & Communication subsystem will call modules in Subsystem: Crypto subsystem for cryptographic processing (e.g. key generation, MAC calculation, key deletion). The crypto operation results will be return to the calling module in Subsystem: Control & Communication subsystem. Here the results will be build into FIDO conform Response APDUs and be returned by using the APDU module of Subsystem: JavaCard platform

Modules described herein are implemented as Java classes. The interface in the modules are implemented methods.

1.1 Subsystem: Control & Communication subsystem

This subsystem contains all modules which handle the input and output data which comes into the system via the TSFIs. The modules described in this subsystem parse the data of the incoming command APDUs, call the appropriate modules in the cryptographic subsystem and build the response APDU data as specified in the [FIDOSpec].

All communication from the TSFIs will be handled in this subsystem. Incoming and outgoing data will be processed herein. This contains checks of correct formatted and FIDO conform commands. If any check fails the data will not be processed or relayed to other subsystems. Also, the data that returns from other subsystems will first be processed and built to a correct formatted response which is conform to the [FIDOSpec].

FIDO specific behavior is controlled and processed in this subsystem. It checks the user presence and increments the internal FIDO counter every time it performs an authentication process.

Also the internal state of the applet and its behavior depending on the state will be controlled in this module.

This subsystem is categorized as **SFR-enforcing** because it contains SFR-enforcing modules.

1.1.1 Module: U2FApplet

This Module instantiates the U2F de.fac2 Javacard applet and implements all interfaces that are needed to receive the calls from the underlying Javacard Platform(Subsystem: JavaCard platform) and additional interface for the Fido U2F functionality.

Incoming APDU will be processed in this module by the Interface: process. Depending on the content of the APDU it will be dispatched to further interfaces in this module. For cryptographic operations this module will call interfaces in Module: FIDO-U2F-Implementation.

This module implements interfaces that are categorized as SFR-enforcing. Therefore, it is also categorized as **SFR-enforcing**.

This Module implements the following interfaces:

Interface: install

This interface will be called from the underlying JavaCard platfrom only one time when the applet will be installed onto the card. This interface can't be called from users either it can't be called once the TOE is initalized.

This interface calls the constructor of the applet and initializes all resources that are needed for using the applet in operational state. The input parameter bArray contains a concatenation of "user presence checkflag", size of the attestation certificates private key and the private key of attestation certificate. They will be saved in secure key storage provided by the platform. At the end of the initialization process an instance of the crypto subsystem is created (see chapter 1.2) and the internal state is set to "uninitialized".

Because this interface can't be called from users either it can't be called once the TOE is initialized, this interface is categorized as **SFR-non-interfering**.

See method **public static void install(byte[] bArray, short bOffset, byte bLength)** in [de.fac2 JavaDoc] for details and parameters of this interface.

Interface: process

This interface processes all incoming APDUs and dispatches them to other interfaces. All data will be checked and only processed if they contain expected values as specified in [FIDOSpec].

Before any incoming command will be executed the interface will check if the requested command is allowed to be performed at the actual internal state. All FIDO commands and the reset command are only allowed in the operational state, while setting the attestation certificate is only allowed in uninitialized state.

Depending on the command which was requested, the interface calls different other interfaces (see interfaces with naming handleXYZ).

Because this interface checks the internal state of the applet before dispatching incoming APDUs it is categorized as **SFR-enforcing**.

See method **public void process(javacard.framework.APDU apdu)** in [de.fac2 JavaDoc] for details and parameters of this interface.

Interface: handleRegister

This interface will be called from Interface: process if the incoming APDU is a U2F Registration request. It will call Interface: getPubKeyAndKeyHandle and build a response APDU which will be returned.

This interface performs the user presence check and does a signing operation. Therefore this interface is categorized as **SFR-enforcing**.

See method **private void handleRegister(javacard.framework.APDU apdu)** in [de.fac2 JavaDoc] for details and parameters of this interface.

Interface: handleAuthenticate

This interface will be called from Interface: process if the incoming APDU is a U2F Authentication request. It will call Interface: getSigningKey and build a response APDU which will be returned.

This interface performs the user presence check and does a signing operation. Therefore this interface is categorized as **SFR-enforcing**.

See method **private void handleAuthenticate(javacard.framework.APDU apdu)** in [de.fac2 JavaDoc] for details and parameters of this interface.

Interface: handleVersion

This interface will be called from Interface: process if the incoming APDU is a U2F Get Version request. It will build a response APDU contains the U2F version string.

This interface performs no SFR relevant operation. Therefore this interface is categorized as **SFR-non-interfering**.

See method **private void handleVersion(javacard.framework.APDU apdu)** in [de.fac2 JavaDoc] for details and parameters of this interface.

Interface: handleGetResponse

This interface will be called from Interface: process if the incoming APDU is a Get Response request. It will build a response APDU which returns remaining response data which was generated from previous calls of Interface: handleRegister or Interface: handleAuthenticate.

Because this interface is just a helper for SRF-enforcing interfaces, it is categorized as **SFR-supporting**.

See method **private void handleGetResponse(javacard.framework.APDU apdu)** in [de.fac2 JavaDoc] for details and parameters of this interface.

Interface: handleReset

This interface will be called from Interface: process if the incoming APDU is a Reset request. It will call Interface: eraseKeys and Interface: initializeKeys to reset the internal keys.

This interface performs the user presence check, securely deletes the internal keys and reinitializes the internal keys. Therefore this interface is categorized as **SFR-enforcing**.

See method **private void handleReset(javacard.framework.APDU apdu)** in [de.fac2 JavaDoc] for details and parameters of this interface.

Interface: handleSetAttestationCert

This interface will be called from Interface: process if the incoming APDU is a Set Attestation Certificate request. It can be only called when the card is in uninitialized state to upload and store the FIDO attestation certificate in the TOE. After the certificate was uploaded the internal keys will be generated and internal state will be set to ready-to-use. Then this interface will be disabled for any further usage.

This interface checks and modifies the internal state of the TOE before reaching the state DELIVERY_STATE and is not active afterwards. It is therefore categorized as **SFR-non-interfering**.

See method **private void handleSetAttestationCert(javacard.framework.APDU apdu)** in [de.fac2 JavaDoc] for details and parameters of this module.

1.2 Subsystem: Crypto subsystem

This subsystem contains the modules which provides the cryptographic functions to Subsystem: Control & Communication subsystem. Herein modules perform the key initialization, key derivation and MAC calculation by using the underlying platforms TSFs.

This subsystem is categorized as **SFR-enforcing** because it contains SFR-enforcing modules.

1.2.1 Module: FIDO-U2F-Implementation

This module contains all interfaces which are used from Module: U2FApplet for cryptographic operations on incoming data.

When this module is instantiated, all needed resources from the underlying platform will be instantiated. This includes Cipher and Signature objects as well as the AES key objects seed (CMAC Key for KDF), MACKey and the FIDO EC Keypair.

This module implements interfaces that are categorized as SFR-enforcing. Therefore, it is also categorized as **SFR-enforcing**.

This Module implements the following interfaces:

Interface: generatePrivateKey

This private interface will be called from Interface: getPubKeyAndKeyHandle and Interface: getSigningKey to derive the FIDO EC private key. For a new private key derivation (used at the registration process) the DRGof the underlying platform is used to generate a nonce. For the authentication process an existing nonce will be used. The nonce is used together with the incoming data (AppId) as input for a Key Derivation Function (KDF) function. The output of the KDF is used as private key. The platform AES-CMAC function will be used as KDF algorithm.

This interface uses the platforms cryptographic functions and derivative U2F keys which is a security requirement. Therefore it is categorized as SFR-enforcing.

See method **private void generatePrivateKey(byte[] applicationParameter, short applicationParameterOffset, byte[] nonceBuffer, short nonceBufferOffset, javacard.security.ECPrivateKey privKey)** in [de.fac2 JavaDoc] for details and parameters of this interface.

Interface: calcMAC

This private interface will be called from Interface: generatePublicKeyPoint and Interface: getSigningKey for MAC calculation over the given input data. The platform AES-CMAC function will be used as MAC algorithm. Input data for this interface are AppId and nonce. The CMACs key is an AES Key.

This interface uses the platforms cryptographic functions and performs MAC calculation which is a security requirement. Therefore it is categorized as **SFR-enforcing**.

See method **private short calcMAC(byte[] applicationParameter, short applicationParameterOffset, byte[] nonceBuffer, short nonceBufferOffset, byte[] mac, short macOffset)** in [de.fac2 JavaDoc] for details and parameters of this interface.

Interface: eraseKeys

This interface will be called from Interface: handleReset to securely clear the card individual key by using the underlying platforms function. This interface will only be processed if the applet is in the state READY_FOR_USE.

This interface uses the platforms cryptographic functions for secure key destruction which is a security requirement. Therefore it is categorized as **SFR-enforcing**.

See method **public short eraseKeys(short actualState)** in [de.fac2 JavaDoc] for details and parameters of this interface.

Interface: initializeKeys

This interface will be called from Interface: handleReset and Interface: handleSetAttestationCert to fill the card individual key objects with random key values with help of the RNG of the underlying platform. This interface will only be processed if the applet is in the state DELIVERY_STATE. Otherwise the keys are kept untouched.

This interface uses the platforms functions for key generation which is a security requirement. Therefore it is categorized as **SFR-enforcing**.

See method **public short initializeKeys(short actualState)** in [de.fac2 JavaDoc] for details and parameters of this interface.

Interface: generatePublicKeyPoint

This private interface will be called from Interface: getPubKeyAndKeyHandle to derive the public key from the private key by using the platforms elliptic curve point multiplication.

This interface uses the platforms functions for generating a public key which is a security requirement. Therefore it is categorized as **SFR-enforcing**.

See method **private short generatePublicKeyPoint(javacard.security.ECPrivateKey privKey, byte[] pointOutputBuffer, short offset)** in [de.fac2 JavaDoc] for details and parameters of this interface.

Interface: getPubKeyAndKeyHandle

This interface will be called from Interface: handleRegister to derive the FIDO EC public key and the keyHandle which is use as response to an U2F Registation request. This interface will call the Interface: generatePrivateKey and Interface: generatePublicKeyPoint to get the FIDO EC public key and Interface: calcMAC for generating parts of the keyHandle.

This interface uses the platforms TRNG for generating a nonce and uses secure key clearing function after generating the FIDO EC private key which is a security requirement. Therefore it is categorized as **SFR-enforcing**.

See method **public short getPubKeyAndKeyHandle(byte[] applicationParameter, short applicationParameterOffset, byte[] publicKey, short publicKeyOffset, byte[] keyHandle, short keyHandleOffset)** in [de.fac2 JavaDoc] for details and parameters of this interface.

Interface: getSigningKey

This interface is called from Interface: handleAuthenticate to generate the FIDO EC private keys. Before generating the private key by calling Interface: generatePrivateKey this interface checks if the keyHandle contains the correct MAC by calling Interface: calcMAC. If the MAC is correct the private key will be returned.

This interface compares securely the given MAC with the calculated MAC which is a security requirement. Therefore it is categorized as **SFR-enforcing**.

See method **public boolean getSigningKey(byte[] keyHandle, short keyHandleOffset, short keyHandleLength, byte[] applicationParameter, short applicationParameterOffset, javacard.security.ECPrivateKey regeneratedPrivateKey)** in [de.fac2 JavaDoc] for details and parameters of this interface.

1.2.2 Module: Curve Parameters

This module contains the curve parameters which will be used for the elliptic curve cryptography. It contains only one interface. It will be used from Module: FIDO-U2F-Implementation to initialize key objects. This module is static and doesn't need to be instantiated to be used.

As this module is a helper module used to set the curve parameters which are used in Module: FIDO-U2F-Implementation it is categorized as **SFR-supporting**.

Interface setCommonCurveParameters

This interface will be called from Interface: generatePrivateKey to set all curve parameters to the given ECKey object. After the ECKey object is filled with the curve parameters it is initialized and can be used for further crypto calls.

1.3 Subsystem: JavaCard platform

This subsystem contains the JavaCard Runtime Environment [JCRE304] which includes the implementation of the Java Card Virtual Machine, the Java Card API classes, and runtime support services such as the selection and deselection of applets. When the applet is installed to the platform the JRCE will call the applets Interface: install with installation parameters. After the installation and after the de.fac2 applet was selected, all incoming APDUs will be dispatched to the applet by calling the applets Interface: process.

This subsystem is the underlying platform for Subsystem: Control & Communication subsystem and Subsystem: Crypto subsystem which provides a JavaCard API which is used to call the platforms processing and security functions. Modules of this subsystem directly map to methods in the [JCAPI304]. Each method in [JCAPI304] is a module which can be called from the subsystems and modules defined in this document. The modules of this subsystem will not be listed in this chapter as they are already defined in the mentioned specification and details of how they will be called is described in the modules of Subsystem: Control & Communication subsystem and Subsystem: Crypto subsystem.

2 TSFI to Module Mapping

In the following table the TSFIs defined in the ADV FSP document are mapped to the modules defined in this document to visualize which modules are called by each TSFI.

	Module: U2FApplet	**Module: FIDO-U2F-Implementation**	**Module: Curve Parameters**
U2F_REGISTER	x	x	x
U2F_AUTHENTICATE	x	x	x
U2F_GET_VERSION	x		
SET_ATTESTATION_CERT	x		
GET_RESPONSE	x		
RESET	x	x	

Table 1: TSFI to module mapping

3 SFR to Module Mapping

The following table provides an overview for the SFR coverage by the modules defined in chapter 1. It shows that all SFR are addressed by the modules.

	Module: U2FApplet	Module: FIDO-U2F-Implementation	Module: Curve Parameters
FDP_IFC.1/ Initialisation	x	x	
FDP_IFF.1/ Initialisation	x	x	
FDP_IFC.1/ Reset	x	x	
FDP_IFF.1/ Reset	x	x	
FDP_IFC.1/ Registration	x	x	x
FDP_IFF.1/ Registration	x	x	x
FDP_IFC.1/ Authentication	x	x	x
FDP_IFF.1/ Authentication	x	x	x
FMT_SMF.1	x	x	
FMT_SMR.1	x		
FMT_MTD.1	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		

	Module: U2FApplet	Module: FIDO-U2F-Implementation	Module: Curve Parameters
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	
FCS_RNG.1/ nonce		x	
FDP_SDI.1[1]			
FCS_CKM.5/ U2F-Private		x	x
FCS_CKM.5/ U2F-Public		x	
FPR_ANO.1[2]			
FIA_UAU.2	x		
FIA_UAU.6	x		
FPT_EMS.1[3]			
FPT_PHP.3[4]			
FPT_TST.1[5]			

Table 2: SFR to module mapping

1 This SFR is implicitly satisfied by the underlying Java Card Platform
2 This SFR is implicitly satisfied by design of the FIDO U2F standard
3 This SFR is implicitly satisfied by the underlying hardware security platform
4 This SFR is implicitly satisfied by the underlying hardware security platform
5 This SFR is implicitly satisfied by the underlying Java Card Platform

Glossary

U2F Universal Second Factor

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

Reference Documentation

de.fac2 JavaDoc	Federal Office for Information Security: Source Code Documentation of de.fac2 U2F Authenticator Applet Version 1.34
JCAPI304	Oracle: Java Card API, Classic Edition, Version 3.0.4
FIDOSpec	FIDO Alliance: Fido Alliance Universal 2nd Factor 1.1 Specifications
JCRE304	Oracle: Java Card 3 Platform Runtime Environment (JRCE) Specification, Classic Edition, Version 3.0.4