

swissbit[®]

User Manual

Swissbit iShield Key

iShield Key FIDO2 [USB-A/NFC]

iShield Key Pro [USB-A/NFC]

Date: 28 February 2024

Document Version: 1.7.0

Copyright 2022 by Swissbit AG

This document as well as the information or material contained is copyrighted. Any use not explicitly permitted by copyright law requires prior consent of Swissbit AG. This applies to any reproduction, revision, translation, storage on microfilm as well as its import and processing in electronic systems, in particular.

The information or material contained in this document is property of Swissbit AG and any recipient of this document shall not disclose or divulge, directly or indirectly, this document or the information or material contained herein without the prior written consent of Swissbit AG.

All copyrights, trademarks, patents and other rights in connection herewith are expressly reserved to Swissbit AG and no license is created hereby.

Subject to technical changes.

All brand or product names mentioned are trademarks or registered trademarks of their respective holders.

Table of Contents

TABLE OF CONTENTS.....	3
1 DOCUMENT INFORMATION	5
2 OVERVIEW ISHIELD KEY	5
3 SWISSBIT MANAGEMENT TOOLS	7
3.1 ISHIELD KEY MANAGER	7
3.1.1 Installation.....	7
3.1.2 Dashboard	9
3.1.3 WebAuthn Dashboard Card	10
3.1.4 TOTP Dashboard Card.....	12
3.1.5 HOTP Dashboard Card.....	16
3.1.6 PIV Dashboard Card	20
3.2 ISHIELD KEY MANAGER COMMAND LINE TOOL	23
3.2.1 FIDO Command.....	23
3.2.2 TOTP Command.....	23
3.2.3 HOTP Command.....	25
3.2.4 PIV Command.....	26
4 FIDO2 APPLICATIONS (STANDARD).....	28
4.1 OVERVIEW	28
4.1.1 FIDO2 Registration	28
4.1.2 FIDO2 Login	29
4.2 GETTING STARTED WITH FIDO2 APPLICATIONS.....	29
4.2.1 Preconditions.....	29
4.2.2 PIN Setup of Swissbit iShield Key.....	29
4.2.3 Test Registration	31
4.2.4 Test Login.....	32
4.2.5 Register Swissbit iShield Key on an online Microsoft account	34
4.2.6 Usernameless/Passwordless Sign-in on an online Microsoft account	38
4.2.7 Sign-in with external Identity Provider	38
4.3 SWISSBIT ISHIELD KEY ON VARIOUS SERVICES	46
4.3.1 Autho.....	46
4.3.2 Bitbucket	50
4.3.3 Github	52
4.3.4 Amazon Web Service (AWS)	55
5 TOTP APPLICATIONS	57
5.1 OVERVIEW	57
5.1.1 Registration.....	57
5.1.2 TOTP Computation.....	58
5.1.3 Password Generation and Authentication	59
5.2 SWISSBIT ISHIELD KEY ON VARIOUS SERVICES	59
5.2.1 Github	59
6 HOTP APPLICATIONS	61
6.1 OVERVIEW AND FUNCTIONALITY	61
6.1.1 Registration.....	61
6.1.2 HOTP Computation.....	62
6.1.3 Password Generation and Authentication	62
6.1.4 Counter Resynchronization.....	63
7 PIV APPLICATIONS.....	64

7.1	OVERVIEW USE CASES.....	64
7.1.1	Logon.....	65
7.1.2	Bitlocker.....	66
7.1.3	Active Directory.....	66
7.2	UNDERLYING COMPONENTS.....	67
7.2.1	Token Provisioning and Usage on Windows	67
7.2.2	Authentication.....	67
7.2.3	Certificate Slots.....	67
7.3	REQUIREMENTS.....	68
7.4	GETTING STARTED WITH PIV ON ISHIELD KEY PRO	68
7.4.1	PIV Installation Package.....	68
7.4.2	Installation of the OpenSC Minidriver and iShield PIV Module	68
7.4.3	Preparation of the iShield Key Pro	71
7.4.4	Reset the iShield Key Pro	71
7.5	USE CASE: LOCAL ACCOUNT BITLOCKER	72
7.5.1	Setup Process	72
7.5.2	Use it to encrypt a Drive.....	78
7.6	USE CASE: ACTIVE DIRECTORY BITLOCKER	79
7.6.1	Setup on Server.....	79
7.6.2	Self-enroll Certificate on Client PC.....	84
7.6.3	Use it on Client.....	86
7.7	USE CASE: ACTIVE DIRECTORY PC LOGON	87
7.7.1	Setup on Server.....	87
7.7.2	Self-enroll Certificate on Client PC.....	87
7.7.3	Use it on Client.....	88
7.8	TROUBLESHOOTING	89
7.8.1	Troubleshooting "The smart card is read-only / cannot perform the requested operation".....	89
7.8.2	Troubleshooting "An internal consistency check failed"	89
8	GLOSSARY.....	90
9	DOCUMENT HISTORY	91

1 Document Information

This document describes how to get started with your Swissbit iShield Key FIDO2 [USB-A/NFC] (hereinafter referred to as "iShield Key FIDO2"), and iShield Key Pro [USB-A/NFC] (hereinafter referred to as "iShield Key Pro") security key. The iShield Key FIDO2 and iShield Key Pro offer strong authentication that is simple, secure and flexible. They also protect users against online attacks such as phishing, social engineering and account takeover.

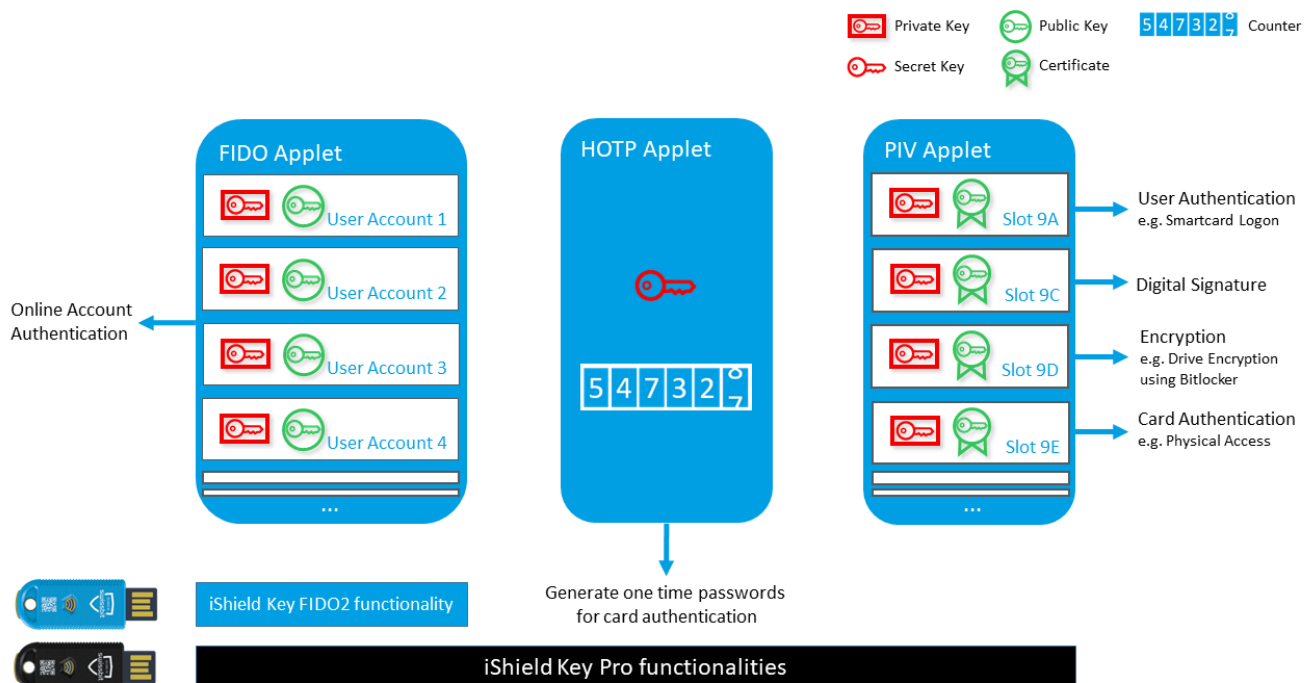
The iShield Key FIDO2 supports FIDO2 and U2F standards to protect online accounts. The iShield Key Pro implement a one-chip solution with multiple applets installed to support various use cases. The iShield Key Pro supports, additionally to FIDO2 applications, generation of time-based one-time passwords (TOTP) and HMAC-based one-time passwords (HOTP) and personal identification and verification (PIV). You can find an overview of supported use cases and references to more detailed descriptions later in this document in section 2. Section 3 introduces the management tools accompanying the security key. Eventually, the uses cases are explained by applet in section 4, section 5, section 6 and section 7. Section 4 gives in-depth guidance on how to get started with the FIDO2 functionality, section 5 and section 6 explain the TOTP and HOTP generation and usage and section 7 presents how to provision and use your iShield Key Pro key as a PIV device.

In section 4, the iShield Key FIDO2 is utilized for all FIDO2 use cases since it shares the same functionality as the iShield Key Pro. Sections 6 and 7 are specifically applicable to the iShield Key Pro and do not pertain to the iShield Key FIDO2. Section 5 only applies to iShield Keys with the TOTP applet installed.

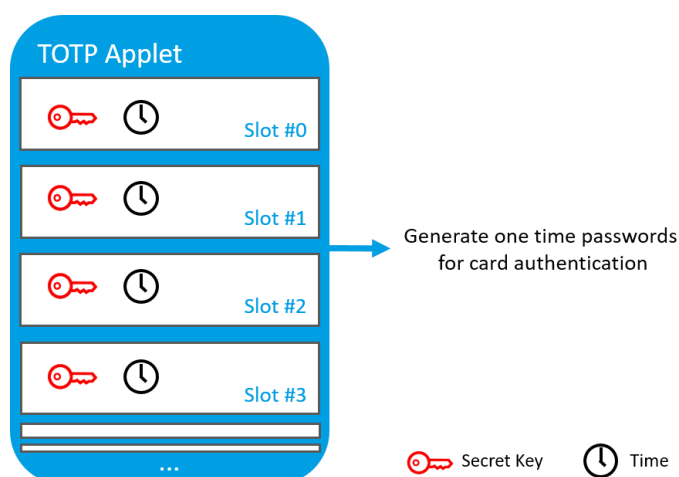
In the following, "iShield Key" will be used in scenarios where both iShield Key FIDO2 and iShield Key Pro are compatible.

2 Overview iShield Key

The Swissbit iShield Key FIDO2 security key has the FIDO2 applet installed only, and the Swissbit iShield Key Pro security key has the FIDO2, HOTP and PIV applets installed.



The iShield Key product family has also product variants with the TOTP applet installed additionally.



In this section, you can find the functionalities that support your use cases.

The following table guides you to the correct section in this guide for a more detailed description of your use case:

Use Case	Description	Applet	iShield Key FIDO2	iShield Key Pro	Reference
Online Authentication	User authentication for FIDO2 and U2F compatible websites and services	FIDO2	✓	✓	section 4.2
2FA for Online Accounts / VPN	Two-factor authentication to websites, services or VPN supporting TOTP	TOTP	(✓)*	(✓)*	section 5.1
2FA for Online Accounts / VPN	Two-factor authentication to websites, services or VPN supporting HOTP	HOTP		✓	section 6.1
Bitlocker – Local Account	Drive encryption using Bitlocker for Local Windows Accounts	PIV		✓	section 7.5
Bitlocker – Active Directory	Drive encryption using Bitlocker for Windows Active Directory Domain Accounts	PIV		✓	section 7.6
Windows Account Logon – Active Directory	Logon into Windows Active Directory Domain Account	PIV		✓	section 7.7

(✓)*: Not all iShield Key FIDO2 and Pro come with the TOTP applet installed. Please check the specification of your iShield Key.

3 Swissbit Management Tools

The iShield Key Manager and iShield Key Manager command line tool support all required operations to manage the applets on your iShield Key and assist the use cases presented in this guide. You can download both tools from the [Swissbit iShield Key landing page](#). When using the iShield Key, we assume that you agree with the license terms. The license can be found in the installation directory of the application.

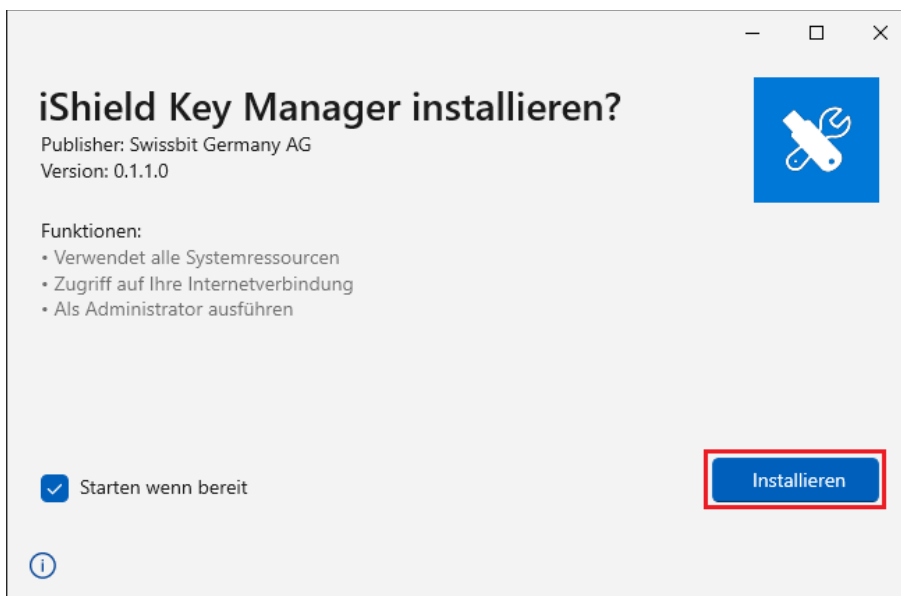
Section 3.1 explains how to use the iShield Key Manager, section 3.2 lists the commands supported by the iShield Key Manager command line tool.

3.1 iShield Key Manager

3.1.1 Installation

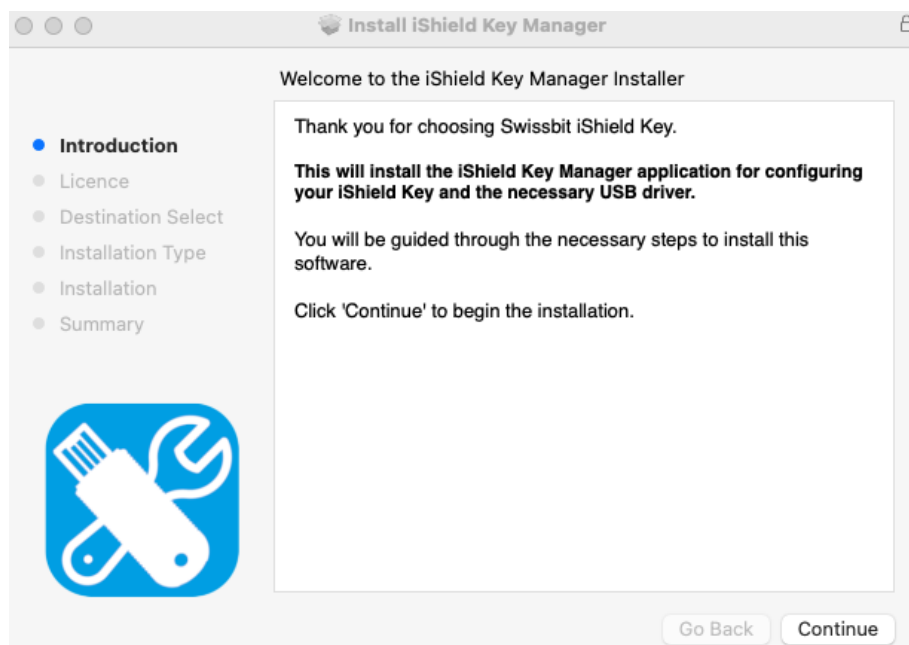
Windows Installation

To install the application, it is necessary to double click the ishield_key_manager.msix package and follow the installer steps.



macOS Installation

To install the application, double click the iShieldKeyManagerInstaller.pkg package. The installer will guide you through the necessary steps to install the application.



Click Continue to start the installation. Then please accept the license and follow the on-screen instructions. After installation is finished, you can start the application by going to “Applications” and clicking “iShield Key Manager”.

Linux Installation

You can install the iShield Key Manager on Linux by

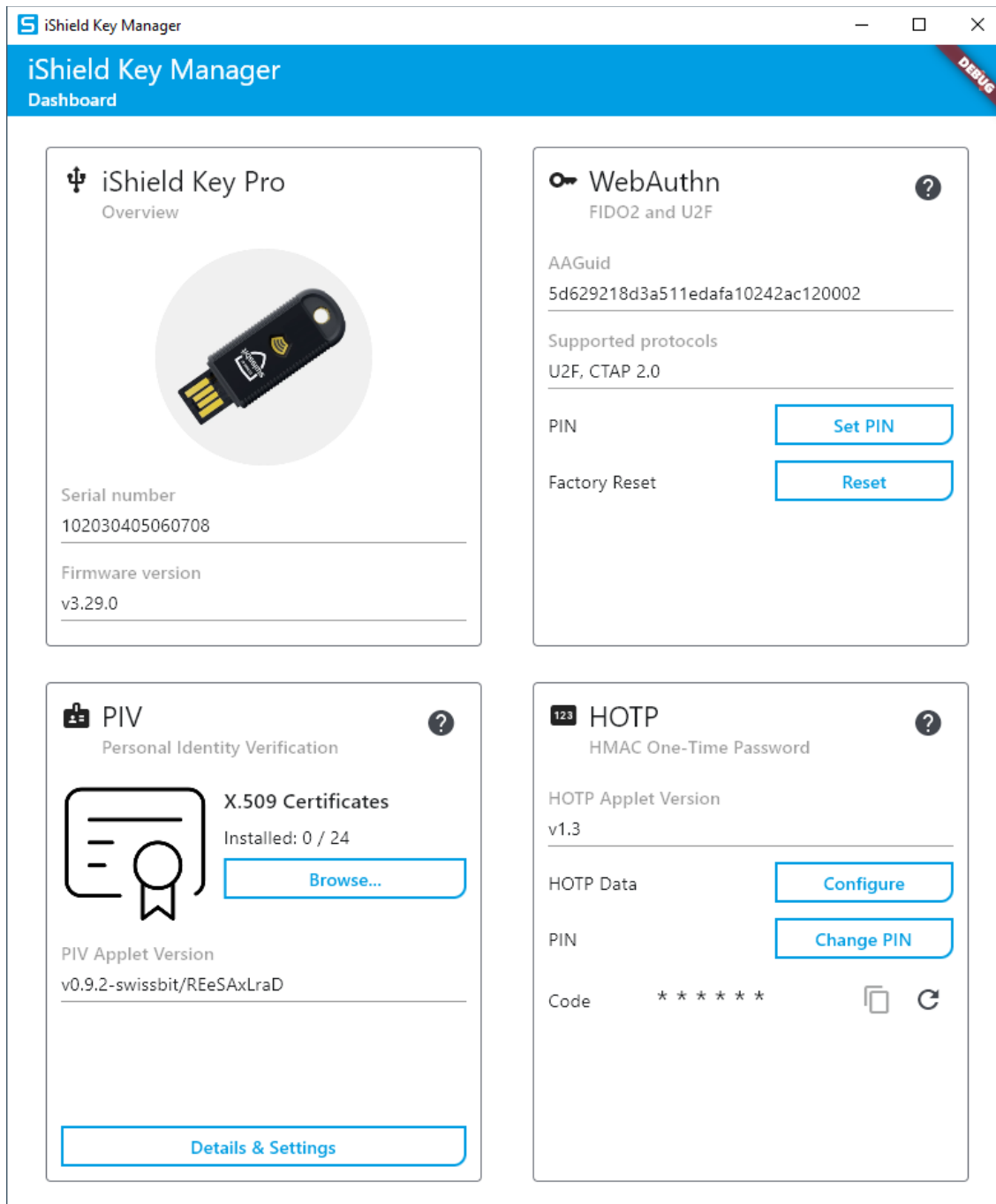
```
sudo dpkg -i ishield-key-manager.deb
```

To start the application run the following command:

```
ishield-key-manager
```


3.1.2 Dashboard

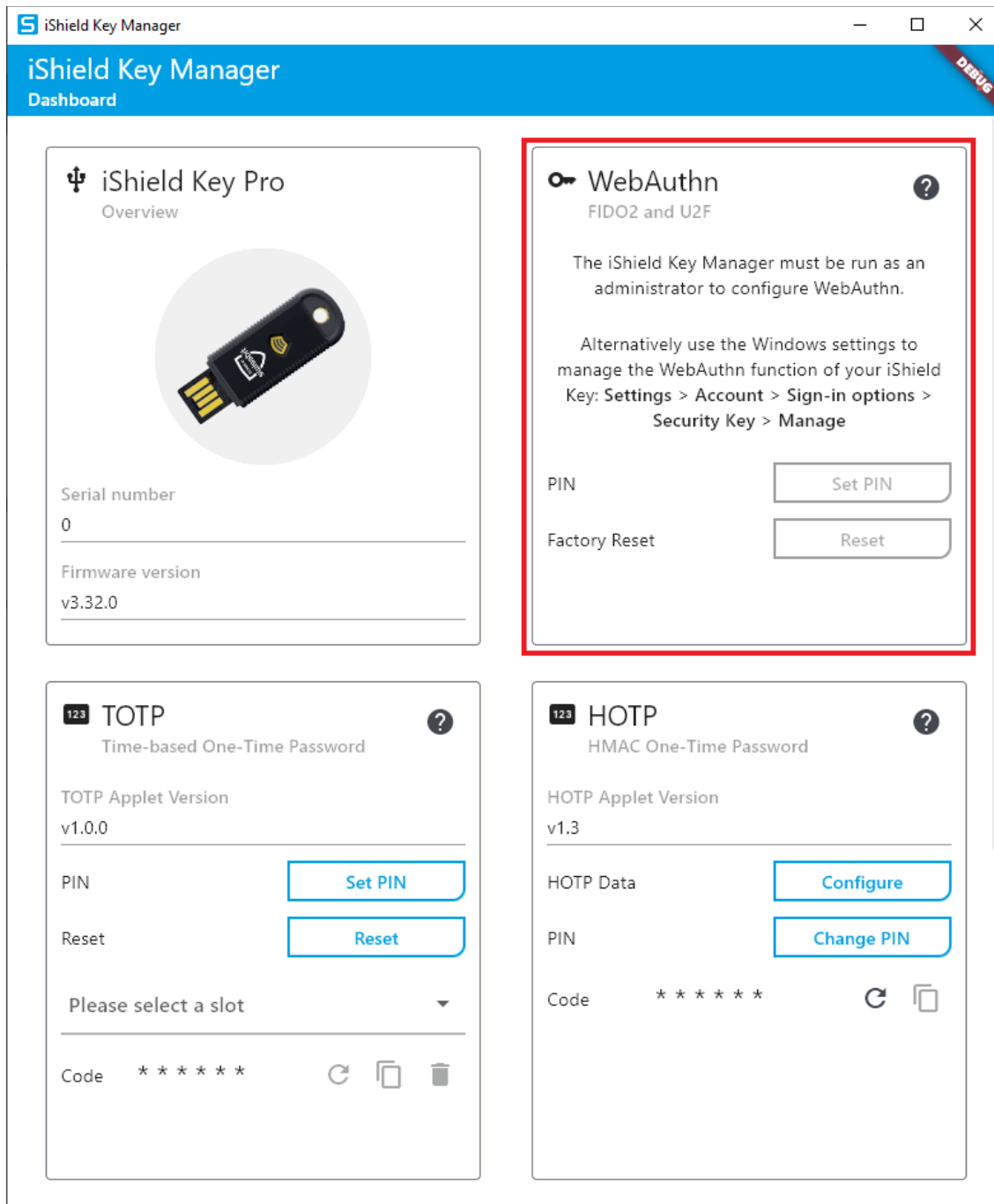
The iShield Key Manager only supports a single iShield Key connected to the host PC. After starting the app, the data of your iShield Key is loaded and a dashboard with an overview and further cards for managing the applets on your iShield Key is shown:



The overview dashboard cards provide general information like the serial number and firmware version of your iShield Key.

3.1.3 WebAuthn Dashboard Card

To manage the WebAuthn function of your iShield Key on Windows, you need to start the iShield Key Manager as administrator. If you start the tool without administrator rights, you are referred to the Windows settings to set a PIN or perform a reset.



On macOS or when starting the app as administrator on Windows, the WebAuthn dashboard card shows the AAGuid and supported protocols and allows managing the FIDO2 applet on your iShield Key. You can set a new PIN or later change it. When you forgot your PIN, you can reset the WebAuthn function of your iShield Key to factory settings. This will delete all WebAuthn data and credentials.

iShield Key Manager
- □ ×

iShield Key Manager
DEBUG

iShield Key Pro
?

Overview

Serial number
102030405060708

Firmware version
v3.29.0

WebAuthn
?

FIDO2 and U2F

AAGuid
5d629218d3a511edafa10242ac120002

Supported protocols
U2F, CTAP 2.0

PIN
Set PIN

Factory Reset
Reset

PIV
?

Personal Identity Verification

X.509 Certificates

Installed: 0 / 24

Browse...

PIV Applet Version
v0.9.2-swissbit/REeSAXLraD

Details & Settings

123 HOTP
?

HMAC One-Time Password

HOTP Applet Version
v1.3

HOTP Data
Configure

PIN
Change PIN

Code
* * * * *

📄
↻

Page 11 of 92

3.1.4 TOTP Dashboard Card

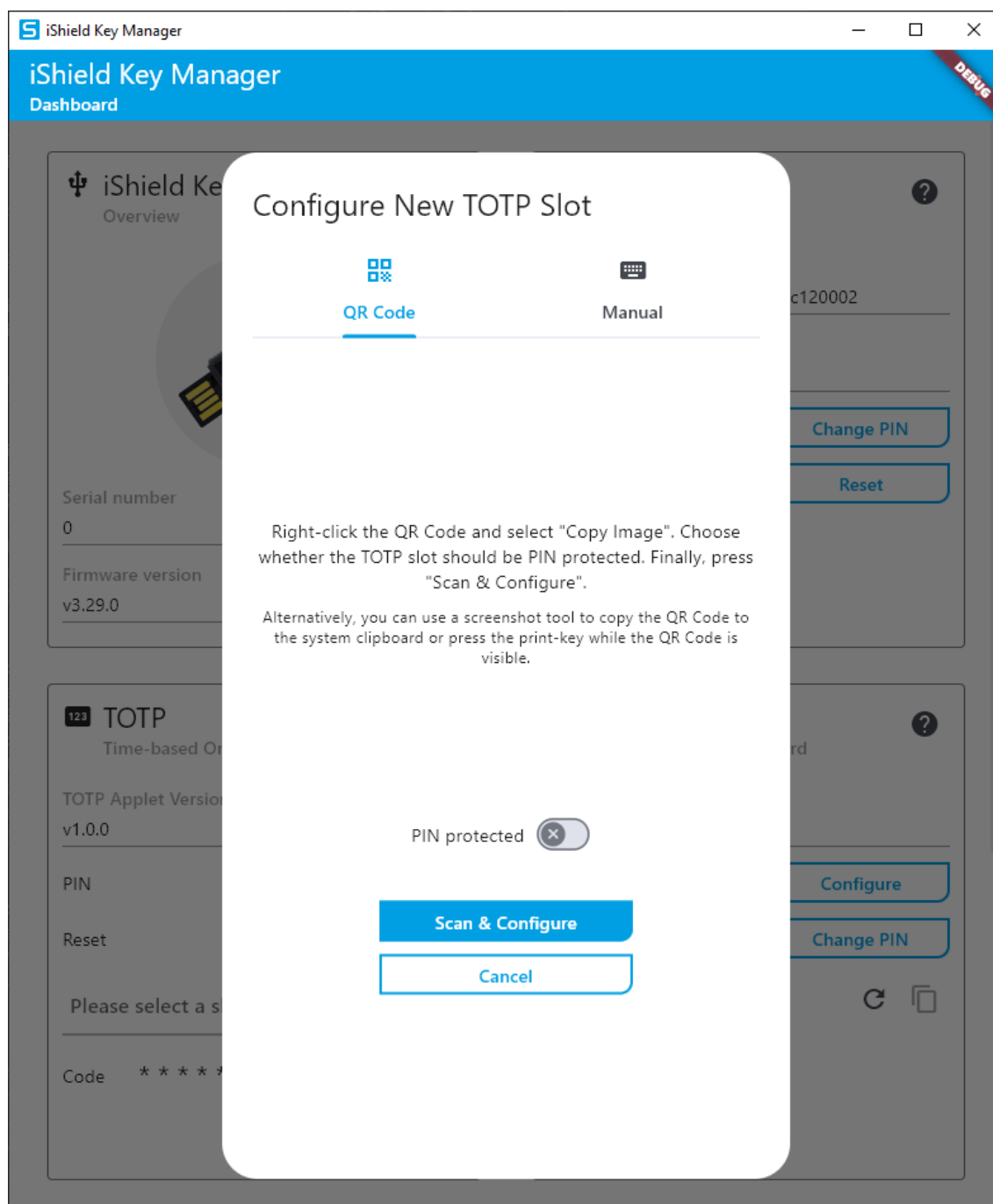
If your iShield Key has the TOTP applet installed, you will find a dashboard card to manage the applet.

The screenshot displays the iShield Key Manager Dashboard with four main cards:

- iShield Key Pro Overview:** Shows the device image, serial number (102030405060708), and firmware version (v3.29.0).
- WebAuthn (FIDO2 and U2F):** Displays the AAGuid (5d629218d3a511edafa10242ac120002), supported protocols (U2F, CTAP 2.0), and buttons for 'Set PIN' and 'Reset'.
- TOTP (Time-based One-Time Password):** This card is highlighted with a red border. It shows the applet version (v1.0.0), buttons for 'Set PIN' and 'Reset', a dropdown menu labeled 'Please select a slot', and a code field with asterisks and refresh/copy/delete icons.
- HOTP (HMAC One-Time Password):** Shows the applet version (v1.3), buttons for 'Configure' and 'Change PIN', and a code field with asterisks and refresh/copy icons.

Before you can use the TOTP function of your iShield Key as a second factor, you need to pair your iShield Key with a service and configure a new TOTP slot. In the dropdown, you can either select an already configured slot or configure a new one.

You can configure a new slot with the account details, system configuration and secret key given in the form of a QR code during user registration:



Optionally, you can protect the one-time password generation for your slot with your PIN. If you choose this option, you need to provide your PIN in order to generate a one-time code. You must first set a PIN, as described below, before you can use the PIN protection function.

Alternatively, you can manually configure a new slot. Define an issuer and an account name for your new TOTP slot and enter the configuration values, i.e. key, algorithm, digits and interval, provided by your service:

iShield Key Manager Dashboard

Configure New TOTP Slot

QR Code Manual

Account

Issuer

SHA1

Time Interval / Period in Seconds (Default: 30)

Base32 Key

TOTP Code Length: 6

Protect OTP Generation with PIN

Cancel Configure TOTP

To finish the setup, your service asks you to enter a one-time password. Select your newly configured slot to generate a new one-time code and submit it to your service as long as it is valid. A countdown progress bar on the dashboard displays the remaining valid time.

The screenshot displays the iShield Key Manager Dashboard with the following sections:

- iShield Key Pro Overview:** Shows a USB key icon, serial number 102030405060708, and firmware version v3.32.0.
- WebAuthn (FIDO2 and U2F):** Displays AAGuid 5d629218d3a511edafa10242ac120002, supported protocols (U2F, CTAP 2.0), and buttons for 'Set PIN' and 'Reset'.
- TOTP (Time-based One-Time Password):** Shows applet version v1.0.0, buttons for 'Change PIN' and 'Reset', a dropdown menu with 'Swissbit Developer Portal:john.doe...', and a code display showing '192211' with refresh, copy, and delete icons.
- HOTP (HMAC One-Time Password):** Shows applet version v1.3, buttons for 'Configure' and 'Change PIN', and a code display showing '*****' with refresh and copy icons.

You can set a new PIN if you want to configure PIN-protected slots or change your PIN. The PIN must be between four and eight characters in length.

If you enter your PIN incorrectly 10 times, your PIN will be blocked irreversibly! Once your PIN is blocked, you will not be able to generate one-time passwords for PIN-protected slots anymore and you have to reset your PIN. A TOTP PIN reset deletes all credentials in PIN-protected slots and removes the PIN. Successful authentication of the PIN resets the retry counter. A complete factory reset restores the TOTP function of your iShield Key to factory settings. All TOTP data and credentials are deleted. You can also clear a single selected slot by clicking on the trash bin icon.

3.1.5 HOTP Dashboard Card

If the HOTP applet is installed on your iShield Key, the dashboard will show a card for HOTP.

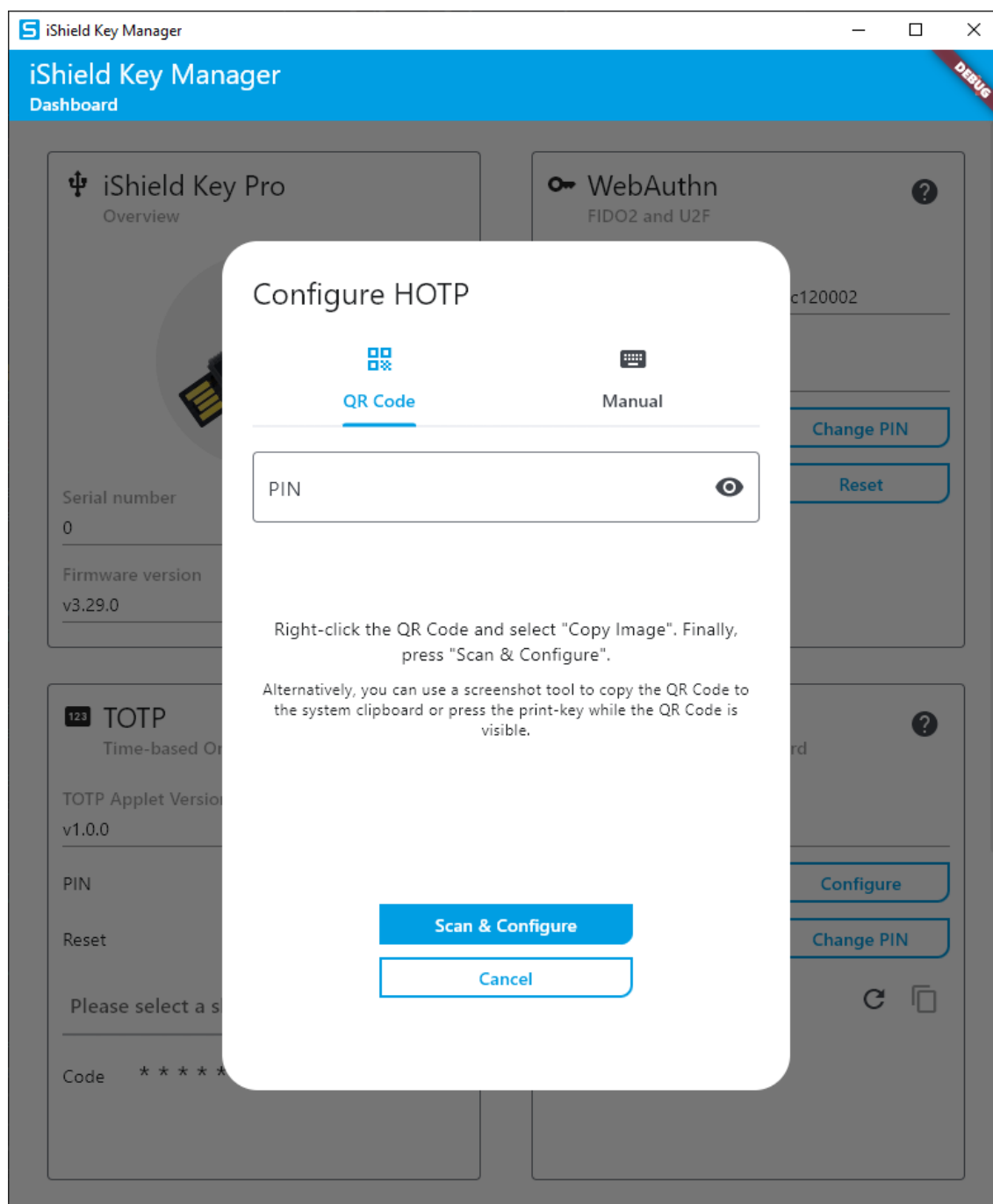
The screenshot shows the iShield Key Manager Dashboard with four main cards. The HOTP card is highlighted with a red border. The cards are:

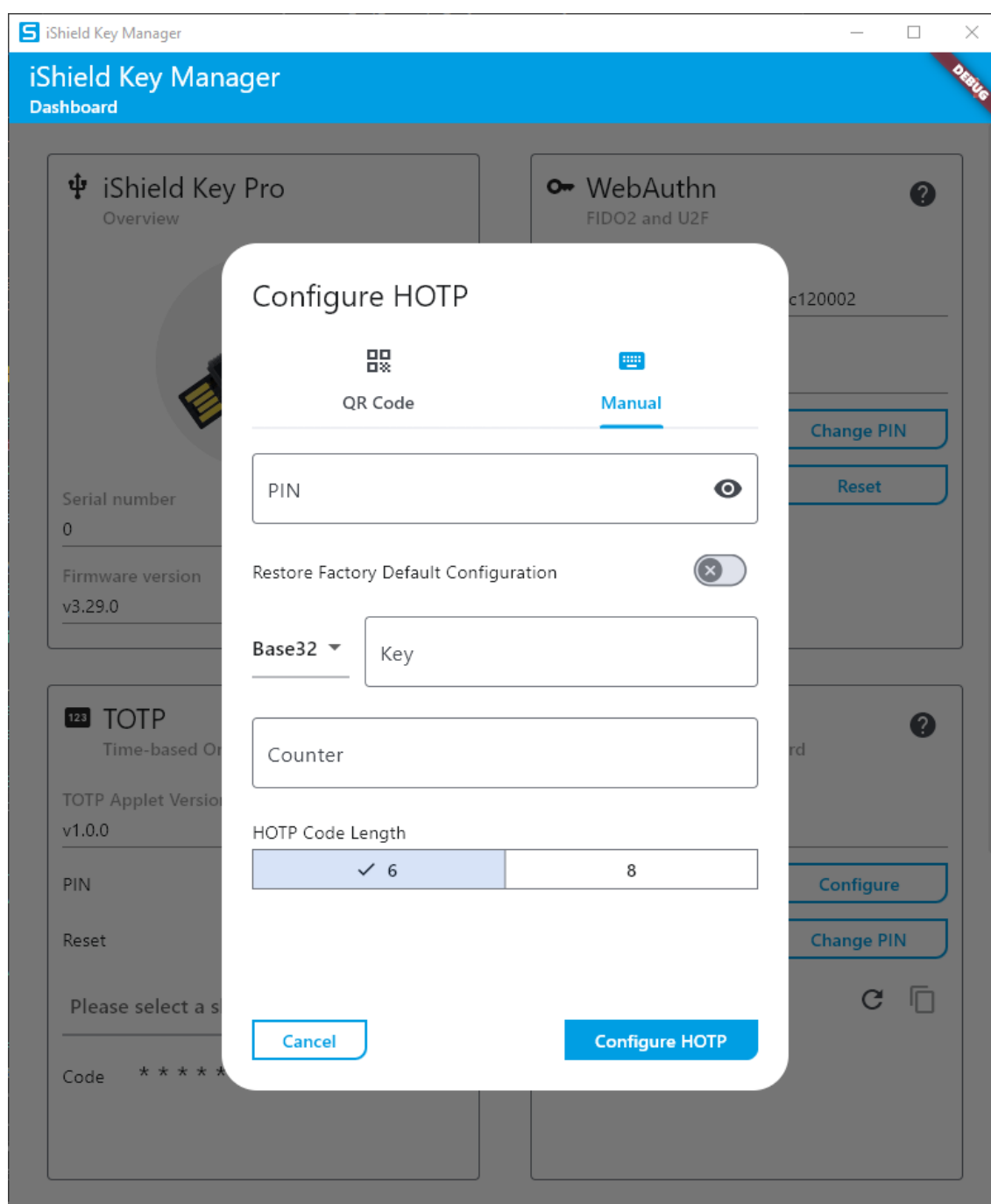
- iShield Key Pro Overview:** Shows a USB key icon, serial number 102030405060708, and firmware version v3.29.0.
- WebAuthn (FIDO2 and U2F):** Shows AAGuid 5d629218d3a511edafa10242ac120002, supported protocols U2F, CTAP 2.0, and buttons for Set PIN and Reset.
- PIV (Personal Identity Verification):** Shows X.509 Certificates (Installed: 0 / 24) with a Browse... button, PIV Applet Version v0.9.2-swissbit/REeSAXLraD, and a Details & Settings button.
- HOTP (HMAC One-Time Password):** Shows HOTP Applet Version v1.3, HOTP Data with a Configure button, PIN with a Change PIN button, and a Code field with six asterisks and copy/paste icons.

Before you pair your iShield Key to use it as second factor for a service, i.e. configure the HOTP data, we strongly recommend to change your PIN. The factory default PIN is 1234. The PIN must be between four and eight characters in length.

If you enter your PIN incorrectly 10 times, your PIN will be blocked irreversibly! You can still generate one-time passwords but you will no longer be able to set a new secret key and counter and register your iShield Key Pro for another service. Successful authentication of the PIN resets the retry counter.

After changing your PIN, you can configure your HOTP applet with a secret key, counter and OTP length given in form of a QR code or you enter the information manually:





The secret key in hex format must be of a length between 16 and 64 bytes and the counter must be a positive number. The iShield Key supports generation of one-time passwords of length 6 or 8. You can also restore the factory default configuration. The factory default key is 3132333435363738393031323334353637383930 and the factory default initial counter value is 0.

You can generate a new HOTP code by touching the end of your iShield Key or you can click the associated icon in the HOTP dashboard card:

The screenshot displays the iShield Key Manager Dashboard with the following sections:

- iShield Key Pro Overview:** Shows a USB key icon, serial number 102030405060708, and firmware version v3.29.0.
- WebAuthn (FIDO2 and U2F):** Displays AAGuid 5d629218d3a511edafa10242ac120002, supported protocols (U2F, CTAP 2.0), and buttons for 'Change PIN' and 'Reset'.
- PIV (Personal Identity Verification):** Shows X.509 Certificates (0/24 installed) with a 'Browse...' button and PIV Applet Version v0.9.2-swissbit/REeSAxLraD. A 'Details & Settings' button is at the bottom.
- HOTP (HMAC One-Time Password):** Shows HOTP Applet Version v1.3, HOTP Data with a 'Configure' button, and a PIN field with a 'Change PIN' button. The current code is 68254676, with copy and refresh icons.

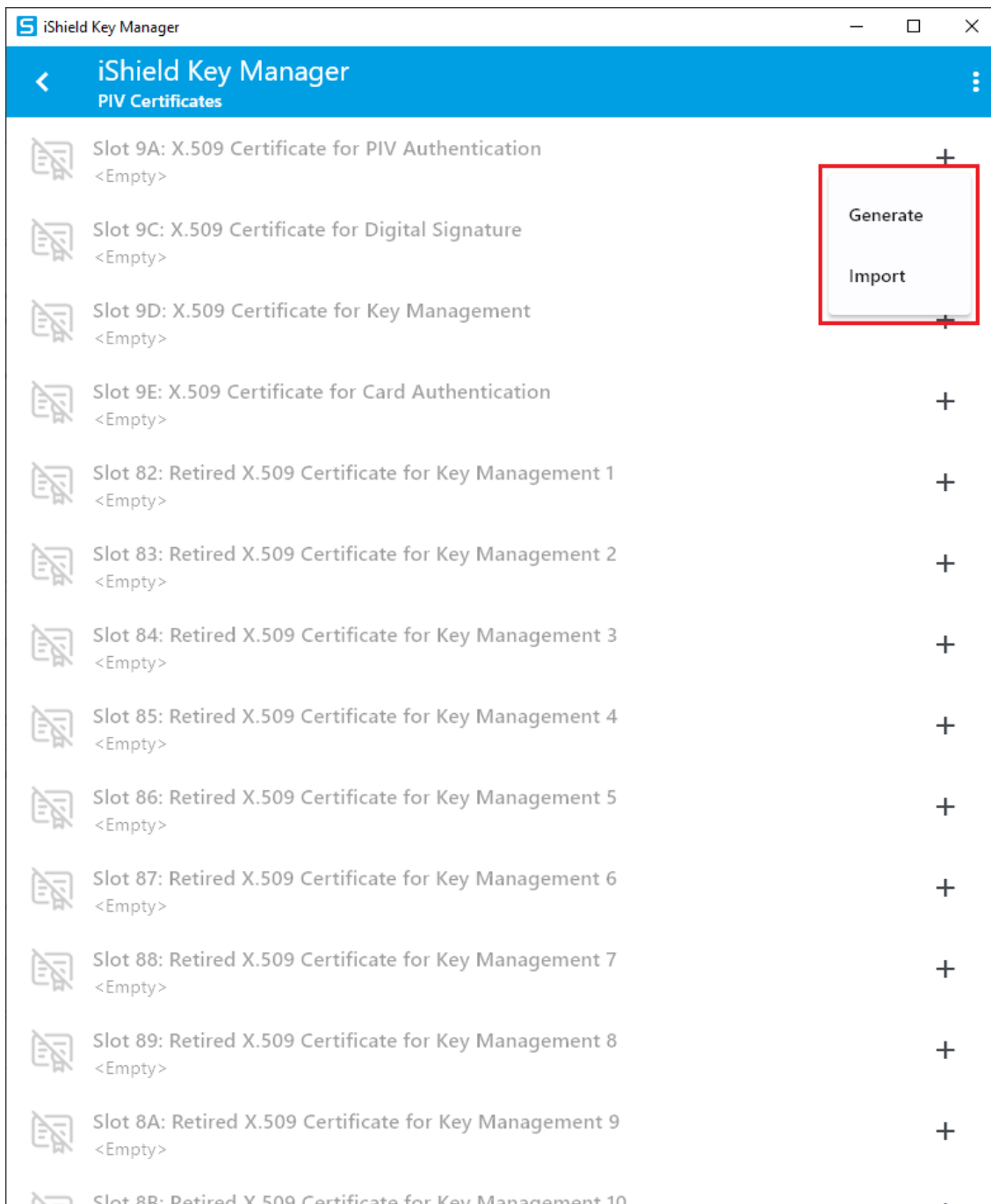
3.1.6 PIV Dashboard Card

Your iShield Key Pro comes with the PIV applet installed. You will also find a card to manage the PIV applet on the dashboard. The card shows the version of the installed PIV applet and the number of installed certificates.

The screenshot displays the iShield Key Manager Dashboard with four main cards:

- iShield Key Pro Overview:** Shows a USB key icon, serial number 102030405060708, and firmware version v3.29.0.
- WebAuthn (FIDO2 and U2F):** Shows AAGuid 5d629218d3a511edafa10242ac120002, supported protocols U2F, CTAP 2.0, and buttons for Set PIN and Reset.
- PIV (Personal Identity Verification):** Highlighted with a red border. Shows X.509 Certificates (Installed: 0 / 24) with a Browse... button, PIV Applet Version v0.9.2-swissbit/REeSAXLraD, and a Details & Settings button.
- HOTP (HMAC One-Time Password):** Shows HOTP Applet Version v1.3, HOTP Data with a Configure button, PIN with a Change PIN button, and a Code field with asterisks and copy/paste icons.

It is highly recommended to change the PIN, PUK and management key before using the iShield Key Pro. For this you can click on *Details & Settings*.



In this view, you can also inspect the details of installed certificates, export or delete them.

3.2 iShield Key Manager Command Line Tool

We recommend adding the iShield Key Manager Command Line Tool (iKMcli.exe) to your path. Then you can execute operations as follows:

```
iKMcli <command> <options>
```

The help of the iKMcli lists all commands and options. You can print the help by `iKMcli --help` or `iKMcli <command> --help` for a specific command.

You can print the info for your iShield Key, including the serial number, by the command

```
iKMcli info --reader <reader>
```

To list all available smartcard readers and FIDO2 devices, use the following command:

```
iKMcli list
```

The detection of FIDO2 devices requires the execution of the command line tool as administrator.

There is one command for addressing the single applets installed in the iShield Key. In the following three sections, the supported operations for FIDO2, HOTP and PIV (iShield Key Pro only) are explained in more detail.

3.2.1 FIDO Command

The command `fidu` for managing the FIDO2 applet provides an option to print information about your connected FIDO2 device:

```
iKMcli fidu --info
```

If no PIN is set yet, you can set a new PIN by

```
iKMcli fidu --set-pin <new pin>
```

You can change your PIN by

```
iKMcli fidu --change-pin <new pin> --pin <pin>
```

To erase all credentials and the PIN, you can reset the FIDO2 device by

```
iKMcli fidu --reset
```

You will be asked to touch the FIDO2 device to reset. The touch point is located at the end of the device.

All these options for the `fidu` command use the first detected iShield Key but you can also specify a FIDO2 device. In order to execute an operation for a specific device, pass its path with the option `--fido-path <path>`. You can use the `list` command to list the paths of all connected FIDO2 devices.

The `fidu` command requires administrator rights.

3.2.2 TOTP Command

The command `totp` has an info option to print the version of the applet and the number of available TOTP slots:

```
iKMcli totp --info
```

You can configure a new TOTP slot by

```
ikmcli totp --conf-slot <slot index>
--key <key> [--key-format <base32|hex>]
--account <account> [--issuer <issuer>]
[--hmac <SHA1|SHA256|SHA512>]
[--otp-length <6|7|8|9>]
[--period <time period / interval>]
[--pin-protected]
```

You need to provide a slot index between 0 and 41 and a key and set an account name. Optionally, you can specify an HMAC hash algorithm, the length of the generated one-time passwords, the validity period of generated passcodes and if the TOTP generation for the slot shall be PIN-protected. The default HMAC algorithm is SHA1, the TOTP length is six and passcodes are valid for 30 seconds. By default, TOTP generation is not PIN-protected. You can also define an issuer name. The account and issuer name must not exceed 60 characters in length.

Already configured slots are reconfigured with the new values. You can list all configured TOTP slots and get the number of maximum available slots by

```
ikmcli totp --list-slots
```

To generate a one-time password for a specific slot and current host time use the following command:

```
ikmcli totp --gen-totp <slot index>
```

The index of a slot can be obtained by the `--list-slots` option.

You can set a new PIN or change your PIN by

```
ikmcli totp --set-pin <new pin> [--pin <pin>]
```

The PIN must be between four and eight characters in length. You can optionally pass the PIN format with the option `--pin-format <ascii|hex>`. If format option is not given, ASCII format is expected.

If you enter your PIN incorrectly 10 times, your PIN will be blocked irreversibly! Once your PIN is blocked, you can no longer generate one-time passwords for PIN-protected slots and have to reset your PIN. A PIN reset also deletes all credentials in PIN-protected slots. Successful authentication of the PIN resets the retry counter. You can reset your PIN by

```
ikmcli totp --reset-pin
```

You can check if a PIN is set and the number of maximum and remaining retries by printing the PIN info with the following command:

```
ikmcli totp --pin-info
```

All TOTP data and credentials can be deleted and default settings can be restored by a factory reset.

```
ikmcli totp --reset
```

You can clear the data in a single TOTP slot by

```
ikmcli totp --clear-slot <slot index>
```

The `totp` operations use the first detected iShield Key or you specify the smartcard reader to be used by the option `--reader <reader>`. You can use the `list` command to show the connected smartcard readers.

3.2.3 HOTP Command

The command `hotp` has an option to show information about the HOTP applet on your iShield Key Pro:

```
ikMcli hotp --info
```

The info contains the version of the applet and the serial number of your iShield Key Pro.

The default PIN for authenticating an HOTP operation is 1234. You can change this factory default PIN by

```
ikMcli hotp --change-pin <new pin> --pin <pin>
```

The PIN must be between four and eight characters in length. You can optionally pass a format for the PIN with the option `--pin-format <ascii|hex>`. If no PIN format is specified, ASCII format is assumed.

The command also offers the options to set the secret key and counter for the HOTP computation by the following commands:

```
ikMcli hotp --set-key <key> --pin <pin>
ikMcli hotp --set-counter <counter> --pin <pin>
```

The secret key must be hex encoded and be of a length between 16 and 64 bytes and the counter must be a positive number. The factory value of the key programmed during device manufacturing is 3132333435363738393031323334353637383930 and the default of the initial counter value is 0. You can restore these factory default values by

```
ikMcli hotp --restore-factory-key <key> --pin <pin>
ikMcli hotp --restore-factory-counter <counter> --pin <pin>
```

If you enter your PIN incorrectly 10 times, your PIN will be blocked irreversibly! You can still generate one-time passwords but you will no longer be able to set a new secret key and counter and register your iShield Key Pro for another application. Successful authentication of the PIN resets the retry counter.

The iShield Key Pro supports generation of one-time passwords of length 6 or 8 whereby 6 is the default length. You can adjust the HOTP length by

```
ikMcli hotp --set-otp-length <length>
```

Like the `totp` operations, the `hotp` operations use the first detected iShield Key. Alternatively, you can specify a smartcard reader by the option `--reader <reader>`. You can use the `list` command to print the connected smartcard readers.

3.2.4 PIV Command

The command `piv` also provides an option to print the version of the PIV applet installed on your iShield Key Pro and your key's serial number:

```
iKMcli piv --info
```

Using the iShield Key Manager you can change the PIN, PUK and management key that are used to authenticate PIV operations. The factory default for the PIN is 123456, the default PUK is 12345678 and the management key is 010203040506070801020304050607080102030405060708. You can change your PIN or unblock it by the PUK using the following commands:

```
iKMcli piv --change-pin <new pin> --pin <pin>
iKMcli piv --unblock-pin <new pin> --puk <puk>
```

You can change the PUK using

```
iKMcli piv --change-puk <new puk> --puk <puk>
```

The PIN and PUK retries can be configured by

```
iKMcli piv --configure-retries --pin-retries <new pin retries> --puk-retries <new puk
retries> --management-key <key> --pin <pin>
```

This command will reset the PIN and PUK to the factory default passwords.

A new management key can be set by

```
iKMcli piv --set-management-key <new key> --management-key <key>
```

You can list all certificates on the smartcard or print the certificate in a slot with the following commands:

```
iKMcli piv --list-certificates
iKMcli piv --read-certificate <slot> [--output <output file>]
```

You can generate a new key pair in a slot by

```
iKMcli piv --generate-key-pair <slot> --management-key <key> [--output <output file>]
```

The public key is printed to an output file or the command line. Then, you can use the public key to create a certificate signing request:

```
iKMcli piv --request-certificate <slot> --pin <pin> --input <input file> [--output
<output file>]
```

You need to pass the public key in an input file and make sure to request the certificate for the slot with the corresponding private key. The PIN is required to authenticate the signing of the certificate signing request with the private key.

Then, you can send the generated certificate signing request to your CA for signing and import the signed certificate with the following command:

```
iKMcli piv --import-certificate <slot> --management-key <key> --input <input file>
```

You can also import private keys by

```
iKMcli piv --import-key <slot> --management-key <key> --input <input file>
```

Always make sure to pass the correct slot, so the private key in a slot belongs to the public key and certificate.

In order to delete a certificate by its slot number use

```
iKMcli piv --delete-certificates <slot> --management-key <key>
```

If both PIN and PUK are blocked, you can reset your smartcard. This erases all PIV data and restores the default settings.

```
iKMcli piv --reset
```

New values for the card holder unique identifier (CHUID) and card capability container (CCC) can be set with the following commands:

```
iKMcli piv --set-chuid --management-key <key>  
iKMcli piv --set-ccc --management-key <key>
```

Execute the `piv` operations for the first detected iShield Key or specify a smartcard reader by the option `--reader <reader>`.

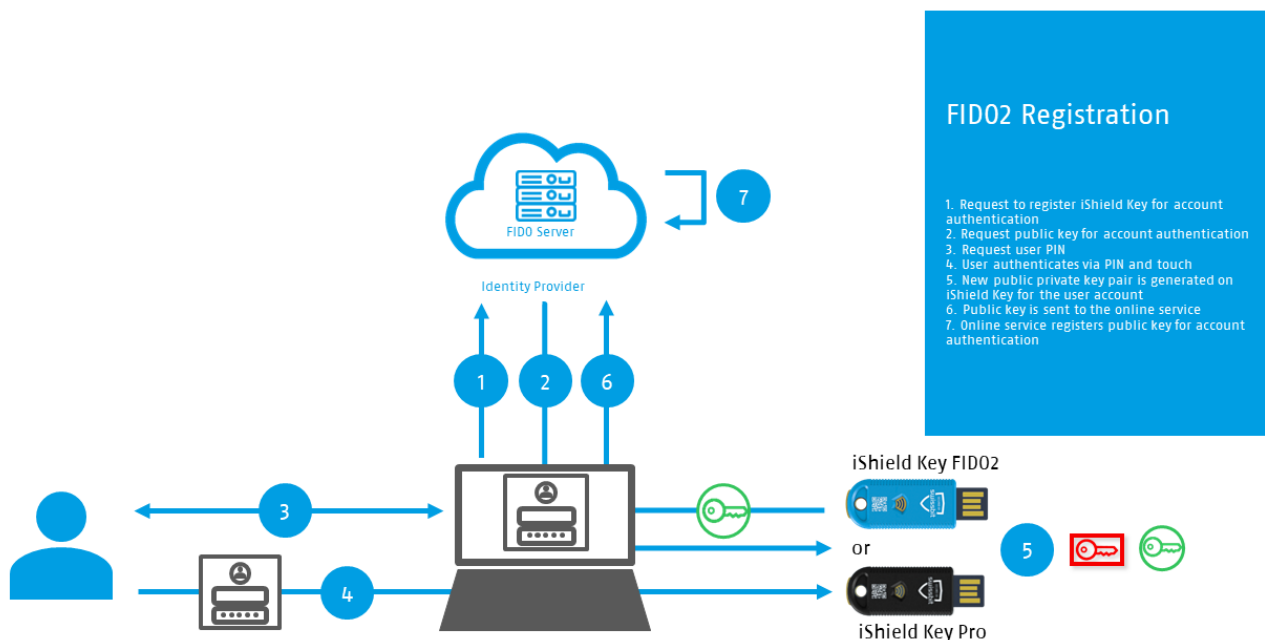
4 FIDO2 Applications (Standard)

4.1 Overview

The Swissbit iShield Key FIDO2 and iShield Key Pro are FIDO-certified plug-and-play security products that support FIDO2 and U2F standards to protect online accounts. They provide strongest and most trusted hardware authentication and allow users to securely access websites, applications, online services and company networks such as Google, Microsoft, Salesforce, Amazon Web Services, etc. You can visit FIDO Alliance (<https://fidoalliance.org/fido2/>) for more information. Swissbit provides a test website (<https://fido.ishield.cloud/>) to allow users to test with Swissbit iShield Key.

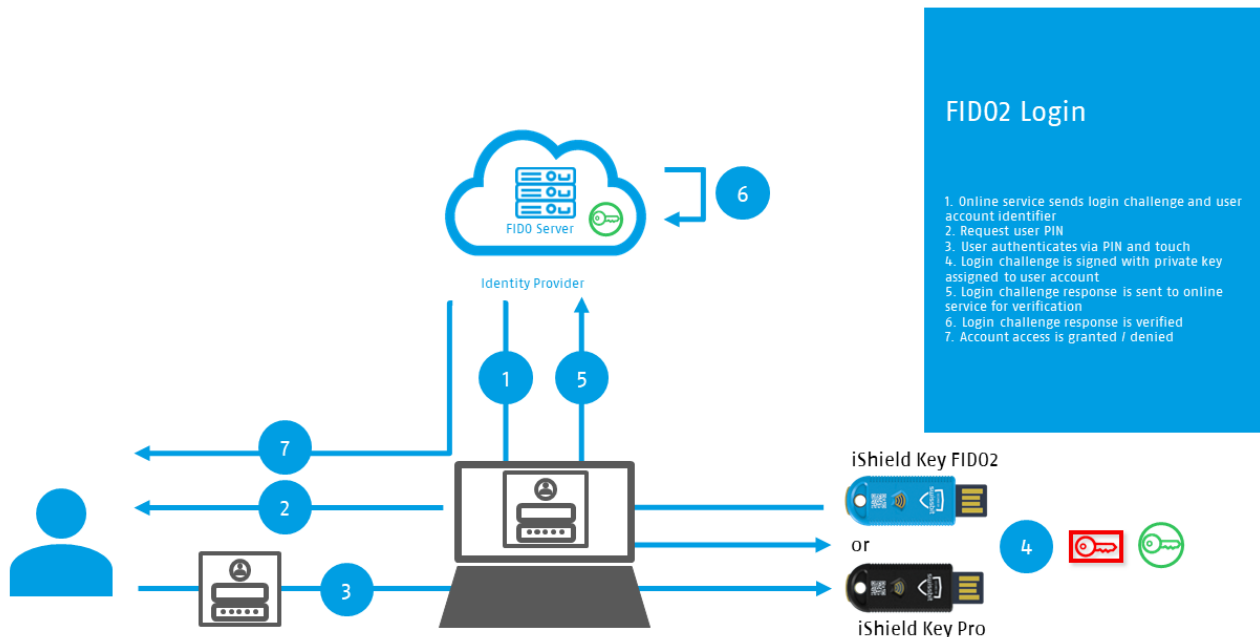
This section explains FIDO2 registration and login and all FIDO2 functionalities are compatible for both iShield Key FIDO2 and iShield Key Pro. Section 4.2 gets you started with your iShield Key and section 4.3 shows how to register the key with various online services.

4.1.1 FIDO2 Registration



When registering for an online service, the server requests a public key that it can assign with the user's account for this one service for online authentication. The user will then be able to authenticate if they are in possession of the corresponding private key. Using the iShield Key for FIDO2 registration, the user needs to authenticate with the user PIN and touch the security key. The public private key is generated on the iShield Key hardware authenticator and assigned with the user account. Lastly, the public key is sent to the server.

4.1.2 FIDO2 Login



After successful FIDO2 registration, the online service has the public key for the user account and the corresponding private key is stored securely on the iShield Key. The online service challenges the user to sign with the private key. If the online server can verify the signature using the public key, the authentication is successful and the user is granted access to their account.

4.2 Getting started with FIDO2 Applications

4.2.1 Preconditions

The Swissbit iShield Key supports platforms and applications that are conform to FIDO/U2F/WebAuthn standards. Following Platforms are supported:

- OS:
 - Windows 10,
 - MacOS,
 - Linux,
 - Chrome OS,
 - Android
- Browsers:
 - Firefox,
 - MS Edge,
 - Chrome,
 - Apple Safari

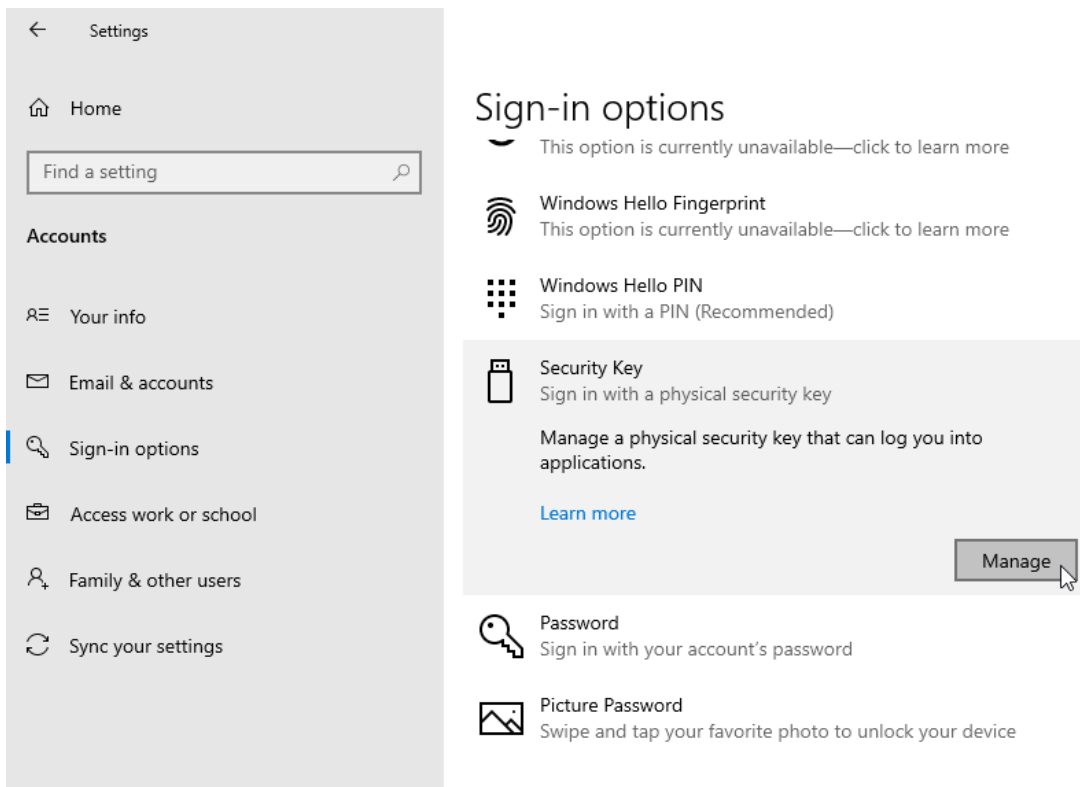
4.2.2 PIN Setup of Swissbit iShield Key

Note: The Swissbit iShield Key is ready to use. If PIN is not required, jump to section 4.2.3

To manage the security PIN of the Swissbit iShield Key, a built-in functionality for security key management of Windows 10 might be used.

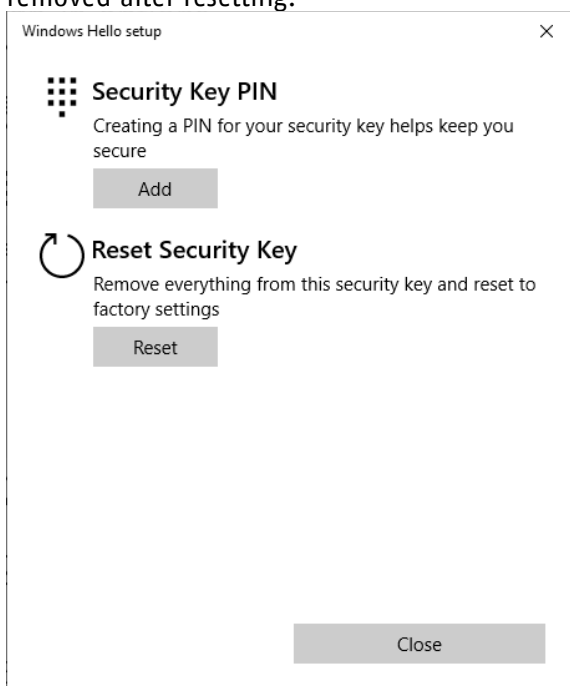
Get your Swissbit iShield Key and windows computer ready.

To launch the security key management, please click "Start --- Settings --- Accounts", then choose the option "Security key" and click the button "Manage"

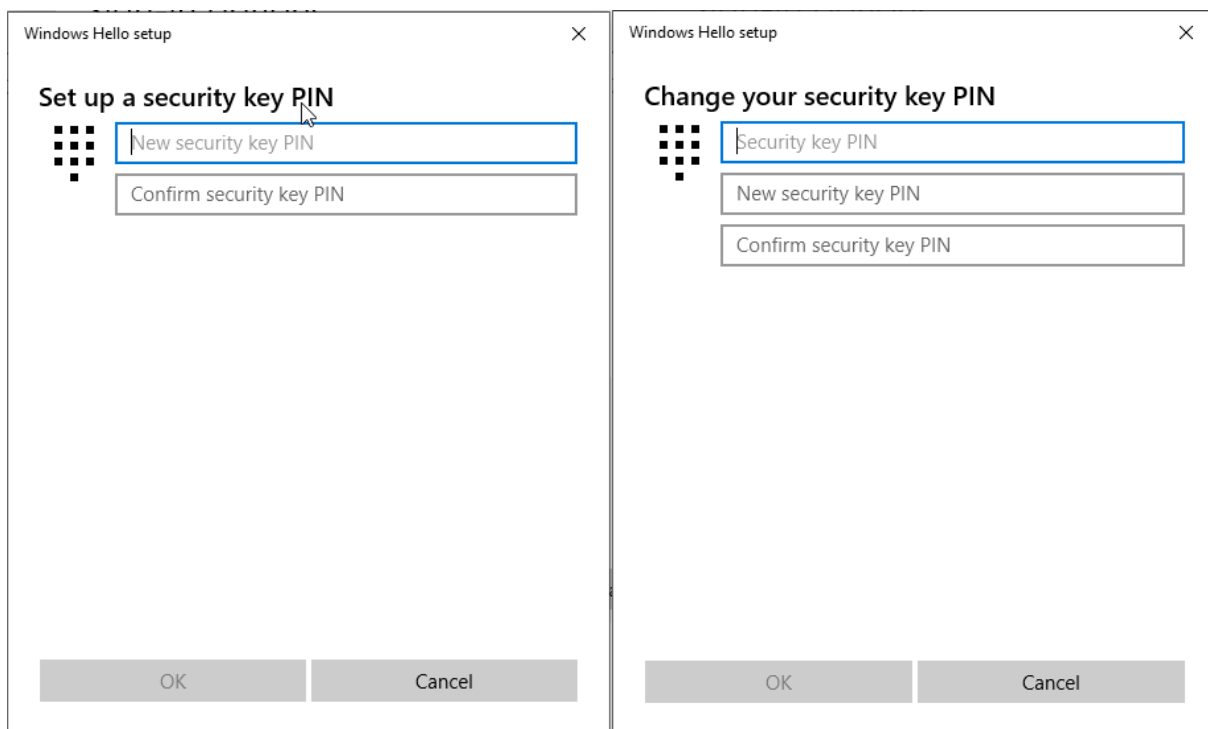


A pop-up window will prompt you to insert your security key into the USB port. Please insert the Swissbit iShield Key.

In case the Swissbit iShield Key is recognized, then you can either choose to create or change the PIN for the Swissbit iShield Key, which depends on whether there was a PIN stored previously. Meanwhile, the PIN of the Swissbit iShield Key can be reset if it was lost or forgotten. Please note that the PIN and credentials will be removed after resetting.



You could set up a new PIN if there is no PIN stored in it. To change the security PIN, the current PIN is required.



Windows Hello setup

Set up a security key PIN

New security key PIN

Confirm security key PIN

OK Cancel

Windows Hello setup

Change your security key PIN

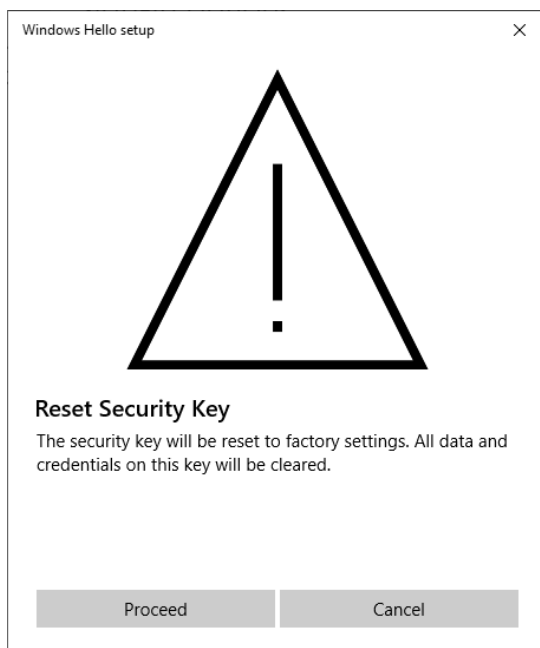
Security key PIN

New security key PIN

Confirm security key PIN

OK Cancel

If you reset the Swissbit iShield Key, please note that the credentials are lost after reset.



Windows Hello setup

Reset Security Key

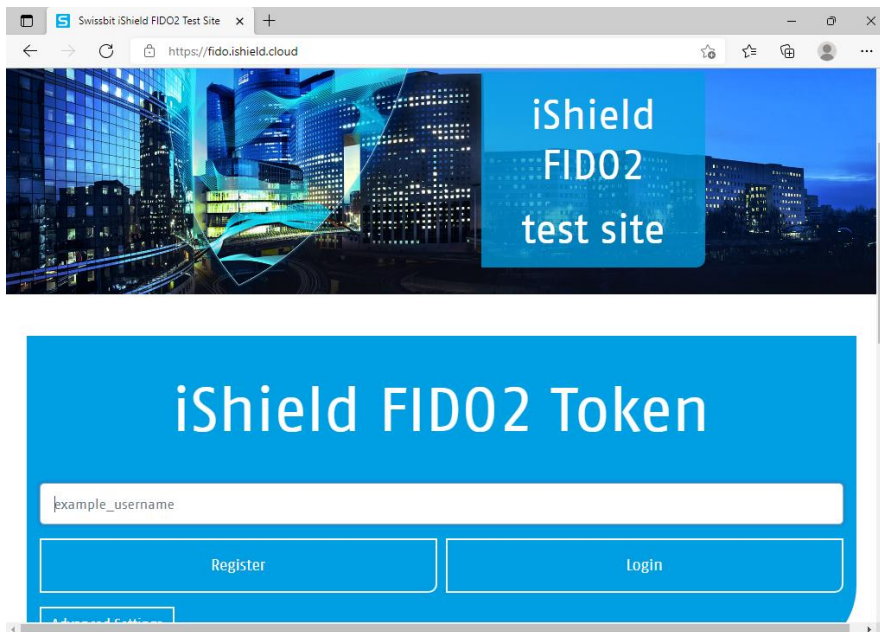
The security key will be reset to factory settings. All data and credentials on this key will be cleared.

Proceed Cancel

4.2.3 Test Registration

Please visit the test website <https://fido.ishield.cloud>, which supports [WebAuthn](#), to test your Swissbit iShield Key.

The website looks like this:



To register the Swissbit iShield Key, enter any name for credential ID, and click "Register"



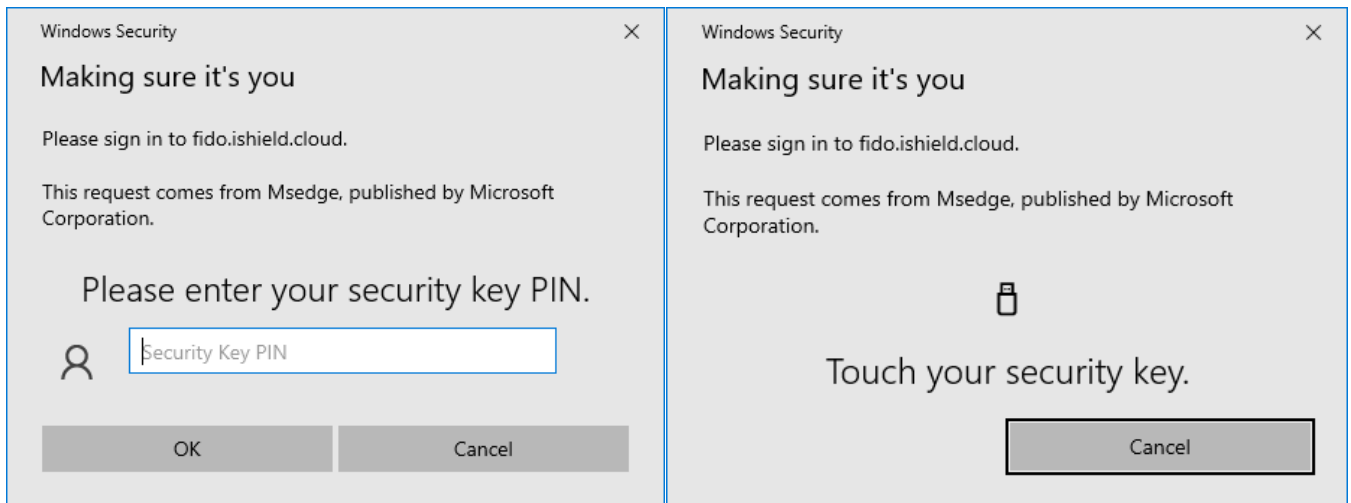
The security PIN is now required if a security PIN was setup as stated in [section 4.2.2](#). Note: If the security PIN is lost or not needed for your use case, move to [section 4.2.2](#) to reset Swissbit iShield Key. Please note that you have to register your key again after resetting.

After security PIN is accepted, you have to touch the end of the Swissbit iShield Key, to make sure that a human is now operating it and not a machine. You will be prompted to touch your security key.

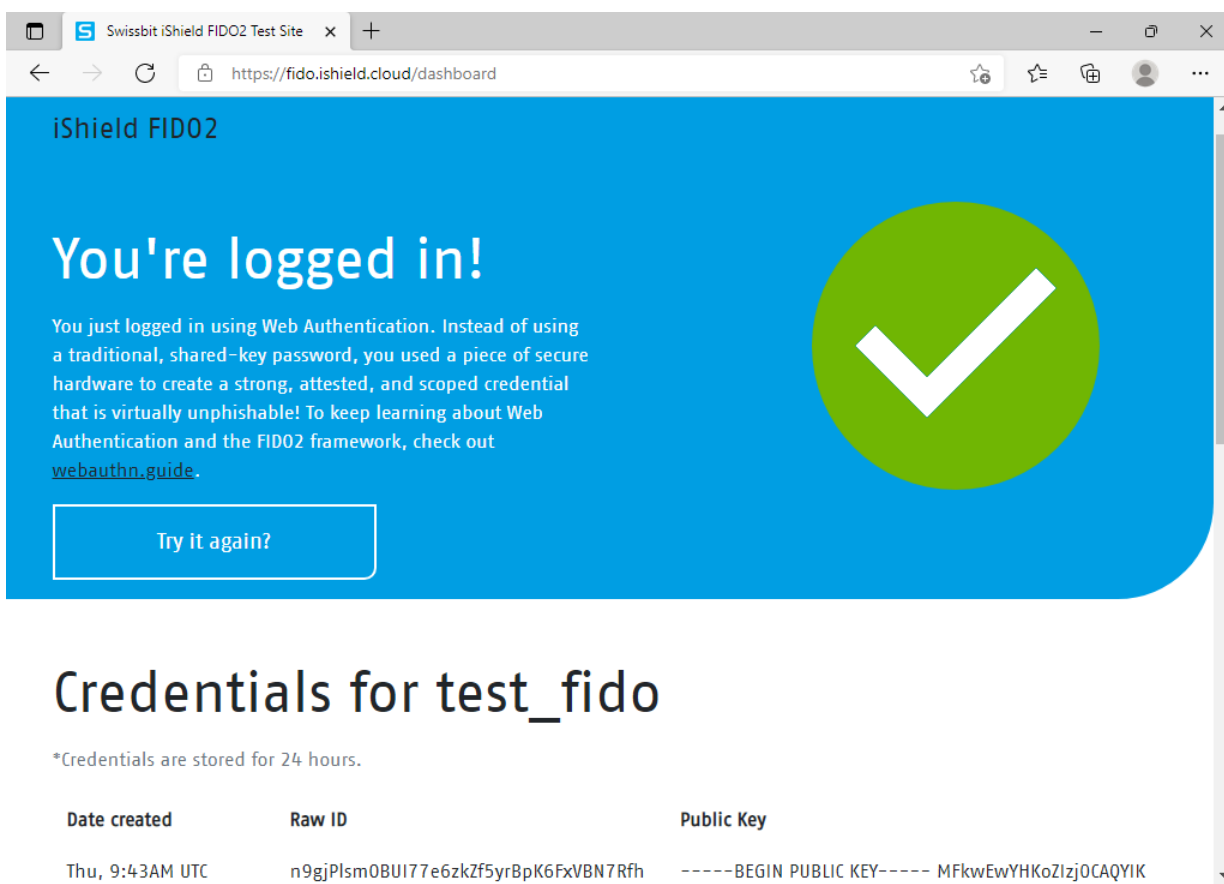
Now the registration is completed. The Swissbit iShield Key is ready to be logged in.

4.2.4 Test Login

Please enter the Test ID that you registered before, and then click "Login" Security PIN may be asked depending on whether the security PIN was set as stated in [section 4.2.2](#). When requested, touch the end of the Swissbit iShield Key to make sure a human is operating it.



Finally, you will see the success message as shown below and the credential information about the Swissbit iShield Key you used. It means that the Swissbit iShield Key is working properly.

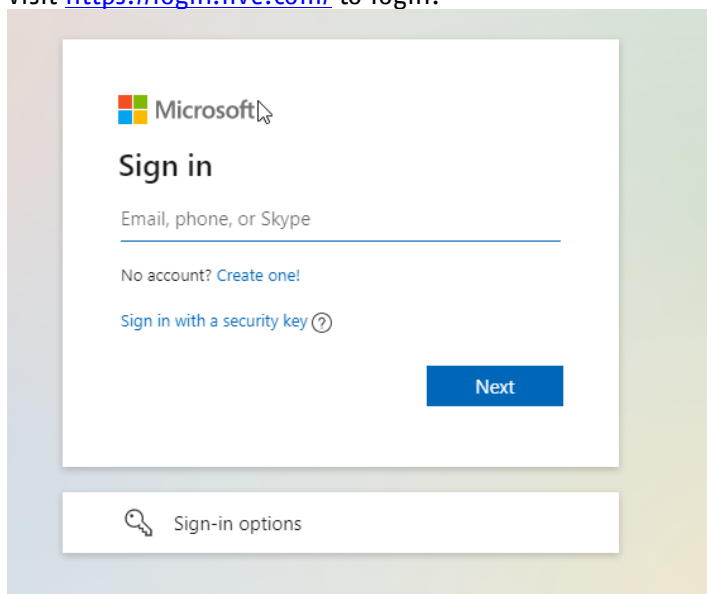


4.2.5 Register Swissbit iShield Key on an online Microsoft account

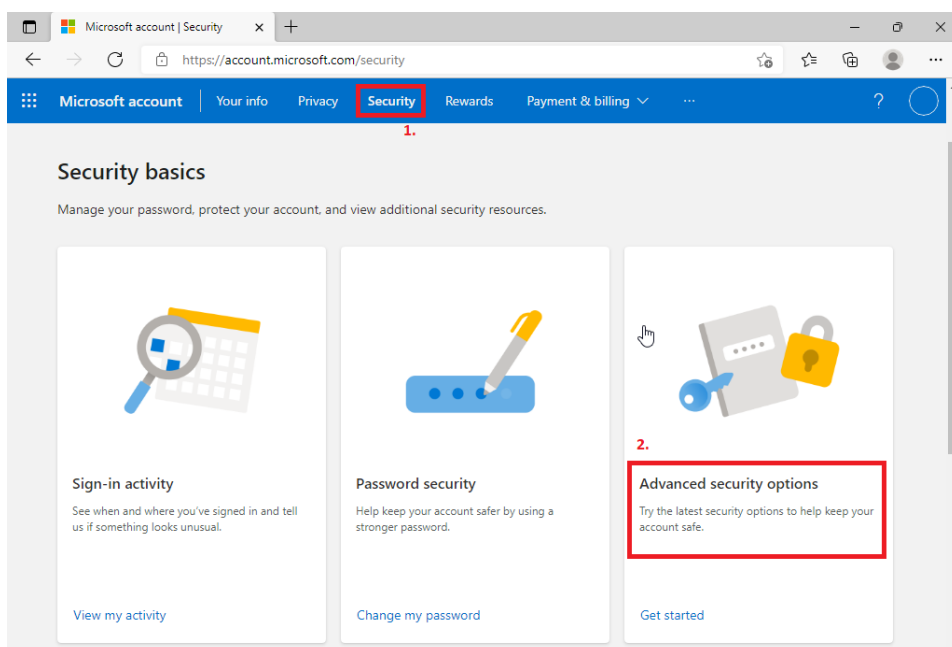
You can easily sign into your Microsoft account with the Swissbit iShield Key without giving your e-mail address and password. In this section, we will guide you how to register the Swissbit iShield Key on an "online" Microsoft account. To log into an offline Microsoft account e.g. a local Windows PC account is not covered in this section.

Note: Please get your Microsoft account ready.

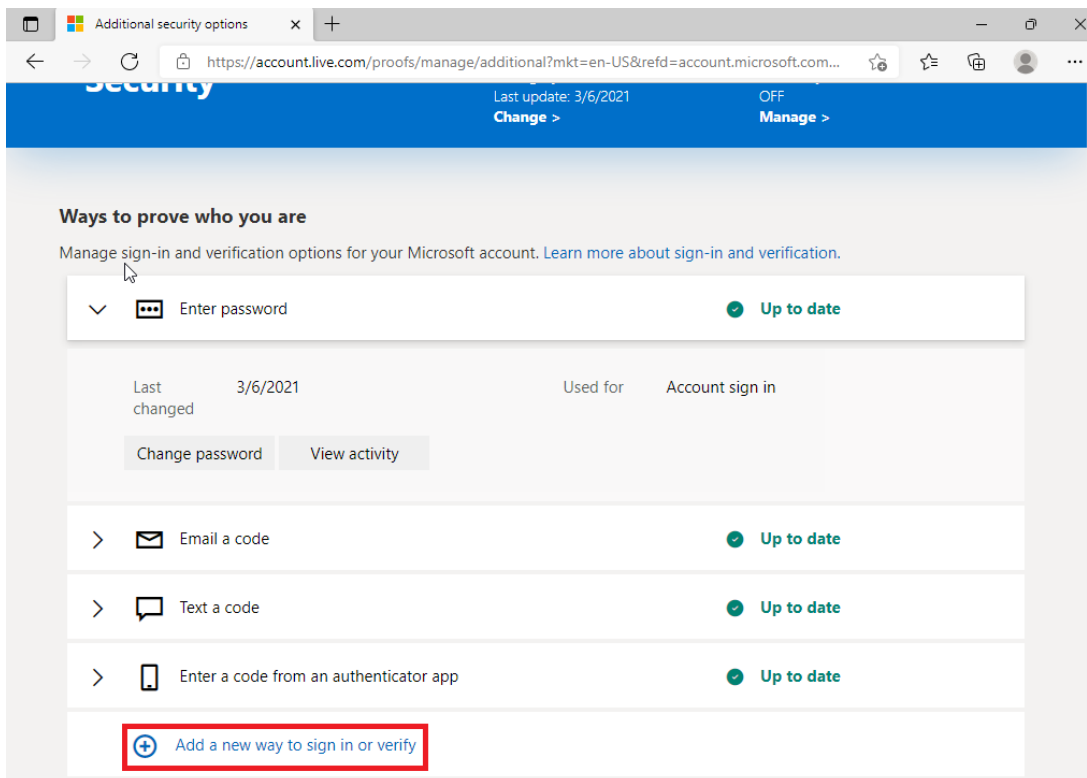
Visit <https://login.live.com/> to login.



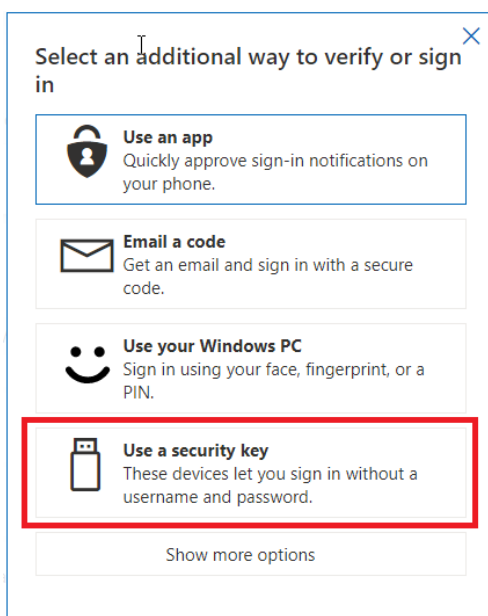
After login, you will be directed to the main page. Click "Security" from the top blue bar and choose "Advanced security options".



On this page, you can manage your activated sign-in and verification options. Click "Add a new way to sign in or verify" to add the Swissbit iShield Key as a security key.



In the following page choose "Use a security key".



Connect your Swissbit iShield Key and click "Next". As the Swissbit iShield Key is an USB security key with NFC, you could choose, either to plug it into your USB port, or to keep it close to your NFC reader.

Set up your security key

Have your key ready

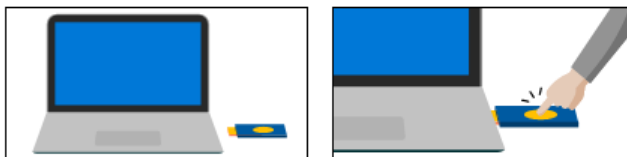


USB device



NFC device

To use a USB security key, when prompted, plug it into your USB port. Then touch the gold circle or button if your key has one when prompted for follow up action.

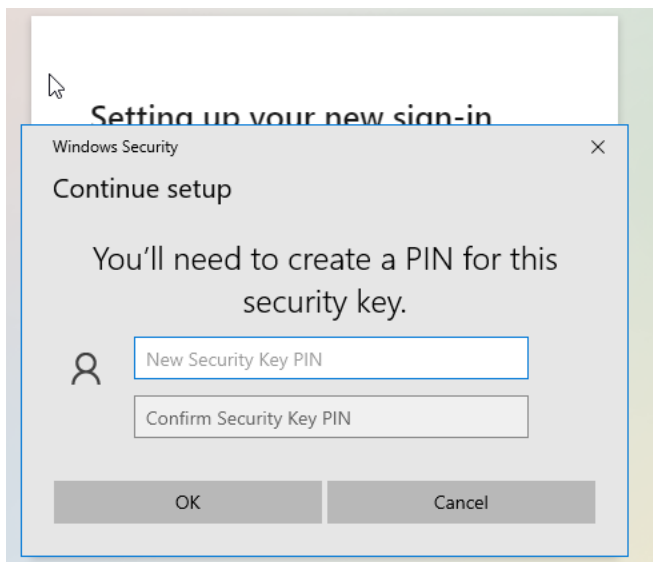


For detailed instructions on how your keys should be connected, please visit your key manufacturer's website.

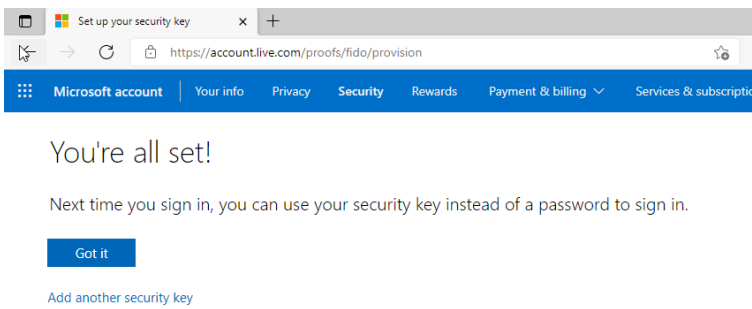
Cancel

Next

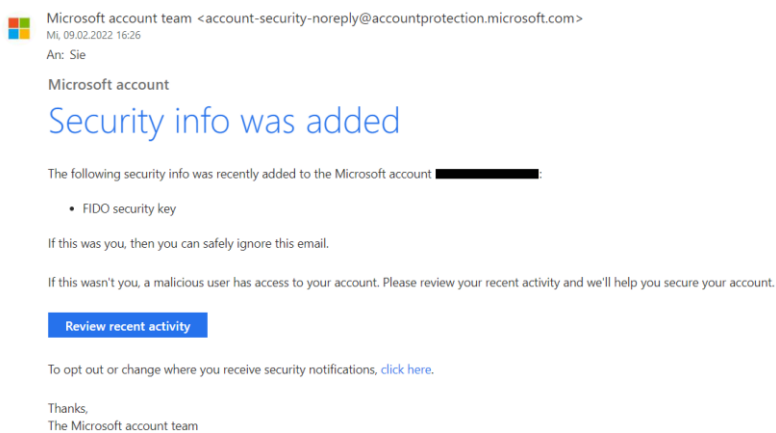
Follow the pop-up to setup your Swissbit iShield Key. Please note that Microsoft requires the user to create a PIN for the Swissbit iShield Key.



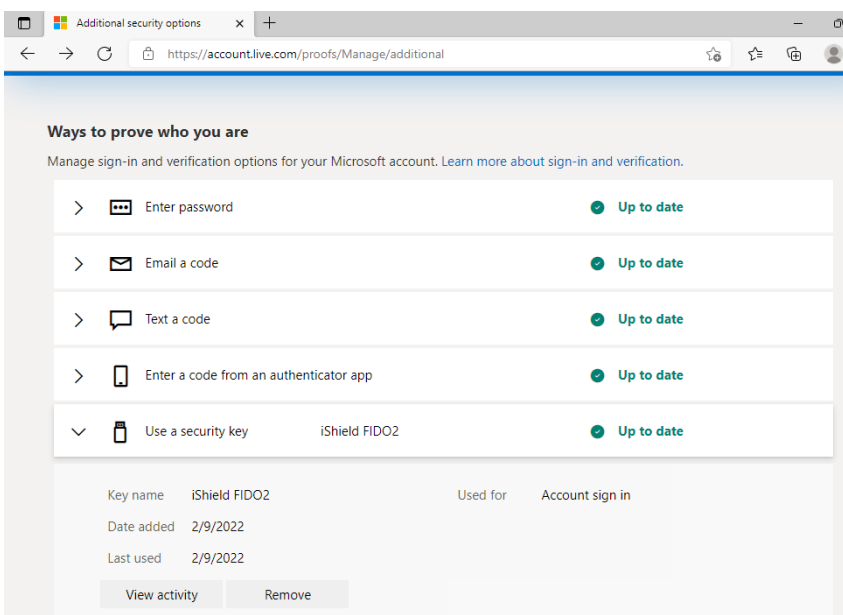
After you finish your setup, you will see the success page as shown below.



Meanwhile, you will receive an e-mail from Microsoft.



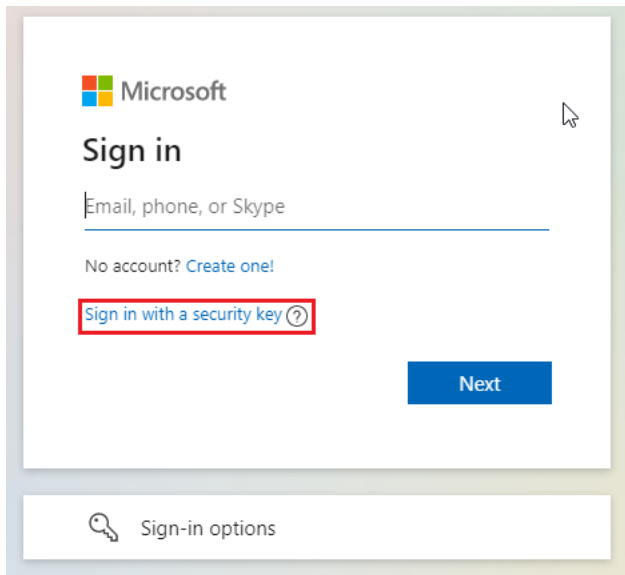
Back on the verification options page, your Swissbit iShield Key should already be listed and you can manage it anytime (in the screenshot it is named "iShield FIDO2").



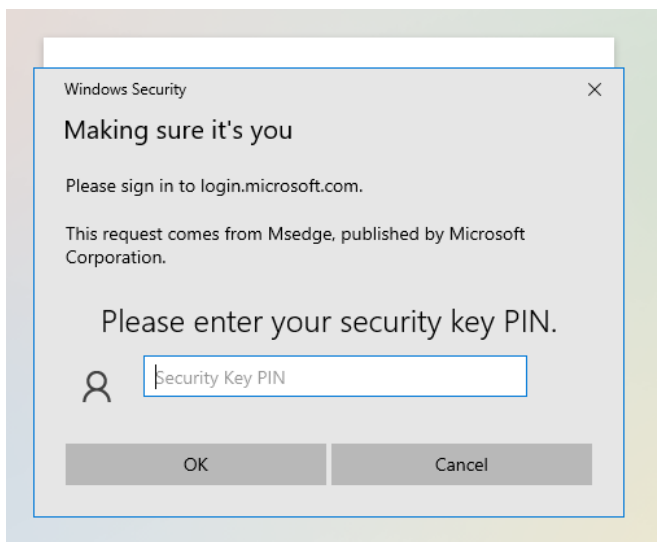
4.2.6 Usernameless/Passwordless Sign-in on an online Microsoft account

As the Swissbit iShield Key is already registered on Microsoft, you can now sign in without an e-mail address and password.

Visit <https://login.live.com/> to login, and click "Sign in with a security key". If you don't see this option, click "Sign-in Options" at the bottom and choose "Sign in with a security key".



Now plug in your Swissbit iShield Key, or keep it close to your NFC reader. When your Swissbit iShield Key is detected, enter your PIN.



Now you are successfully logged into the Microsoft account.

4.2.7 Sign-in with external Identity Provider

If you want to setup Single Sign-On with the Swissbit iShield Key, but your target service does not support FIDO/WebAuthn natively, then you can use an external Identity Provider like Keycloak. In the following section, we will demonstrate how to setup Single Sign-On with Keycloak and your Swissbit iShield Key.

This demo consists of two parts. A Dracoon App ("Dracoon") which is a third party cloud service provider and Keycloak which is an open-source Identity and Access Management service.

In this case, Dracoon requests Keycloak to authenticate a user and to secure themselves and provides a single sign-on solution. A user, who has already enabled the passwordless login by registering his Swissbit iShield Key with Keycloak as an identifier could log into Dracoon with his user name and Swissbit iShield Key without a new registration. As a user is authenticated, Keycloak will then inform Dracoon that a user was successfully authenticated and provides the identity information of this user. The type of identity information could be configured and in this case, it is the e-mail address. Finally, Dracoon will create a new user if the user is not registered.

The purpose of this guide is to walk through the steps that need to be completed for this demonstration. The pre-requisite is that the basic settings of Keycloak and Dracoon are completed.

Configuration in Keycloak

Please note that you should finish adding an OIDC client on Keycloak before starting. You can find the guide on how to add and configure an OIDC client at https://www.keycloak.org/docs/11.0/server_admin/#oidc-clients.

The screenshot shows the Keycloak Admin Console interface for configuring a client named 'dracoon'. The left sidebar contains navigation menus for 'Configure' (Realm Settings, Clients, Client Scopes, Roles, Identity Providers, User Federation, Authentication) and 'Manage' (Groups, Users, Sessions, Events, Import, Export). The main content area is titled 'Clients > dracoon' and includes tabs for Settings, Credentials, Keys, Roles, Client Scopes, Mappers, Scope, Revocation, Sessions, Offline Access, Clustering, and Installation. The 'Settings' tab is selected, showing the following configuration options:

- Client ID: dracoon
- Name: (empty)
- Description: (empty)
- Enabled:
- Always Display in Console:
- Consent Required:
- Login Theme: (dropdown)
- Client Protocol: openid-connect
- Access Type: confidential
- Standard Flow Enabled:
- Implicit Flow Enabled:
- Direct Access Grants Enabled:
- Service Accounts Enabled:
- OAuth 2.0 Device Authorization Grant Enabled:
- Authorization Enabled:
- Root URL: https://[redacted]dracoon.cloud
- Valid Redirect URIs: https://[redacted]dracoon.cloud/*
- Base URL: https://[redacted]dracoon.cloud
- Admin URL: (empty)
- Web Origins: (empty)
- Backchannel Logout URL: (empty)
- Backchannel Logout Session Required:
- Backchannel Logout Revoke Offline Sessions:

Below these settings are expandable sections for 'Fine Grain OpenID Connect Configuration', 'OpenID Connect Compatibility Modes', 'Advanced Settings', and 'Authentication Flow Overrides'. At the bottom, there are 'Save' and 'Cancel' buttons.

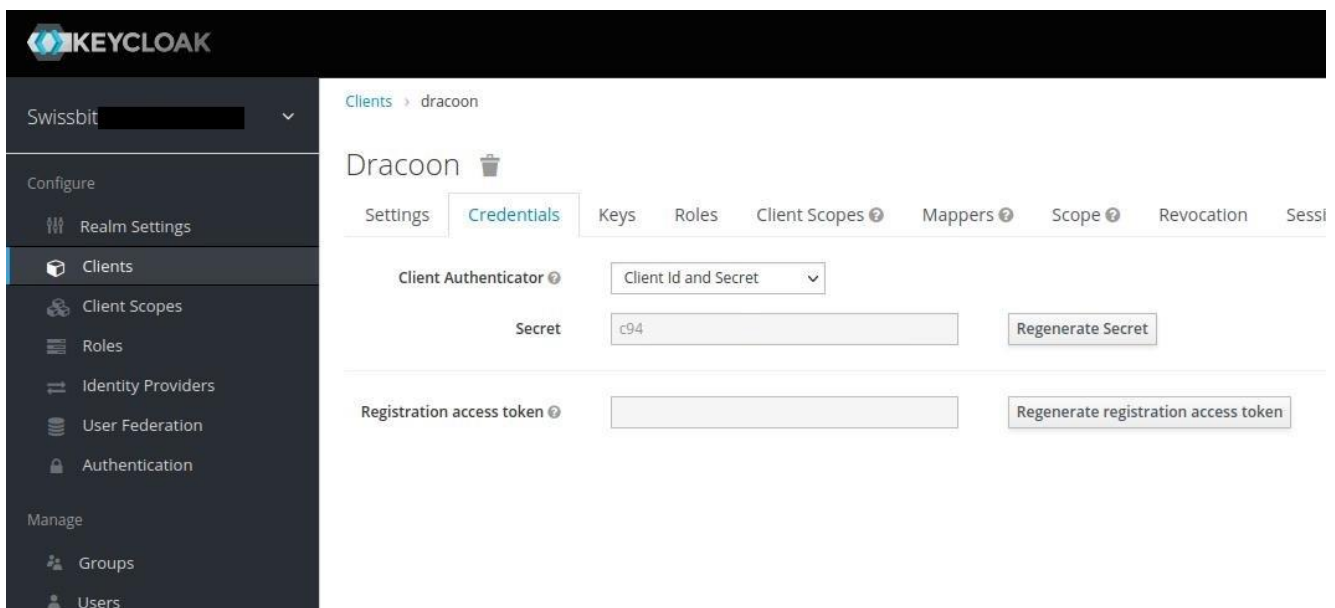
In the screenshot above, we would like to go through some important configurations.

Client Protocol: As OpenID connect protocol is being used, this value should be "openid-connect"

Root-URL and BaseURL: This value should be the URL of your Dracoon service.

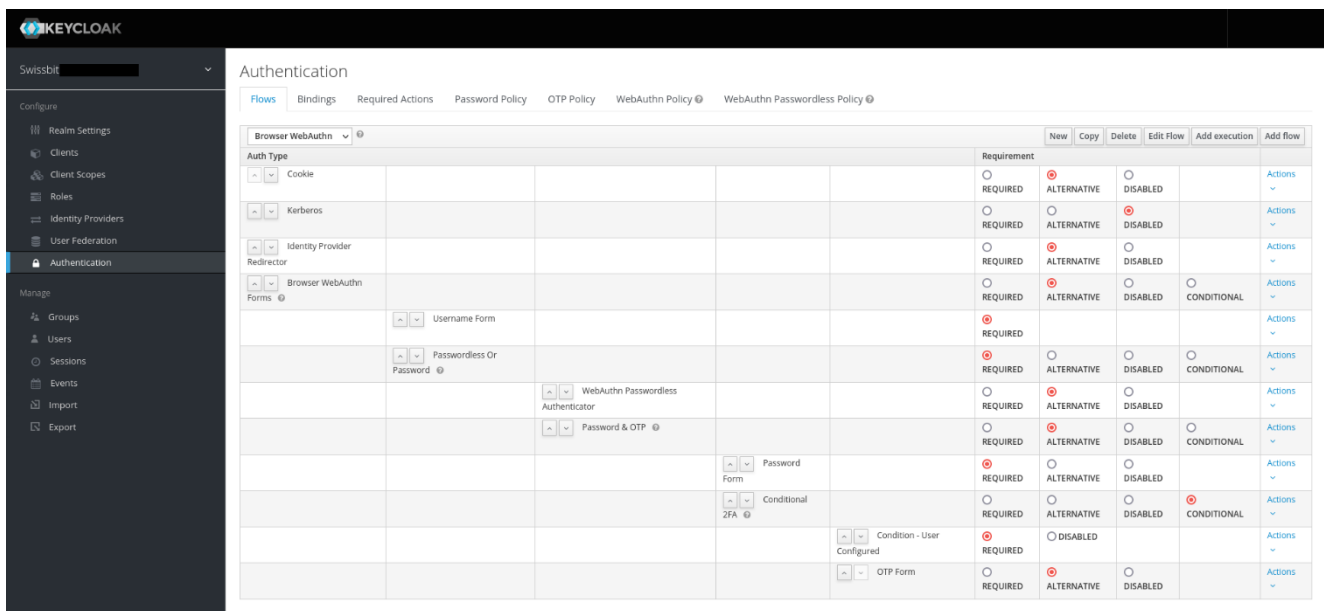
Valid Redirect URLs: This value should be the Dracoon link that you would like to redirect from. You can use Wildcards(*) at the end of a URL.

Access Type: the type of OIDC client. Please note that if it is set to confidential, the credential of clients must be configured. "Client Id and Secret" is the default setting of Client Authenticator. The secret is automatically generated for future use. For more information you can visit https://www.keycloak.org/docs/latest/server_admin/index.html#_client_credentials



For other configurations of OIDC client, please visit https://www.keycloak.org/docs/11.0/server_admin/#oidc-clients for more information.

Please enable "WebAuthn Register Passwordless" in the tab "Required Actions", and then setup a passwordless browser login flow. You can add it by following this guide https://www.keycloak.org/docs/latest/server_admin/#creating-a-password-less-browser-login-flow. After this flow has been created, click on Authentication and switch to the Tab "Flows". Then choose your flow (In our example it is "Browser WebAuthn") from the drop down list. You can now manage the Authentication types based on your use case. The screenshot shown below is an example of possible flow configuration, which allows you to use either "WebAuthn Passwordless Authenticator" or "Password & OTP".



More information can be found at https://www.keycloak.org/docs/latest/server_admin/#_authentication-flows.

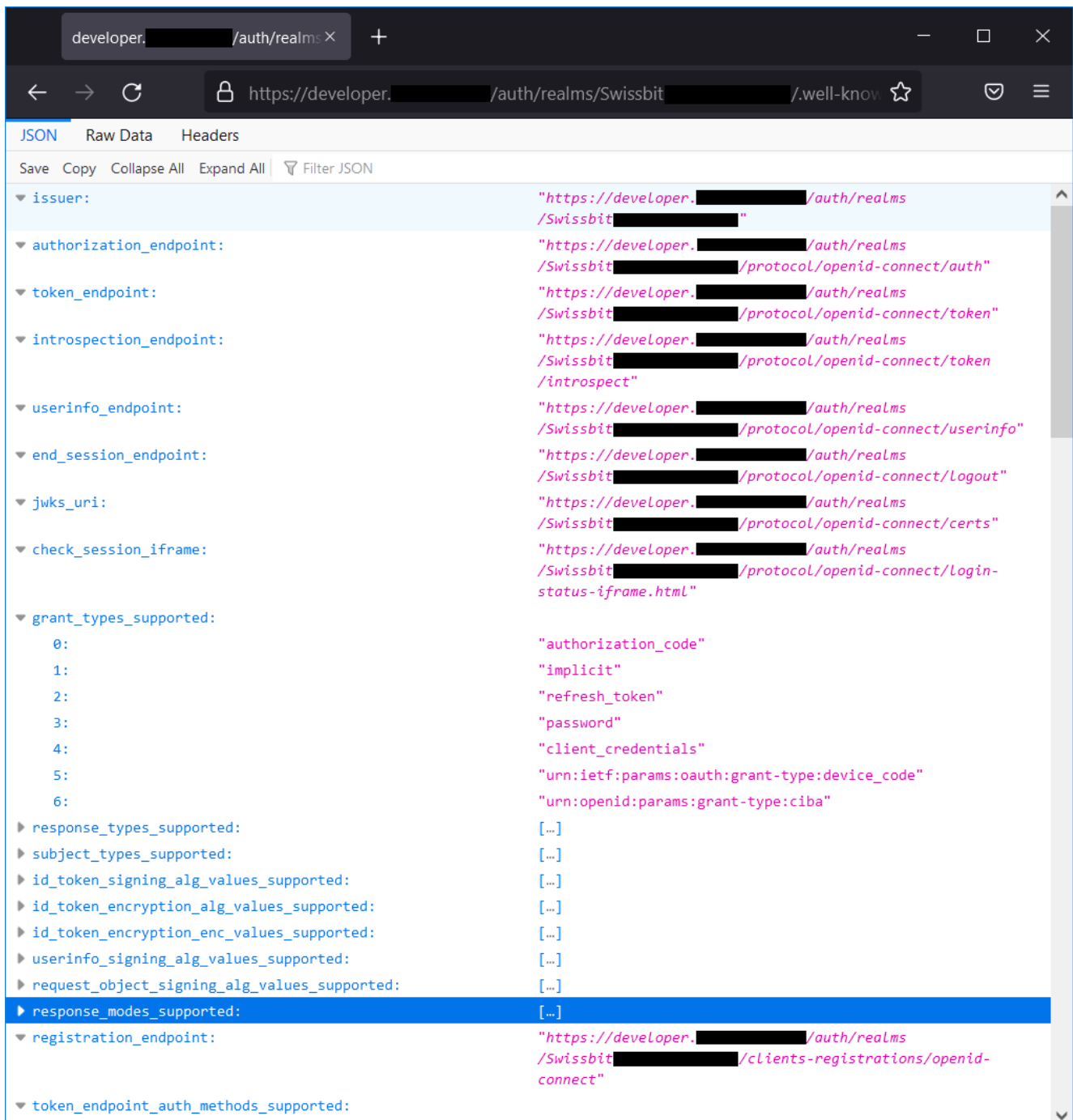
Configuration in Dracon

OpenID connect protocol is used for the communication between Keycloak server and Dracon. Please note that you should complete your basic setup of Dracon before starting.

Click "System settings --- Authentication", switch to the tab "OpenID Connect"

The screenshot shows the Dracon Administration interface. At the top left is the 'DRACON' logo and a search bar. A left sidebar contains navigation options: 'Back to all files', 'Administration' (Users, Groups), 'Configuration' (System, Policies, Security, Apps, Authentication, Storage, Logging, Subscription, Branding). The main content area is titled 'Authentication' and has three tabs: 'Local Users', 'Active Directory', and 'OpenID Connect' (which is selected). Under the 'OpenID Connect' tab, there is a section 'OpenID Connect' with a description: 'Manage your OpenID providers. Before you can enable sign-in with OpenID Connect, you must first add an OpenID provider.' To the right, there is a toggle switch for 'Enable login with OpenID Connect' which is turned on. Below that is a table for 'OpenID Provider' with one entry named 'Swissbit'. An 'Add' button is visible next to the table, and edit/delete icons are next to the 'Swissbit' entry.










Click "add" to add a new profile. This means Keycloak is applied as the Identity provider. (In the screenshot above, "Swissbit" is the profile name). The configuration value of Identity Provider can be fetched from `<Keycloak-URL>/auth/realms/{realm-name}/.well-known/openid-configuration`. This link is a JSON document describing metadata about the Identity Provider.



```
developer. [redacted] /auth/realms x +
https://developer. [redacted] /auth/realms/Swissbit [redacted] /well-know
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
▼ issuer: "https://deveLoper. [redacted] /auth/realms /Swissbit [redacted]"
▼ authorization_endpoint: "https://deveLoper. [redacted] /auth/realms /Swissbit [redacted] /protocol/openid-connect/auth"
▼ token_endpoint: "https://deveLoper. [redacted] /auth/realms /Swissbit [redacted] /protocol/openid-connect/token"
▼ introspection_endpoint: "https://deveLoper. [redacted] /auth/realms /Swissbit [redacted] /protocol/openid-connect/token /introspect"
▼ userinfo_endpoint: "https://deveLoper. [redacted] /auth/realms /Swissbit [redacted] /protocol/openid-connect/userinfo"
▼ end_session_endpoint: "https://deveLoper. [redacted] /auth/realms /Swissbit [redacted] /protocol/openid-connect/Logout"
▼ jwks_uri: "https://deveLoper. [redacted] /auth/realms /Swissbit [redacted] /protocol/openid-connect/certs"
▼ check_session_iframe: "https://deveLoper. [redacted] /auth/realms /Swissbit [redacted] /protocol/openid-connect/Login-status-iframe.html"
▼ grant_types_supported:
  0: "authorization_code"
  1: "implicit"
  2: "refresh_token"
  3: "password"
  4: "client_credentials"
  5: "urn:ietf:params:oauth:grant-type:device_code"
  6: "urn:openid:params:grant-type:ciba"
▶ response_types_supported: [...]
▶ subject_types_supported: [...]
▶ id_token_signing_alg_values_supported: [...]
▶ id_token_encryption_alg_values_supported: [...]
▶ id_token_encryption_enc_values_supported: [...]
▶ userinfo_signing_alg_values_supported: [...]
▶ request_object_signing_alg_values_supported: [...]
▶ response_modes_supported: [...]
▼ registration_endpoint: "https://deveLoper. [redacted] /auth/realms /Swissbit [redacted] /clients-registrations/openid-connect"
▼ token_endpoint_auth_methods_supported:
```

The configuration value that you got from above should be entered in Dracon as shown below.

Configure OpenID Provider

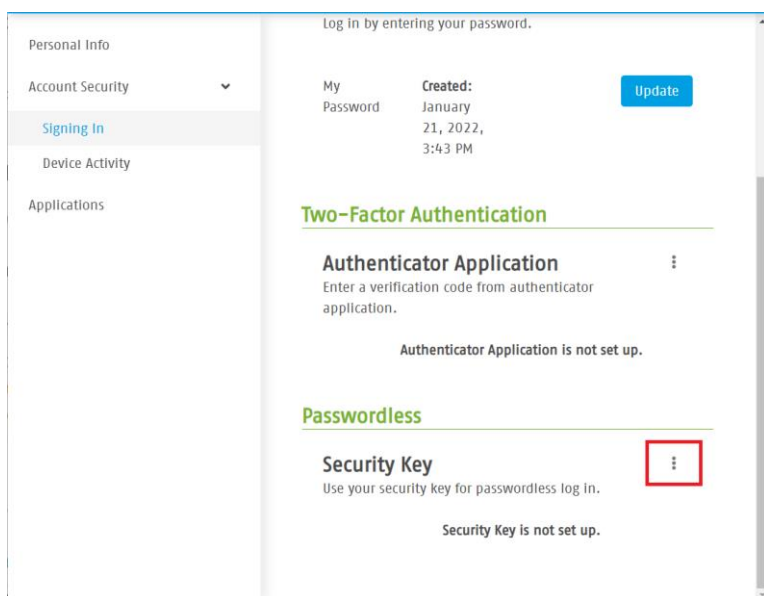
Name: *	Swissbit
Issuer URL: *	https://developer.██████████/auth/realms/Swissbit ██████████
Authorization Endpoint URL: *	https://developer.██████████/auth/realms/Swissbit ██████████
Token Endpoint URL: *	https://developer.██████████/auth/realms/Swissbit ██████████
User Info Endpoint URL: *	https://developer.██████████/auth/realms/Swissbit ██████████
JWKS Endpoint URL: *	https://developer.██████████/auth/realms/Swissbit ██████████
User Management URL:	User Management URL
Client ID: *  
Client Secret: *  
Scopes: *	email 
	openid  
Redirect URIs: *	https://██████████.dracoon.cloud 
Authorization Mode:	Authorization Code 
Proof Key for Code Exchange (PKCE):	<input type="checkbox"/>

As a final step, save the settings using the Save button. OpenID can now be used as an authentication method in Dracoon.

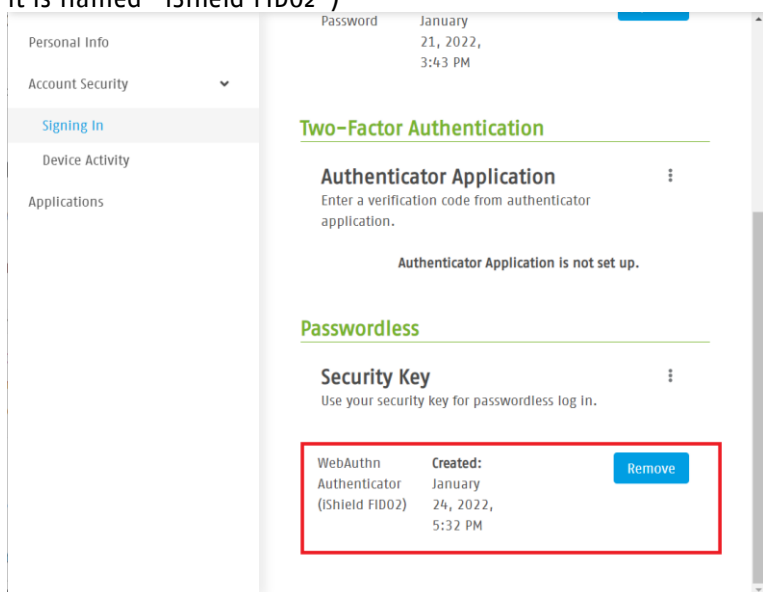
You can visit <https://cloud.support.dracoon.com/hc/en-us/articles/360001372679-OpenID-Connect-Keycloak> for more information about OpenID connect client configuration.

Swissbit iShield Key Registration

When the user authenticates on the account site of Keycloak, the user may choose multiple ways to sign in. You can find the account site at : `<Keycloak-URL>/auth/realms/{realm-name}/account/` Expand "Account Security" and then click the "Signing In" option and set up a new Security Key by clicking the configure symbol. In the screenshot below the user has not configured any security key.



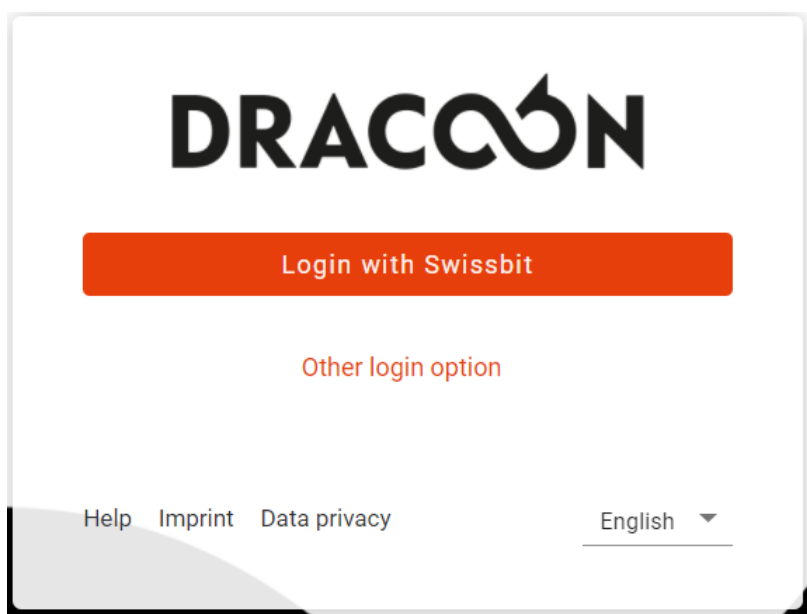
Please follow the instructions to finish setting up your Swissbit iShield Key. The registered Swissbit iShield Key is listed below. In the screenshot above, the user has configured one passwordless security key (in the screenshot it is named "iShield FIDO2")



Single Sign-On Test

Now it is time to test the Single Sign-On functionality.

Visit the Dracon website or app and there should be two login options. Choose "Login with <Profile Name>". (In the example below it is, "Login with Swissbit")



You will automatically be redirected to Keycloaks login interface. Then enter your user name and choose the option "Use your security key for passwordless sign in" under the Sign in button instead of entering a password.

Password

[Forgot Password?](#)

SIGN IN

Sign in by entering your password.

Use your security key for passwordless sign in.

Choose "Sign In with Security Key"

SIGN IN WITH SECURITY KEY

Sign in by entering your password.

Use your security key for passwordless sign in.

Follow the pop up instruction to log in. If your sign in is successful, you will be automatically redirected to Dracoon. A new account is now created. You are now ready to use Dracoon.

If something goes wrong or you cancel this process, you can see an error information and the name of the registered Swissbit iShield Key (in the screenshot it is named "iShield FID02").

Available authenticators

- WebAuthn Authenticator (iShield FID02)

Rechteckiges Ausschneiden

TRY AGAIN

Sign in by entering your password.

Use your security key for passwordless sign in.

4.3 Swissbit iShield Key on various services

In this section, we would like to guide you how to register Swissbit iShield Key as a security key to enable 2-factor authentication on various services.

4.3.1 Auth0

Auth0 (<https://www.auth0.com/>) is an identity provider like KeyCloak and it supports WebAuthn for multi factor authentication. In this section, we will guide you how to enable the WebAuthn with FIDO on Auth0.

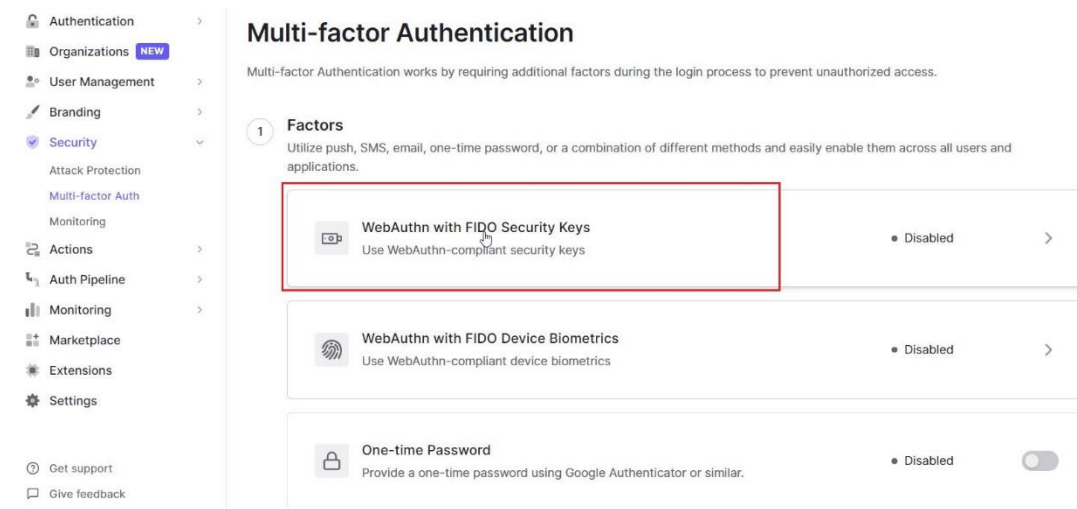
As pre-requisite, you should get your Auth0 instance ready.

Go to Dashboard, Click "Security" and choose "Multi-factor Auth"

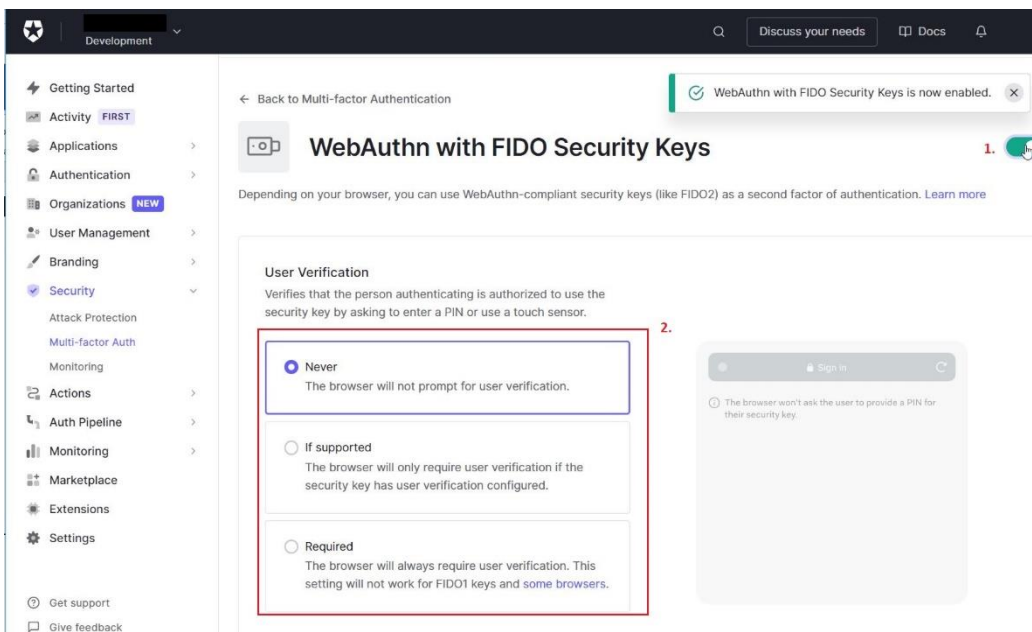
The screenshot shows the Auth0 dashboard interface. On the left, a navigation sidebar is visible with the 'Security' menu item selected and highlighted with a red box. Under 'Security', the 'Multi-factor Auth' option is also highlighted with a red box. The main content area is titled 'Getting Started' and contains three instructional cards:

- Try your Login box:** With Auth0 your authentication experience is ready to go. Customize it to match your brand identity and try it now to see how it works. Includes links for 'Try it out' and 'Customize'.
- Invite your team members:** Add additional admins to help with your integration and act as a backup account in case you lose access. Includes a link to 'Learn more about Tenant Administrator permissions'.
- Integrate Auth0 into your application:** Add Auth0 to any kind of application and technology or use one of our sample apps to get you started in minutes.

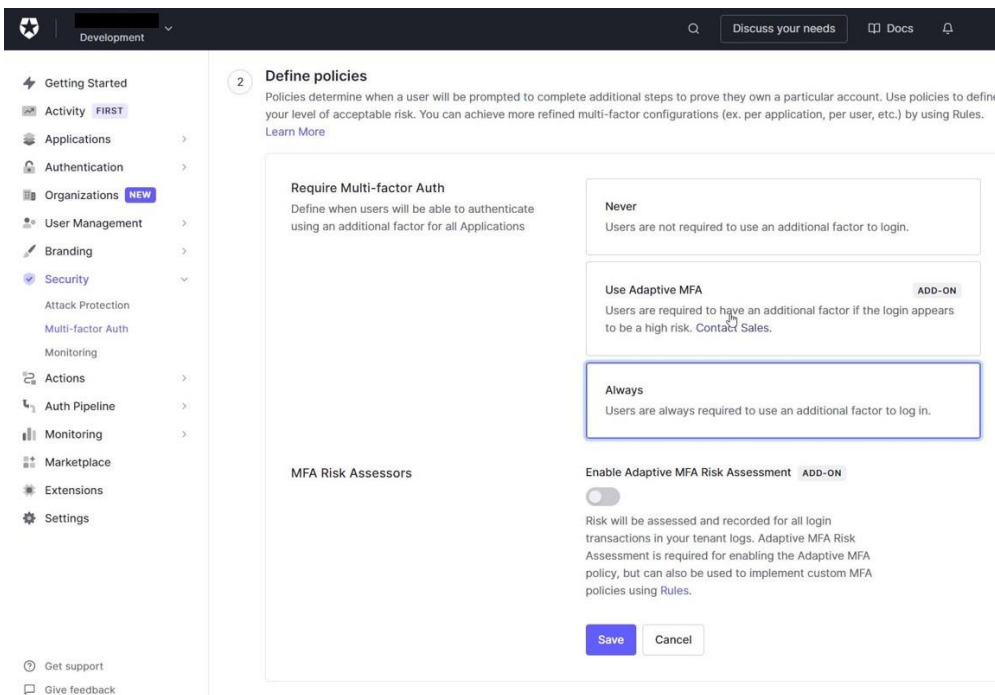
Click "WebAuthn with FIDO Security Keys"



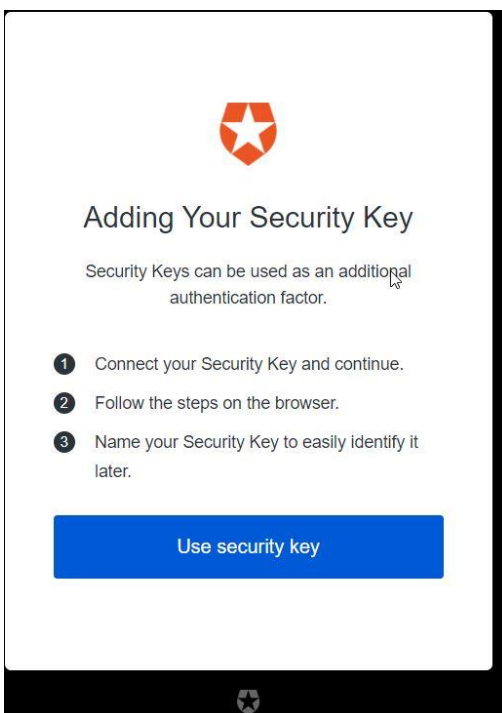
Toggle the switch (1.) to enable the “WebAuthn with FIDO Security Keys”
 For 2. Setting is shown below, you could change the verification condition when the Swissbit iShield Key is being registered as a security key. If you choose “If supported” or “Required”, a PIN is required when the Swissbit iShield Key is being used as a second authentication factor.



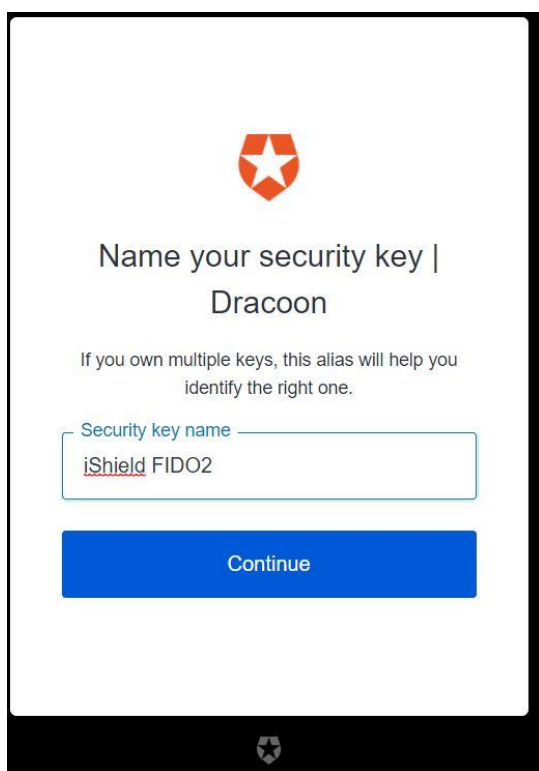
Back to the previous content (Multi-factor Authentication), you could now define whether the Multi-factor Authentication that you just enabled is always required. Click “Save” to save your setting. In the screenshot below, “Always” means user is always required to use the Swissbit iShield Key for authentication.



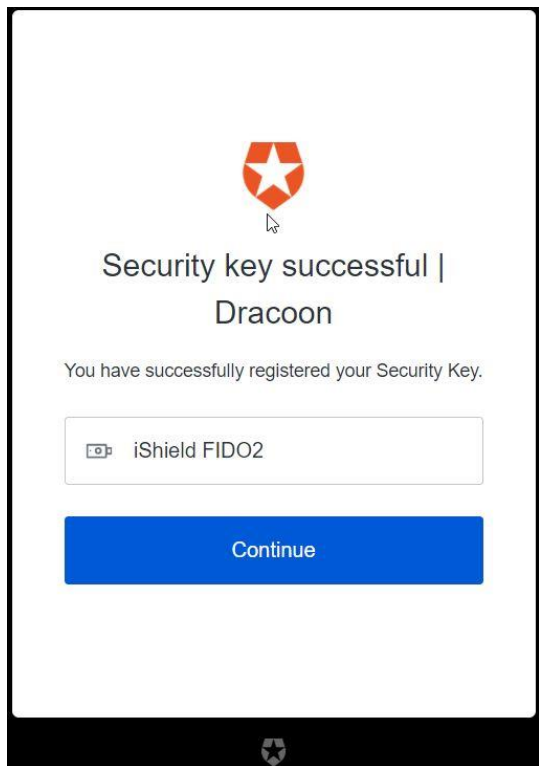
After this settings, as the security key is not registered, the user is asked to add one.



Following the pop up to register your Swissbit iShield Key (in the screenshot it is named "iShield FID02"), then you could give an alias for it.



Finally, you have successfully registered your Swissbit iShield Key. Now you could use it to login.



4.3.2 Bitbucket

After Login, go to the Personal settings and click "Two-step verification" under the security group. Then you can see you must setup SSH on your account before you are able to enable the two step verification. You can visit <https://support.atlassian.com/bitbucket-cloud/docs/set-up-an-ssh-key/> for more information about SSH configuration at Bitbucket.

Bitbucket
Your work
Repositories
Projects
More ▾
Create ▾

Personal settings

GENERAL

Account settings

Email aliases

Notifications

ACCESS MANAGEMENT

App authorizations

App passwords

SECURITY

SSH keys

Two-step verification

Sessions

Audit log


FEATURES

Labs

Two-step verification

Make your account safer!

Two-step verification secures your account by requiring a second confirmation, in addition to your password, to access your account. That second step means your account stays secure even if your password is compromised.



Get ready for two-step verification

It looks like you aren't quite ready to set up two-step verification on your account. You'll need to address the following items before continuing.

- 1

Set up SSH on your account

Once you've enabled two-step verification on your account, you will only be able to clone, push, or pull your repository over SSH. Your HTTPS access to Bitbucket repositories will be disabled. With SSH, you'll also be able to recover your account should you lose your device.

Manage your SSH keys
Learn more about SSH
- ✓

Confirm your primary email address

Your primary email is ██████████ (Change)
- 3

Make sure you've updated any applications that connect to Bitbucket

Enabling two-step verification will disable all remote HTTPS actions for Git and the Bitbucket API. Any applications which use HTTPS to access Bitbucket will be impacted. [Learn more.](#)

Once you've completed the steps above, simply return to this page to continue. See you soon!

After the SSH setup, visit "Two-step verification" site again, you are now able to enable the two-step verification with an app. Please follow the guide to complete the setup.

Personal settings

GENERAL

- Account settings
- Email aliases
- Notifications

ACCESS MANAGEMENT

- App authorizations
- App passwords

SECURITY

- SSH keys
- Two-step verification**

- Sessions
- Audit log

FEATURES

- Labs

Two-step verification

Setting up two-step verification is easy, just follow the steps below.

- Download a two-step verification app
 - iPhone, iPod Touch, or iPad: Authy for iOS
 - Android devices: Authy for Android
 - Windows devices: Microsoft Authenticator

- Scan this QR code with your verification app



Can't scan the code? You can add the code to your application manually using the following details:

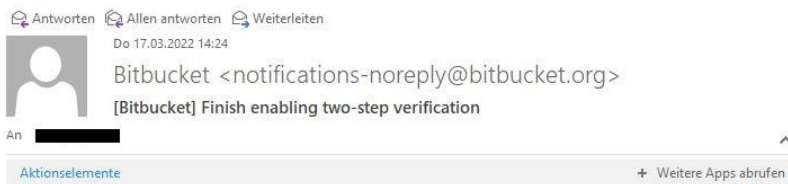
Account: [redacted]
 Key: LR26 ZSPD [redacted]
 Time based: Yes

- Enter the resulting verification code

Can't find your code? [Get help.](#)

[Enable two-step verification](#)

After that you will receive a confirmation Email from Bitbucket. Don't forget to click the link from the mail to confirm that you are enabling two-step verification.



Bitbucket Finish enabling two-step verification

To finish enabling two-step verification, we need to make sure that you're the one who set it up. Click the button below to finish enabling two-step verification.



[Enable two-step verification](#)

Now you could register your Swissbit iShield Key as a security key at Bitbucket. Give the device name and click "Add security key" on the right side (in the screenshot it is named "iShield FIDO2").

Personal settings

GENERAL

Account settings
Email aliases
Notifications

ACCESS MANAGEMENT

App authorizations
App passwords

SECURITY

SSH keys
Two-step verification
Sessions
Audit log

FEATURES

Labs

Two-step verification **ENABLED**

Disable two-step verification

Recovery codes

Show recovery codes

Security keys

Security keys are hardware devices that you insert when signing in, replacing the need to enter a verification code. We only support security keys that use the FIDO and FIDO2 standards.

To link a security key to your account, enter a device name, click **Add security key**, and then insert your security key and press the device's button.

Device name	Added on	Last used	
iShield FIDO2			Add security key

You haven't added any security keys.

Follow the pop-up instructions; finally, you have successfully registered your Swissbit iShield Key. Now you could use it to login.

Personal settings

GENERAL

Account settings
Email aliases
Notifications

ACCESS MANAGEMENT

App authorizations
App passwords

SECURITY

SSH keys
Two-step verification
Sessions
Audit log

FEATURES

Labs

Two-step verification **ENABLED**

Disable two-step verification

Recovery codes

Show recovery codes

Security keys

Security keys are hardware devices that you insert when signing in, replacing the need to enter a verification code. We only support security keys that use the FIDO and FIDO2 standards.

To link a security key to your account, enter a device name, click **Add security key**, and then insert your security key and press the device's button.

Device name	Added on	Last used	
What do you want to call your device?			Add security key
iShield FIDO2	just now	Never	✕

4.3.3 Github

Go to settings, click "Password and authentication" under the tab "Access", then choose "Security keys" from Two-factor methods.

Account

- Appearance
- Accessibility
- Notifications

Access

- Billing and plans
- Emails
- Password and authentication**
- SSH and GPG keys
- Organizations
- Moderation

Code, planning, and automation

- Repositories
- Packages
- Pages
- Saved replies

Security

- Code security and analysis

Integrations

- Applications
- Scheduled reminders

Archives

- Security log
- Sponsorship log

Developer settings

Old password

New password

Confirm new password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

[Update password](#) [I forgot my password](#)

Two-factor authentication Enabled

Two-factor authentication adds an additional layer of security to your account by requiring more than just a password to sign in. [Learn more.](#)

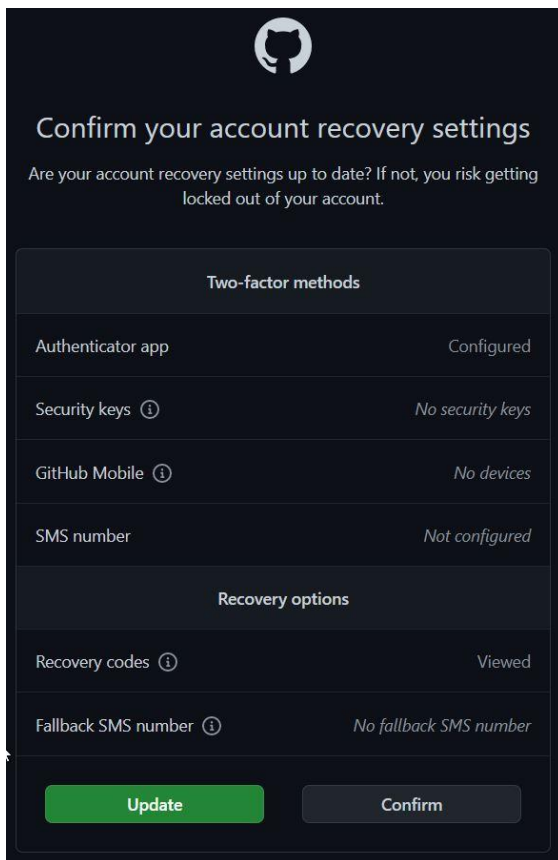
Two-factor methods

Authenticator app	Configured	Edit
Security keys ¹	No security keys	Add
GitHub Mobile ¹	No devices	Show
SMS number	Not configured	Edit

Recovery options

Recovery codes ¹	Viewed	Show
Fallback SMS number ¹	No fallback SMS number	Add

Now you should confirm your account recovery settings. Please note that you must finish setting up an Authenticator app and Recovery code before the next step.



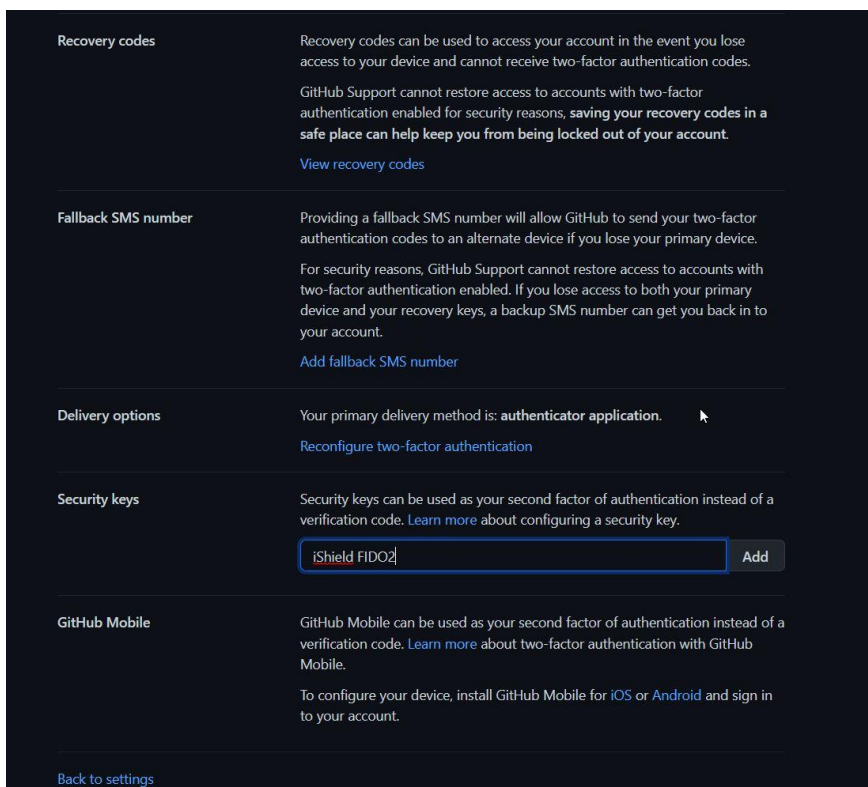
The screenshot shows a dark-themed GitHub interface for confirming account recovery settings. At the top is the GitHub logo. Below it is the heading "Confirm your account recovery settings" and a warning: "Are your account recovery settings up to date? If not, you risk getting locked out of your account." The settings are organized into two sections: "Two-factor methods" and "Recovery options".

Two-factor methods	
Authenticator app	Configured
Security keys ⓘ	No security keys
GitHub Mobile ⓘ	No devices
SMS number	Not configured

Recovery options	
Recovery codes ⓘ	Viewed
Fallback SMS number ⓘ	No fallback SMS number

At the bottom, there are two buttons: a green "Update" button and a grey "Confirm" button.

Now you can define a name for your Swissbit iShield Key (in the screenshot it named "iShield FIDO2").



The screenshot shows the configuration page for account recovery settings. It includes sections for "Recovery codes", "Fallback SMS number", "Delivery options", "Security keys", and "GitHub Mobile".

Recovery codes: Recovery codes can be used to access your account in the event you lose access to your device and cannot receive two-factor authentication codes. GitHub Support cannot restore access to accounts with two-factor authentication enabled for security reasons, saving your recovery codes in a safe place can help keep you from being locked out of your account. [View recovery codes](#)

Fallback SMS number: Providing a fallback SMS number will allow GitHub to send your two-factor authentication codes to an alternate device if you lose your primary device. For security reasons, GitHub Support cannot restore access to accounts with two-factor authentication enabled. If you lose access to both your primary device and your recovery keys, a backup SMS number can get you back in to your account. [Add fallback SMS number](#)

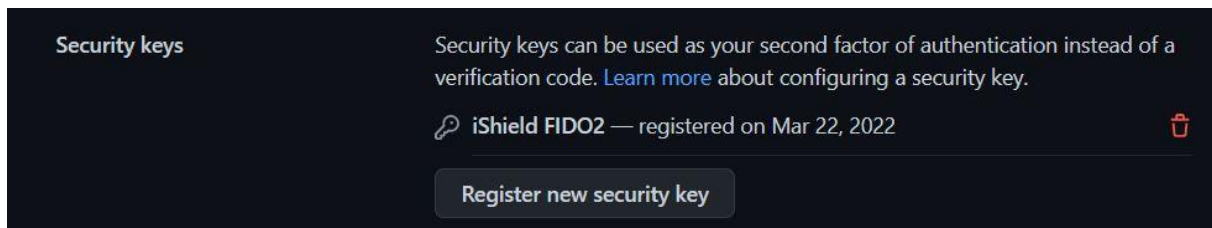
Delivery options: Your primary delivery method is: **authenticator application**. [Reconfigure two-factor authentication](#)

Security keys: Security keys can be used as your second factor of authentication instead of a verification code. [Learn more](#) about configuring a security key.

GitHub Mobile: GitHub Mobile can be used as your second factor of authentication instead of a verification code. [Learn more](#) about two-factor authentication with GitHub Mobile. To configure your device, install GitHub Mobile for [iOS](#) or [Android](#) and sign in to your account.

[Back to settings](#)

Follow the pop up instruction and finally you have successfully registered your Swissbit iShield Key. Now you could use it to login.



4.3.4 Amazon Web Service (AWS)

After you log into the AWS Management console, click your ID at top-right side, and choose "Security credentials" (1.). On the bottom you can find the option "Assign MFA device" (2.).

The screenshot shows the AWS IAM console interface. On the left is the navigation menu for Identity and Access Management (IAM). The main content area is titled 'Password for console access' and includes a 'Change password' button. Below that is the 'Access keys for CLI, SDK, & API' section, which contains a 'Create access key' button and a table of existing access keys. The table has columns for 'Access key ID', 'Status', 'Created', and 'Last used'. One access key is listed with an 'Active' status, created on 2020-04-28 20:23 UTC+0100. Below the access keys is the 'Multi-factor authentication (MFA)' section, which includes an 'Assign MFA device' button. A dropdown menu is open on the right side of the console, showing options like 'Account ID', 'IAM user', 'Account', 'Organization', 'Service Quotas', 'Billing Dashboard', 'Security credentials' (highlighted with a red box and labeled '1.'), 'Switch role', and 'Sign out'.

Access key ID	Status	Created	Last used
[Redacted]	Active	2020-04-28 20:23 UTC+0100	N/A

Choose "U2F security key" and click continue

Manage MFA device ✕

Choose the type of MFA device to assign:

Virtual MFA device
Authenticator app installed on your mobile device or computer

U2F security key
YubiKey or any other compliant U2F device

Other hardware MFA device
Gemalto token

For more information about supported MFA devices, see [AWS Multi-Factor Authentication](#)

Cancel Continue

Insert your Swissbit iShield Key and touch the end side of your Swissbit iShield Key. Your Swissbit iShield Key will be automatically detected.

Set up U2F security key ✕

[See information about supported configurations for using U2F security keys](#)

1. Insert your U2F security key into your computer's USB port.



2. Tap the button or gold disk on your U2F security key.
Waiting for security key...

[Troubleshoot U2F](#) Cancel Previous Tap U2F key

Finally, you have successfully registered your Swissbit iShield Key. Now you could use it to login.

Set up U2F security key ✕

✓ Setup is complete for this U2F security key.
Tapping this U2F security key is now required during sign-in.

Close

You could manage your security key by clicking the button "Manage MFA device".

5 TOTP Applications

5.1 Overview

Some product variants of the iShield Key support the generation of Time-based One Time Passwords. You can use the TOTP function of the iShield Key for two-factor authentication with services that support the implemented TOTP algorithm. The iShield Key implements the [RFC 6238](#) algorithm by IETF. There are 42 slots for TOTP credentials available. Therefore, you can use your iShield Key for two-factor authentication with various services and accounts. Optionally, you can configure the TOTP generation for specific accounts to be PIN-protected. Then, the PIN is required to generate a new one-time password.

Section 5.1.1 covers setup of your iShield Key as second factor for a service, section 5.1.2 depicts the computation of time-based one-time passwords and section 5.1.3 illustrates the passcode generation and authentication with TOTP.

5.1.1 Registration

The first step is to pair your iShield Key with the account for which you want to use the TOTP function of your security key as second factor. The validation server and iShield Key need to share a secret key and use the same configuration values for the algorithm, digits and interval. Most services use the SHA1 algorithm in the TOTP computation as this is the default choice but some services will also support SHA256 and SHA512. Those algorithms are optional in the standard but are supported by the iShield Key. The iShield Key can generate one-time codes of 6 to 9 digits. The period, i.e. the time interval a code is considered valid, is also customizable but most services will use the default value of 30 seconds. If you have set a PIN, you can also choose to protect your TOTP generation with it. Then your PIN is required anytime a new password is requested. **Please keep in mind that your PIN is irreversibly blocked once you entered it incorrectly 10 times in a row. You will not be able to authenticate with your PIN-protected slots anymore, but will have an option to reset the PIN with the loss of PIN-protected slots.**

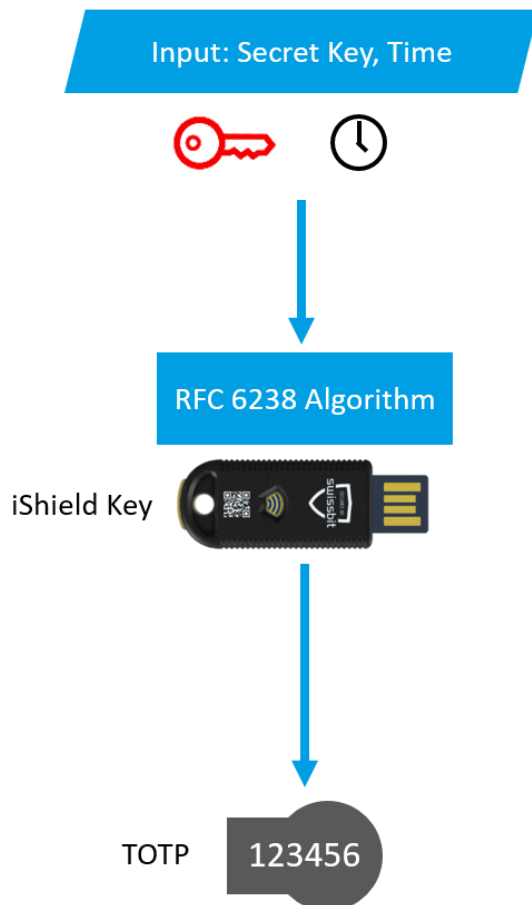
See section 3.1.4 for instruction on how to configure a new slot with the iShield Key Manager or use the command line tool as follow:

```
iKMcli totp --conf-slot <slot index> --key <key>
[--key-format <base32|hex>]
[--hmac <SHA1|SHA256|SHA512>]
[--otp-length <6|7|8|9>]
[--period <time period / interval>]
[--pin-protected]
```

The service usually asks you to generate a one-time password to finish the registration of your security key as second factor. Select your newly configured slot on your iShield Key Manager dashboard and click on the icon for code generation or use the iKMcli command:

```
iKMcli totp --gen-totp <slot index>
```

5.1.2 TOTP Computation

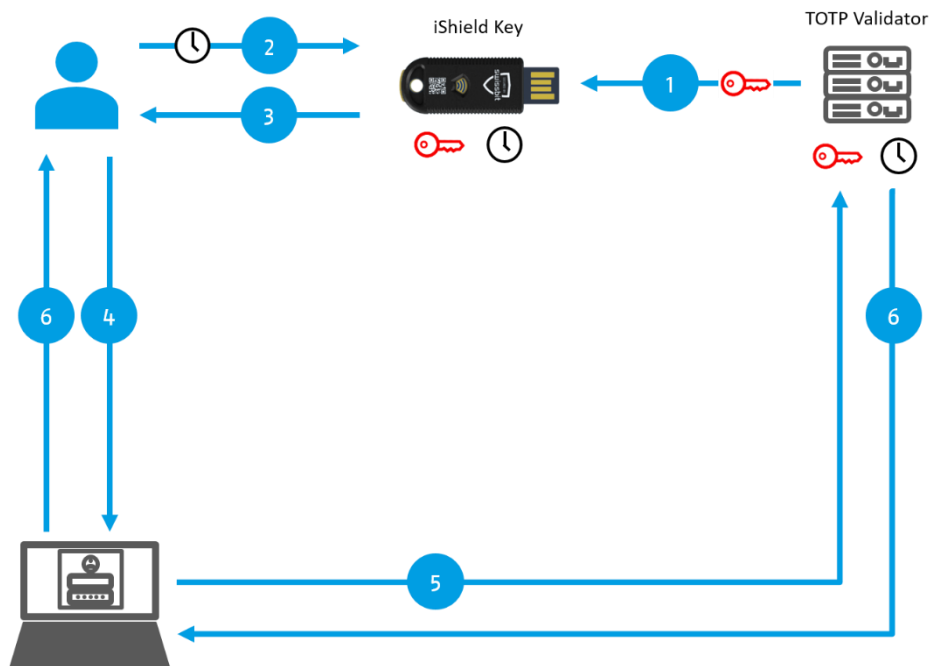


TOTP Computation

Given a secret key and the current time as input values, the iShield Key computes a new TOTP using the RFC 6238 algorithm

The iShield Key computes the time-based one-time passwords using the RFC 6238 algorithm. Given a secret key and the current time as input values, the iShield Key computes a six to nine digit passcode. The time is thereby the moving factor. Within a time interval, the computation result for the TOTP will be the same.

5.1.3 Password Generation and Authentication



TOTP Generation and Authentication

1. iShield Key is registered
2. User requests new TOTP for current time
3. New TOTP is generated
4. User enters login data with TOTP
5. Verify TOTP with Server
6. Authentication granted

After successful pairing of your iShield Key and service, your security key and service will compute the same series of passwords for the configured slot accordingly. The iShield Key generates a new password on request for the provided current time. When the user performs a login into the service and submits the generated password, the service also generates a password for the current time and compares it to the one entered by the user. If both passwords match, access is granted. The step 3 is described in more detail in the previous section.

Since the iShield Key and validation server are not given the same timestamp as input for the TOTP computation, it can happen that the results fall into different intervals. Typically, the service uses the time it receives a password for validation as input. So when a code is submitted when it is already about to expire or when a short period is configured, the validation is more likely to fail. Further, a clock drift between client and server side can lead to a mismatch. Usually, the TOTP validator computes a number of codes and passwords for a certain number of forward and backward time intervals are within a tolerance range.

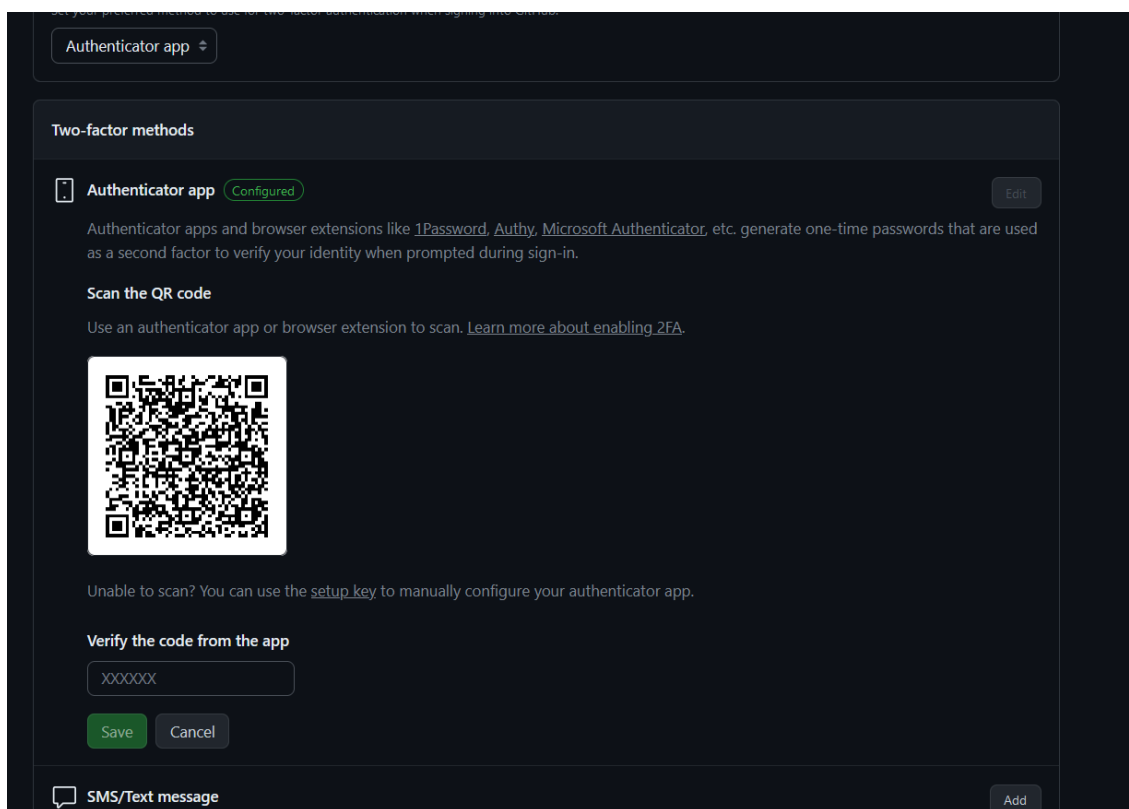
5.2 Swissbit iShield Key on various services

You can use the iShield Key TOTP function as second factor for a variety of services as many services support the Time-based One-Time Password Algorithm method. Typically, you can find the option to setup an "Authenticator Application" as two-factor method in your profile settings. For instance, you can use your iShield Key with AWS, Github, Bitbucket, Microsoft online accounts and identity providers such as Keycloak and Auth0.

In section 5.2.1, you will be guided through the setup with Github.

5.2.1 Github

Log into your account, go to the settings and navigate to "Access" – "Password and authentication". Choose "Authenticator app" from "Two-factor methods" and configure a new TOTP slot with the configuration provided for your account in the form of a QR code or setup key. Use the iShield Key Manager or command line tool to configure a slot as described in the previous sections.



To finish the setup, generate a new one-time password with the iShield Key tools; enter it on the Github website and click "Save". Now you can login with the TOTP function of your iShield Key as a second factor.

6 HOTP Applications

6.1 Overview and Functionality

The iShield Key Pro also offers one HMAC-based One Time Password slot. We recommend using HOTP for two-factor authentication on a service that does not support WebAuthn compliant FIDO2 security keys. The iShield Key Pro implements touch-triggered HOTP generation with the [RFC 4226](#) algorithm by IETF. Network applications such as VPN access often use this standard algorithm. The HOTP functionality of the iShield Key Pro can be used without any installation effort. It is as simple as plug and play.

In section 6.1.1, the registration of your iShield Key Pro with a service is explained, section 6.1.2 provides details on the computation of a HOTP, section 6.1.3 describes the process of password generation and authentication and section 6.1.4 illustrates the resynchronization of the hardware authenticator and server.

Please note that this section is not relevant for the iShield Key FIDO2.

6.1.1 Registration

For symmetric generation of one-time passwords of the iShield Key Pro and service for which you want to register a second factor, the secret key and initial counter value is shared. The service provides the secret key and initial counter value and you can use the iShield Key Manager tools to configure your iShield Key Pro HOTP applet accordingly. See section 3.1.4 for instructions on how to configure your iShield Key with the iShield Key Manager or use the iShield Key Manager command line tool as follows:

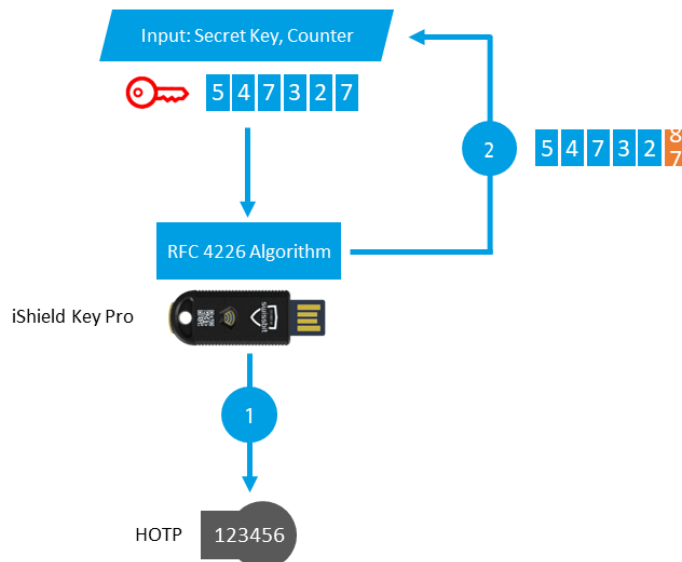
```
iKMcli hotp --set-key <key> --pin <pin>
iKMcli hotp --set-counter <counter> --pin <pin>
```

Moreover, you can choose if you want to generate passwords of six or eight digits:

```
iKMcli hotp --set-otp-length <length>
```

See section 3.2.2 for usage instructions for the iKMcli and all supported HOTP operations. We recommend changing the factory default PIN before registering your iShield Key Pro with a service. The default PIN is 1234. **Please note that the PIN cannot be unblocked once you entered it incorrectly 10 times in a row.**

6.1.2 HOTP Computation

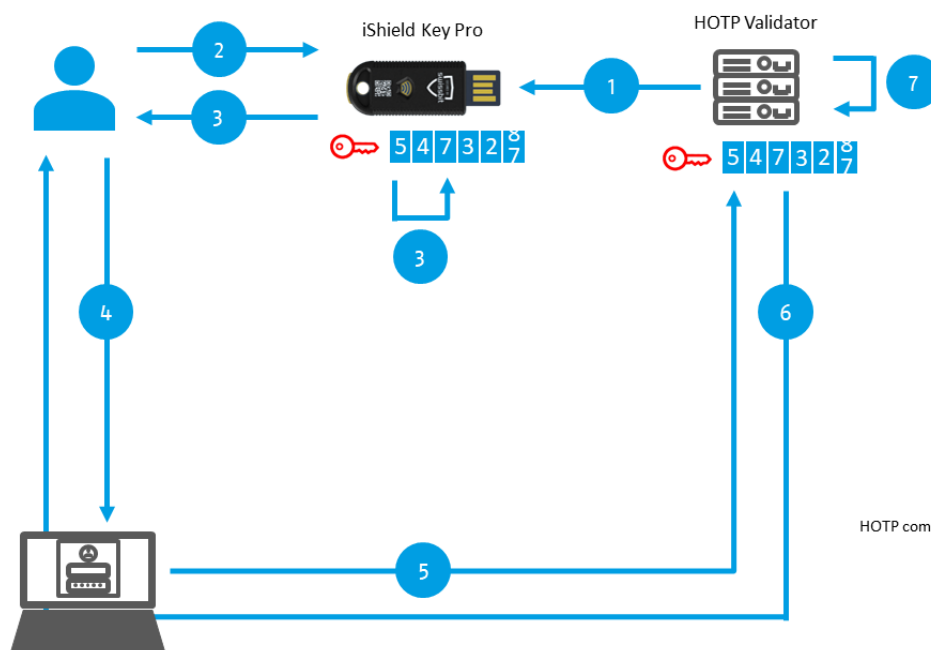


HOTP Computation

1. Given a secret key and counter value as input values, the iShield Key Pro computes a new HOTP using the HMAC based RFC 4226 algorithm
2. iShield Key Pro counter is incremented – the next HOTP will be computed given this new input value for the counter

The iShield Key Pro implements secret key and counter-based HOTP generation using the HMAC based RFC 4226 algorithm. Given a secret key and counter value as input values, the iShield Key Pro computes a six or eight digit human-readable password. The counter is a moving factor. After computation of a new HOTP, the counter is incremented and the next HOTP is computed based on the incremented counter. Therefore, each counter value enters only once the computation of a password.

6.1.3 Password Generation and Authentication



HOTP Generation and Authentication

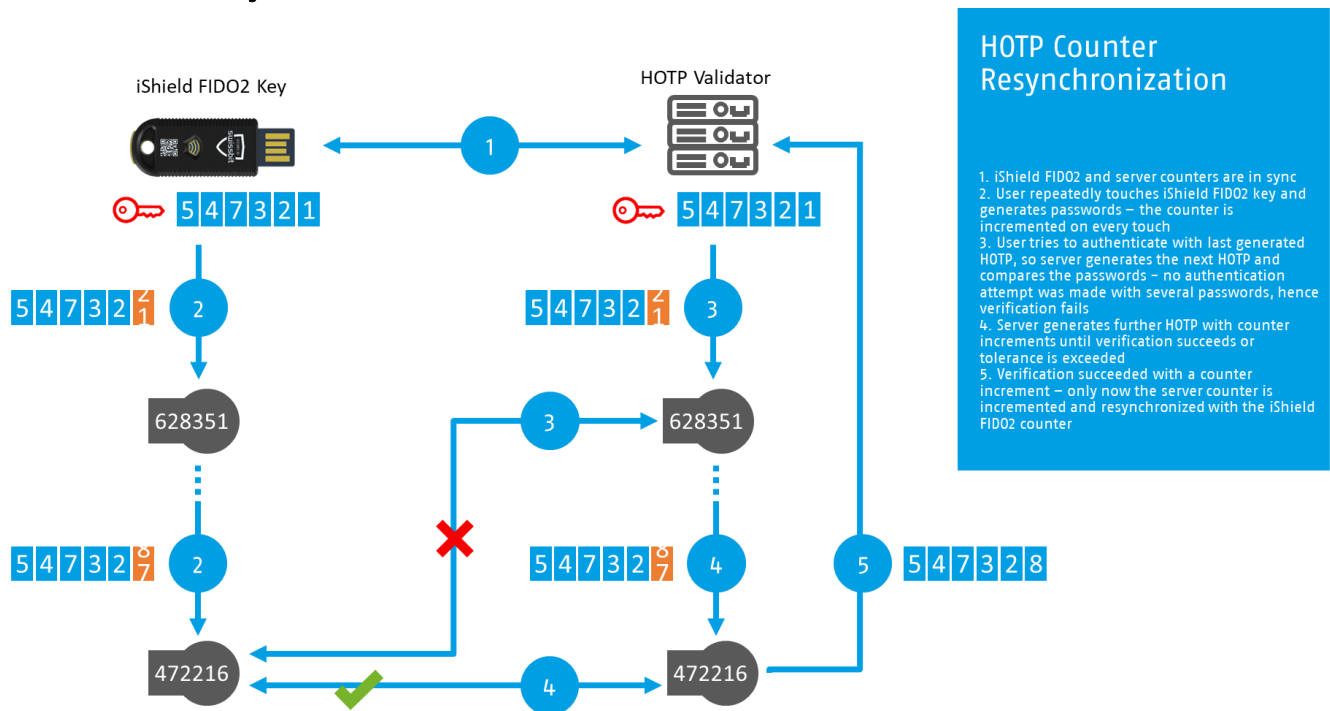
1. iShield Key Pro security key is registered
2. User touches iShield Key Pro
3. New HOTP is generated and iShield Key Pro counter is incremented
4. User enters login data with HOTP
5. Verify HOTP with Server
6. Authentication granted
7. Server counter is incremented

HOTP computation (3.) is illustrated in more detail on previous slide

After you registered your iShield Key Pro with a service that uses HOTP, no connection is required to generate coinciding series of passwords. This is due to the deterministic nature of the HOTP algorithm and the shared secret key and counter between token and server. The token generates a new password on touch and increments the counter. When the user now authenticates to the service, the password is compared to the password generated by the server. On successful authentication, also the counter of the server is incremented.

The step 3 of the HOTP computation is illustrated in more detail in the previous section.

6.1.4 Counter Resynchronization



It can happen that the token and server counter lose synchronization. For instance, if the user touches the token, generates a new password but does not authenticate to the associated service with it. In order to avoid unsuccessful authentication attempts, configure the look-ahead parameter on the server. This parameter defines how many counter increments are considered for password comparison, so some sort of tolerance is allowed. Successful authentication resynchronizes the counters of the generator and server.

7 PIV Applications

In this part of the guide, you will learn how to use your Swissbit iShield Key Pro as a personal identification and verification (PIV) device on Windows. The iShield Key Pro with PIV applet provides different slots to store and provide various certificates for different use-cases. This guide uses the OpenSC Minidriver complemented with the Swissbit iShield PIV module to provision the smartcard with these certificates. We will show how to generate and use these in the following sections.

Please note that this section is not relevant for the iShield Key FID02.

7.1 Overview Use Cases

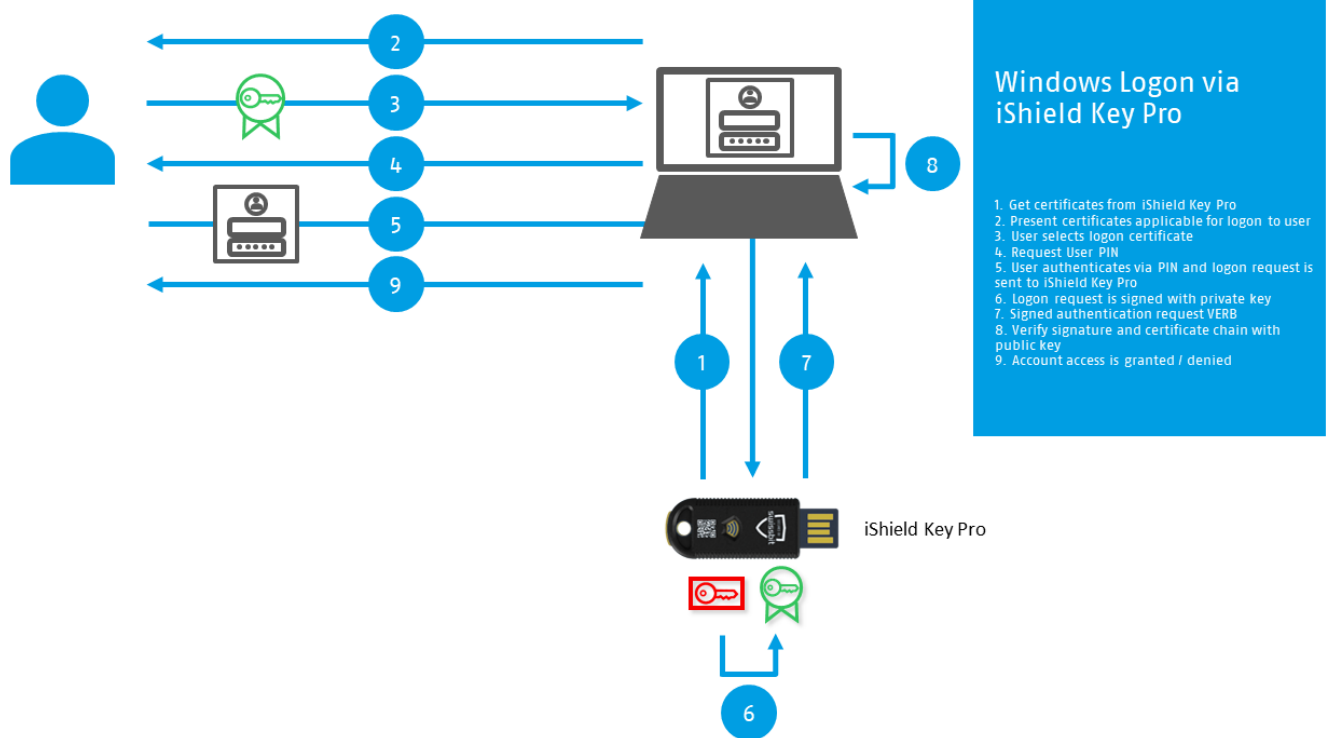
	Local Windows Account	Active Directory Domain Account
Windows Logon	Use iShield Key Pro to logon to a local windows user account.	Use iShield Key Pro to logon to any Workstation in a Domain setup. The certificate on the smartcard stores account information. A user only has to connect it to the PC and provide their smartcard PIN.
Bitlocker	Encrypt data drives with self-signed certificates.	Encrypt data drives with domain CA issued certificates.

For these PIV use-cases, we need to differentiate between two configurations of the Windows environment:

- **Local Account:** Where one or multiple accounts reside in one PC. They do not belong to a domain, where a central organization would manage them. This configuration typically observed in home PCs.
- **Active Directory Domain Accounts:** A central domain server manages machines, users, printers and others within this domain network. This configuration is useful for large organizations like companies to manage employee accounts/machines.

Both configurations are able to host all PIV use-cases, albeit with some differences regarding setup or usage. This guide will in general target information regarding the local account scenario first and continue with the setup instructions for active directory. Furthermore, it is possible to have one PC with local and active directory domain accounts on it. This might result in some incompatibility between the different use-cases. For more information, have a look at the respective instructions in this guide.

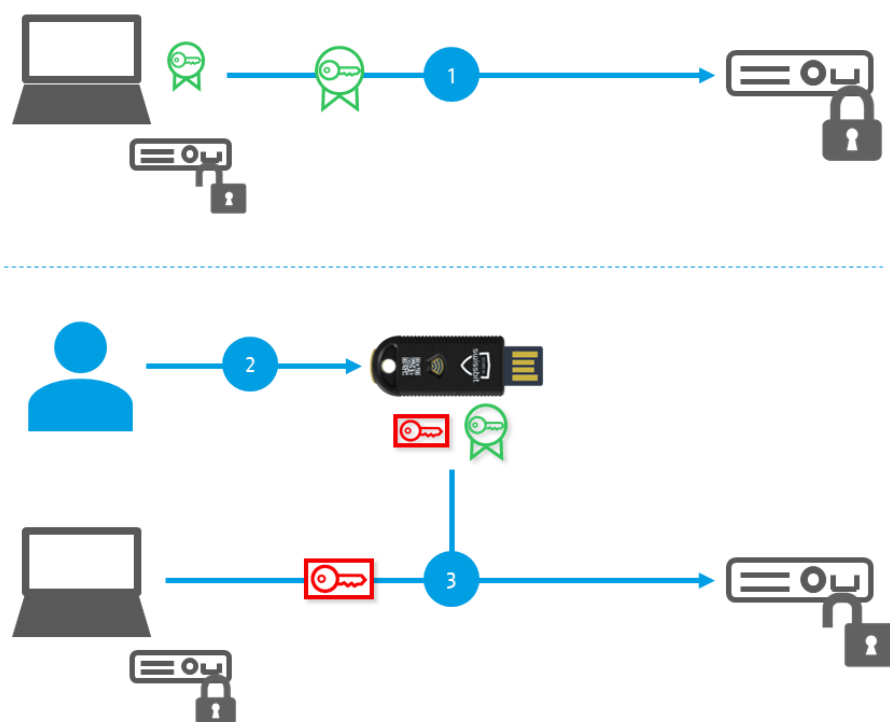
7.1.1 Logon



With the iShield Pro Key you implement a more secure logon based on a PKI hardware token, instead of a password. Your windows user account is configured to trust the certificate on your smartcard. Moreover, you only need to plug in your iShield Pro Key and provide a short PIN, which not only is more secure than passwords but also more convenient.

For exact instructions on how to setup and use the iShield Pro Key for logon purposes within a domain, have a look at section 7.7.

7.1.2 Bitlocker



Drive Encryption with Bitlocker

Encryption:
1. Encrypt drive with public key from iShield Key Pro – Drive is locked after being unplugged

Decryption:
2. User authenticates via PIN
3. Decrypt drive with private key stored on iShield Key Pro – Drive is unlocked until it is unplugged

Bitlocker is a Microsoft tool that is used to encrypt and decrypt data drives. This targets drives, that are installed in a PC internally or external drives, which typically connect to various different machines. Whenever you are storing data on the drive, Bitlocker encrypts it with a pre-defined certificate. As soon as a Bitlocker session terminates (for example by unplugging the external USB drive or powering down a PC) the data is encrypted and not readable anymore. To get access to the device again, you will need to provide the decryption information. In the most common case, they exist in the form of a PIN or key, which you will need to enter when you insert the drive or start your PC.

For higher security, you can use the iShield Key Pro to store the required certificates and decryption information. Whenever a user wants to access data on said medium, they need to insert the smartcard with the corresponding encryption certificate and provide its PIN. This allows for higher security of the data: To access them, you will need to provide the iShield Key Pro with the certificates and enter its PIN.

Note that the Windows **OS (C:)** partition is not a data drive. Bitlocker is not set up to use a smartcard to encrypt it but has other limited configurations for this.

For exact instructions on how to setup and use the iShield Key Pro for Bitlocker within a domain, have a look at section 7.6. If you want to use it on a local account, continue in section 7.5.

7.1.3 Active Directory

Microsoft sells components for Active Directories (AD), which is a domain infrastructure for which we will explain setup and compatibility in parts of the remainder of this PIV guide. This Active Directory Domain Structure consists of several components:

- **Domain Server(s):** Management utilities, to configure all components within this structure. Among many others, this include users, workstations, server components and (important for PIV) certificates.
- **Workstations:** Multiple Client PCs on which users can logon to their accounts. These receive their configuration by the Domain Server(s).

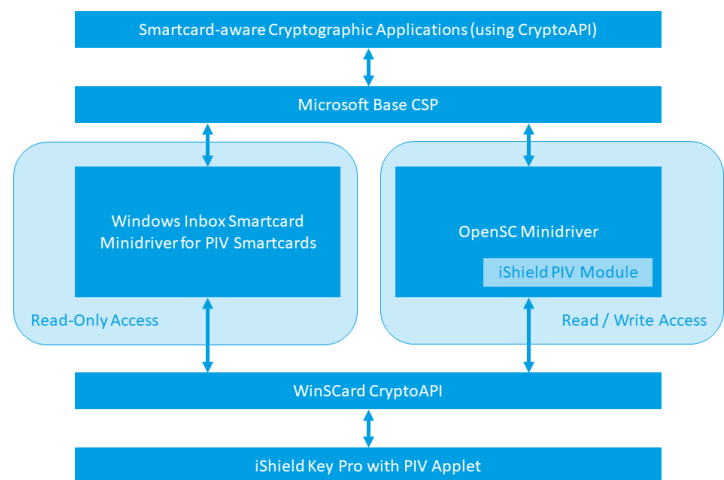
For AD use-cases, this guide assumes that you have already setup the required components.

7.2 Underlying Components

7.2.1 Token Provisioning and Usage on Windows

The *Windows Inbox smartcard minidriver* supports PIV smartcards but only read access is possible. The minidriver is sufficient for usage of PIV functionality of the iShield Key Pro but card administration is not possible. Once your PIV card is provisioned for your use cases i.e. all required certificates are generated on the smartcard, you can use the Windows PIV driver to sign or decrypt with your iShield Key Pro.

For write access, that is to say for provisioning your card, the installation of the *OpenSC Minidriver* and configuration to use the *Swissbit iShield PIV Module* for card administration is required.



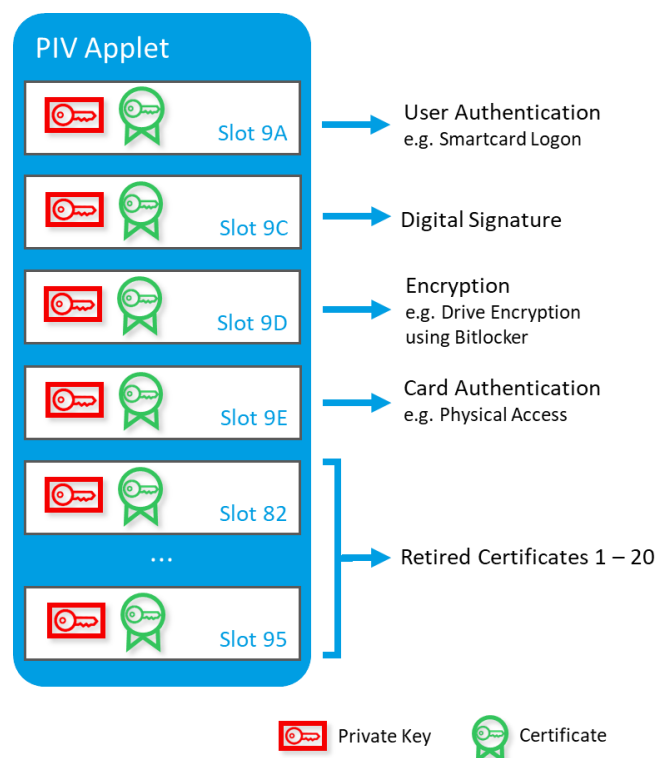
7.2.2 Authentication

The iShield Key Pro has three passwords that are required for management operations and usage. The PIN is required for normal usage of the token, i.e. for private key operation such as signing and decryption. The PIN is of course also required for changing the PIN and for setting the PIN retries. The PUK can be used to unblock the PIN. The management key is necessary to authenticate for performing administrative operations. Examples of operations for which you need to authenticate as administrator are key import or generate deleting or importing certificates or generally writing PIV data to the card.

7.2.3 Certificate Slots

The PIV applet stores certificates and the corresponding public private key pairs in slots – one slot can hold one certificate. The iShield Key Pro holds four standard PIV slots 9A, 9C, 9D and 9E and 20 further retired slots 82 – 95. Each standard PIV slot has an intended purpose, the retired certificate slots are typically used for expired certificates but can be used for signing or decryption like the certificates in any other slots.

The certificate in slot 9A is usually used for user authentication. You can use this slot for your certificate for smartcard Windows logon; see section 7.1.1 for the use case. The certificate in slot 9C is for digital signature e.g. for email signing. The private key of the slot is according to the standard generated with a special PIN policy. As per the standard, the PIN is required for each private key operation. The certificate in slot 9D is used for encryption for confidentiality. A use case, for which you would want to use this slot, is drive encryption using Bitlocker; see section 6.5. Slot 9E again has a special PIN policy. The PIN is never required for private key operations, so the certificate in this slot is used for card authentication. A typical use case is access to building.



When provisioning the iShield Key Pro using the OpenSC Minidriver, supplemented with the iShield PIV Module for PIV administration, the certificates are generated in the next free slot with standard PIN policy: The first certificate is stored in slot 9A, the next one in slot 9D, followed by the retired slots in order.

7.3 Requirements

For now, Swissbit has tested the following systems and applications with Swissbit iShield Key Pro for PIV.

- PC Operating System: **Windows 10 Pro**; Home editions do not ship with Bitlocker or Domain Account support. The Instructions should be similar in Windows 7/8/11.
- Server Infrastructure: **Windows Server 2019**; for setup with Active Directory. Setup instructions might work differently for other Server versions.

7.4 Getting started with PIV on iShield Key Pro

Please download the installation package from the Swissbit iShield Key Pro landing page.

7.4.1 PIV Installation Package

In the installation package, you should find the following tools:

- |— UserManual.pdf This guide
- |— iKMcli.exe iShield Key Manager command line tool
- |— win64/ishield-piv-module
- | |— bin
- | | |— ishield_piv_module.dll PIV module
- | | |— vcruntime140.dll, msvcrt140.dll, etc. Required system runtime libraries
- | |— PIV-II.profile OpenSC PIV profile
- |— win32/ishield-piv-module
- | |— bin
- | | |— ishield_piv_module.dll PIV module
- | | |— vcruntime140.dll, msvcrt140.dll, etc. Required system runtime libraries
- | |— PIV-II.profile OpenSC PIV profile

7.4.2 Installation of the OpenSC Minidriver and iShield PIV Module

Within the following setup instructions, Windows requires that you install a vendor specific minidriver for some actions. For the iShield Key Pro, this comes in the form of the OpenSC Minidriver, which is extended by the Swissbit iShield PIV Module.

This step is always required for operations, where Windows is writing certificates to the smart card. In case all necessary certificates are already present on the iShield Key Pro, the pre-installed default Windows Minidriver is sufficient. All setup instructions will hint to whether this vendor specific minidriver is required.

The OpenSC Minidriver in combination with the iShield PIV module supports all use-cases in the same way, the in-built Windows Smartcard Minidriver does.

Install OpenSC

- From the Swissbit OpenSC GitHub¹ repository, download the latest release of OpenSC.
- Run the installer for your operating system. Make sure to select "Complete" installation
- Copy the **.profile** file to the *profiles/* directory in your OpenSC installation folder (Unless you have changed it, by default the OpenSC installation folder is in *C:/Program Files/OpenSC Project/OpenSC*)

Create Management Key configuration

OpenSC expects a text file containing the management key in the following format:

```
XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX
```

e.g. *01:02:03:04:05:06:07:08:01:02:03:04:05:06:07:08:01:02:03:04:05:06:07:08* for the default management key. An environment variable `PIV_EXT_AUTH_KEY` must point to this text file. Then the file can be stored anywhere but we would recommend placing it in your user directory. Do not forget to update this file, whenever you are assigning a new management key.

Configure Environment Variables:

Search for "Environment Variables" in the start menu and click *Edit Environment Variables*

- Add the environment variable "PIV_EXT_AUTH_KEY" pointing to the file with the management key.
- Edit the environment variable "PATH", add the path to the OpenSC *tools* directory to the entries

Include iShield PIV Module

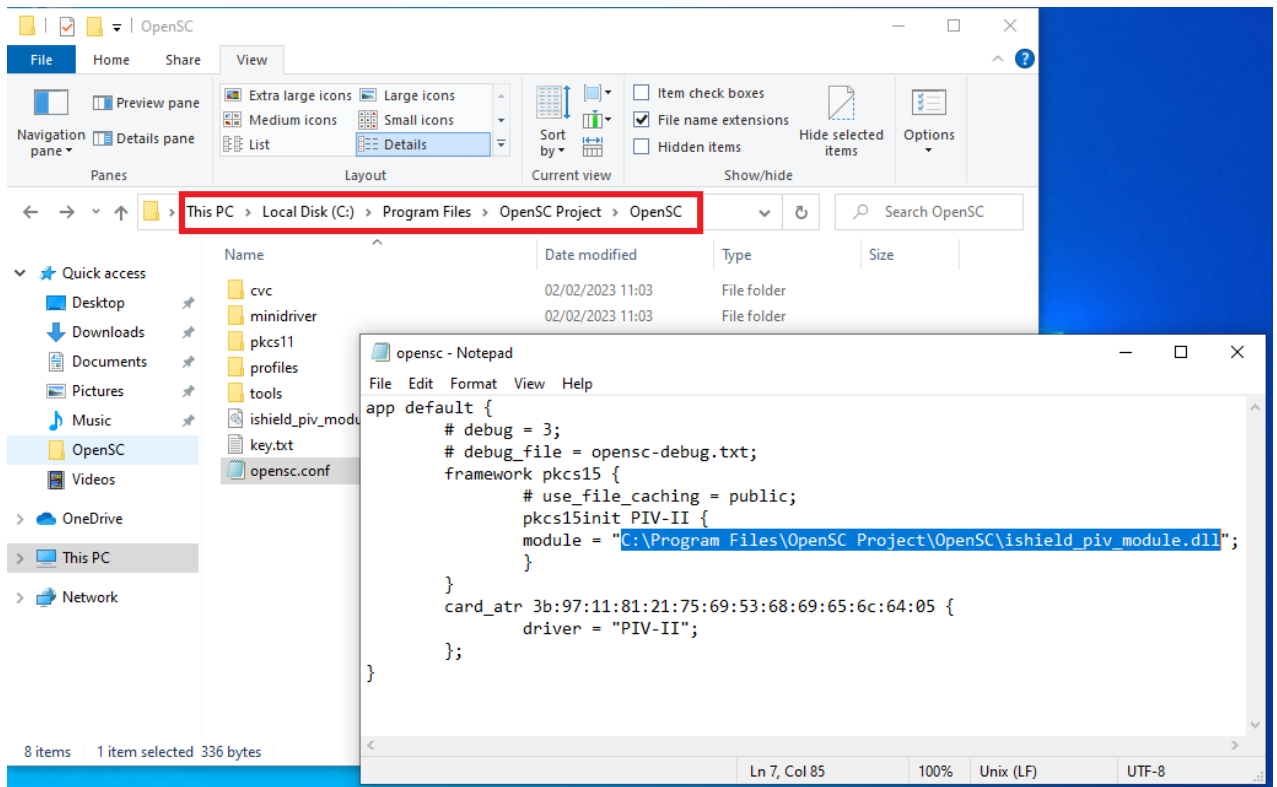
Adjust the OpenSC configuration to use an external PKCS#15 Init module for PIV:

- Copy the iShield PIV module *ishield_piv_module.dll* file to this PC (e.g. into the OpenSC installation folder)
- Make sure you have installed the "Microsoft Visual C++ Redistributable for Visual Studio 2015" or newer from the official Microsoft webpage² or place the required system runtime libraries from the installation package next to the iShield PIV module
- Modify the configuration file *opensc.conf* to include the module by adding the following information. Make sure to **replace the module path** with the location to where you have copied the .dll-file. The ATR of the iShield Key via USB is always the same and can be left unchanged. If you want to use your iShield Key via NFC, add an entry for the contactless ATR *3b:87:80:01:69:53:68:69:65:6c:64:50* to the OpenSC configuration file. Please note that the three first PIV slots 9A, 9C and 9D cannot be accessed through a contactless interface like NFC as specified in the PIV standard.

¹ <https://github.com/swissbit-eis/OpenSC/tags>

² <https://visualstudio.microsoft.com/downloads/#microsoft-visual-c-redistributable-for-visual-studio-2022>

```
app default {
    ...
    framework pkcs15 {
        pkcs15init PIV-II {
            module = "<ishield_piv_module_path>";
        }
    }
    card_atr 3b:97:11:81:21:75:69:53:68:69:65:6c:64:05 {
        driver = "PIV-II";
    };
    ...
}
```



Restart your PC now!

7.4.3 Preparation of the iShield Key Pro

It is highly recommended to change the PIN, PUK and management key before using the iShield Key Pro. For this purpose, you can use the iShield Key Manager; see Section 3.1.6, or the command line tool iKMcli. Please plug in your iShield Key Pro now and execute the following commands:

```
iKMcli piv --change-pin <new pin> --pin <pin>
iKMcli piv --change-puk <new puk> --puk <puk>
iKMcli piv --set-management-key <new key> --management-key <key>
```

Remember to adjust the management key also in your key file from 4.1.2.

The factory default for the PIN is 123456, the default PUK is 12345678 and the management key is 010203040506070801020304050607080102030405060708.

Generate a new card holder unique identifier and card capability container by

```
iKMcli piv --set-chuid --management-key <key>
iKMcli piv --set-ccc --management-key <key>
```

7.4.4 Reset the iShield Key Pro

If you want to reset the PIV applet on your iShield Key Pro, that is to say erase all PIV data and restore the default settings, use the iShield Key Manager or follow these steps:

Block your PIN and PUK by authenticating with wrong passwords, e.g.

```
iKMcli piv --change-pin 123456 --pin 111111
iKMcli piv --change-puk 12345678 --puk 11111111
```

Once you blocked your PIN and PUK, you can reset the iShield Key Pro by

```
iKMcli piv --reset
```

Set a new card holder unique identifier and card capability container by

```
iKMcli piv --set-chuid --management-key <key>
iKMcli piv --set-ccc --management-key <key>
```

Prepare your iShield Key Pro for usage and set a new PIN, PUK and management key as in section 7.4.3

7.5 Use Case: Local Account Bitlocker

In this scenario on the **local account**, the Bitlocker certificate is self-signed on the local PC, which requires a different setup procedure, than in other Bitlocker scenarios. This guide targets private customers that want to use the Bitlocker functionality on their home devices.

7.5.1 Setup Process

To setup Bitlocker with iShield Key Pro you will need to write a certificate to the iShield Key Pro. We will use the Microsoft in-built EFS certificate utility, which handles certificate generation and storage on the smartcard automatically. You will also need to modify the Bitlocker configuration to accept these certificates.

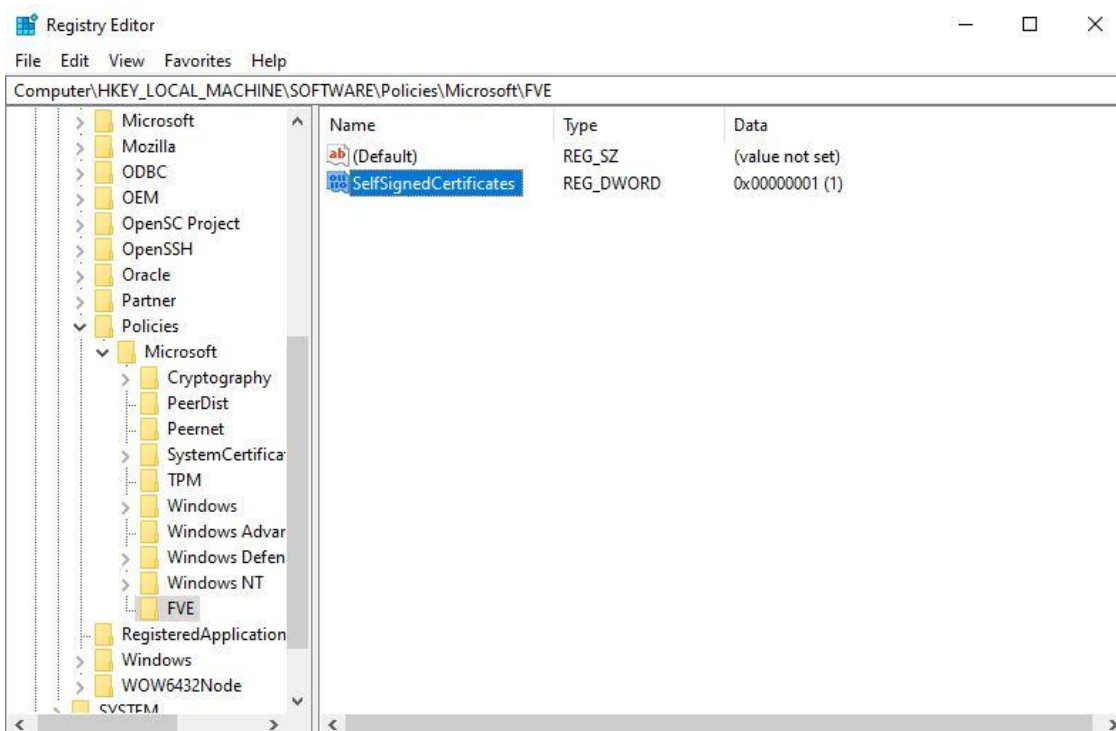
This requires you to install the **OpenSC Minidriver** and the **iShield PIV Module** as described in section 7.4.2

Allow self-signed certificates on Windows PC

This step is necessary for development purposes. It configures windows to allow using self-signed certificates for Bitlocker. You can generate such certificates yourself and do not need to request them from an official certificate authority. If possible, we recommend official certificates for high security purposes, but this guide will not go into details on them.

Open the Registry editor by searching for **regedit** in Windows Home menu.

- Locate `Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE` (If any of these directories does not exist do a right-click, New, Key)
- Create (if not existent) the Key `SelfSignedCertificates` (DWORD 32 bit) and set its value to `1`.

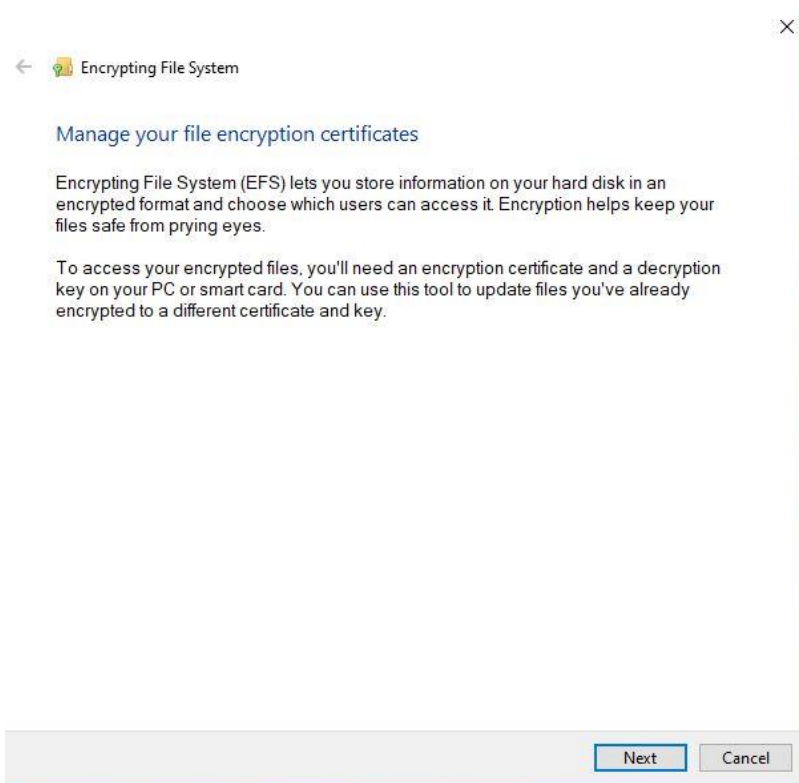


Generate EFS Certificate

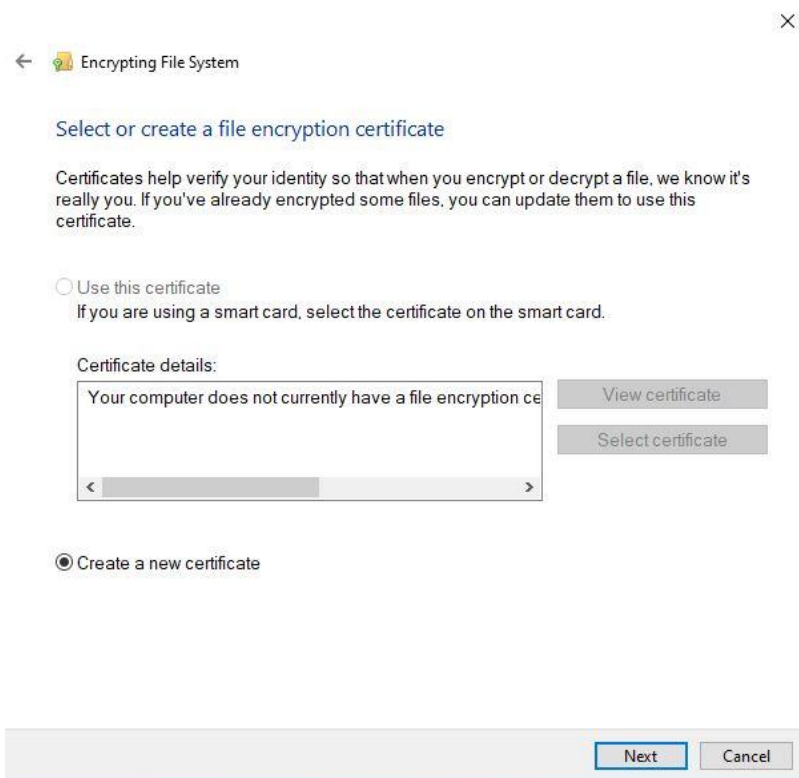
Within this step, you will use a windows tool to generate an EFS-certificate on the iShield Key Pro. If such a certificate (or any Bitlocker compatible certificate) is already present, you can skip this step.

Launch a **control panel**:

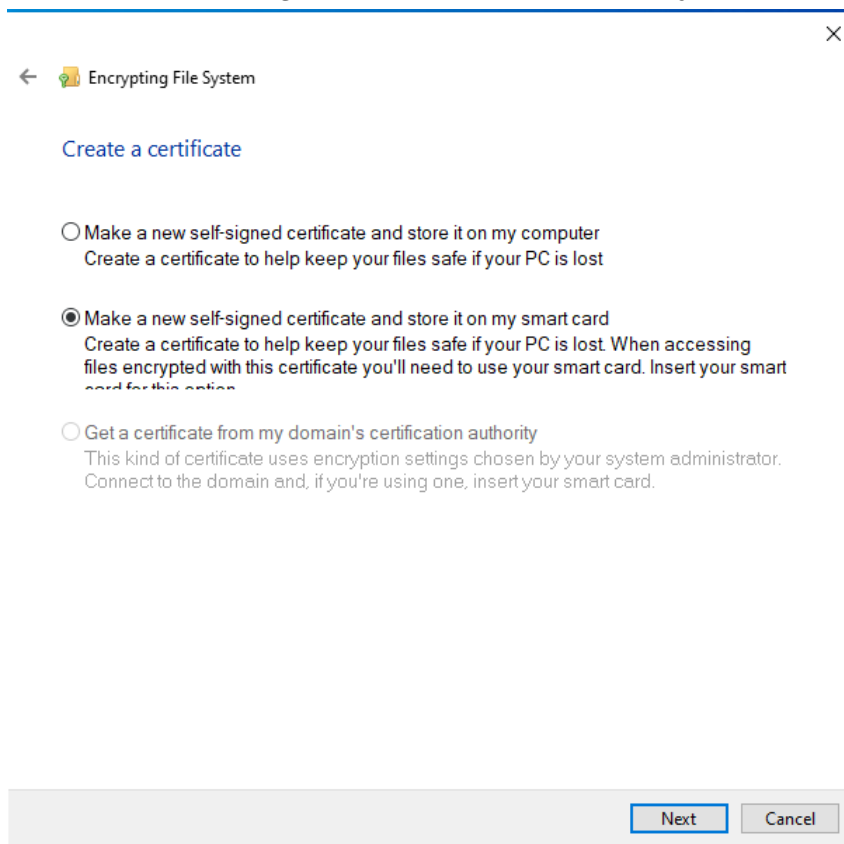
- Search for "certificates" and click *Manage file encryption certificates*



- Click through the wizard: *Next* > " Create a new certificate" > *Next*



- "[X] Make a new self-signed certificate and store it on my smart card" > *Next*



- Provide the PIN for your iShield Key Pro
- Click **Cancel** (You do not need to finish the procedure, since by now, the utility has already written the certificate to the smartcard).

Allow EFS Certificates for Bitlocker

In this section, you will configure your local Bitlocker setup to accept certificates with the OID, which was originally assigned to the Microsoft File Encryption Certificates.

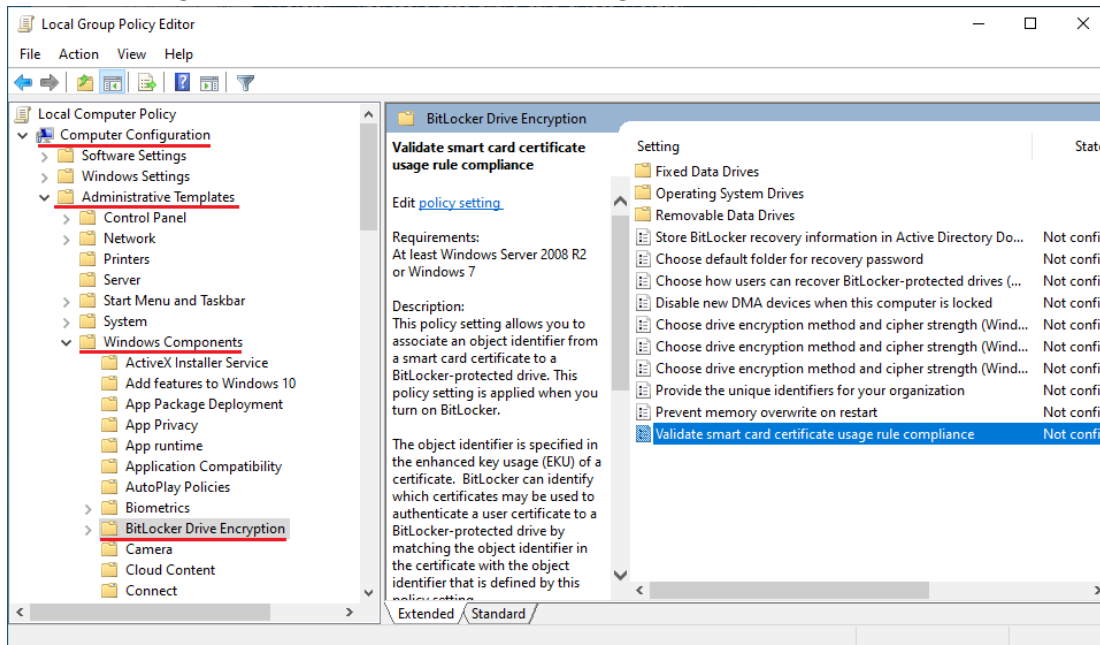
In a **PowerShell**, execute `certutil -scinfo`:

- You will be prompted for your smartcards PIN multiple times
- After execution you will find `Application[0] = ...` in the output. Copy the *OID* after the '=' sign

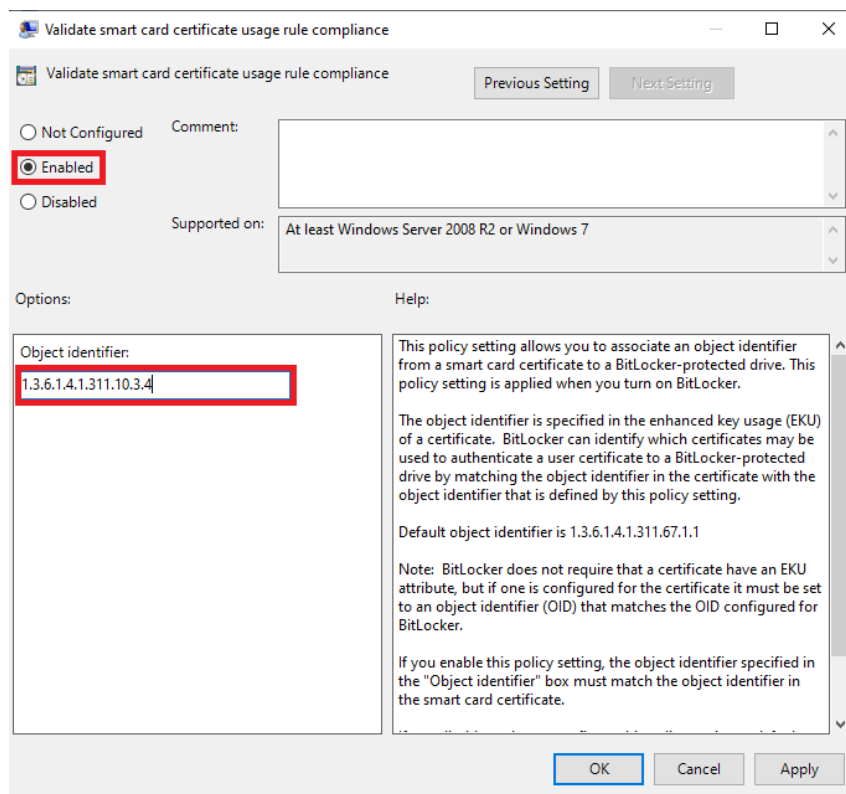
```
CertContext[0][0]: dwInfoStatus=10c dwErrorStatus=20
Issuer: CN=Swissbit
NotBefore: 03/02/2023 17:32
NotAfter: 10/01/2123 17:32
Subject: CN=Swissbit
Serial: 3dccd83464b8229b41a
SubjectAltName: Other Name:Principal Name=Swissbit@DESKTOP-12DRKV4
Cert: e5b99b40d99d362f8befe3eec5b
Element.dwInfoStatus = CERT_TRUST_HAS_NAME_MATCH_ISSUER (0x4)
Element.dwInfoStatus = CERT_TRUST_IS_SELF_SIGNED (0x8)
Element.dwInfoStatus = CERT_TRUST_HAS_PREFERRED_ISSUER (0x100)
Element.dwErrorStatus = CERT_TRUST_IS_UNTRUSTED_ROOT (0x20)
Application[0] = 1.3.6.1.4.1.311.10.3.4 Encrypting File System
```

In the Windows Home Menu search for **gpedit** and open the "Group policy editor":

- Navigate to *Local Computer Policy > Computer Configuration > Administrative Templates > Windows Components > Bitlocker Drive Encryption*
- Edit the setting "Validate smart card certificate usage rule compliance"



- Check *Enabled*
- Paste the previously copied OID underneath *Object identifier*
- Then click *OK*



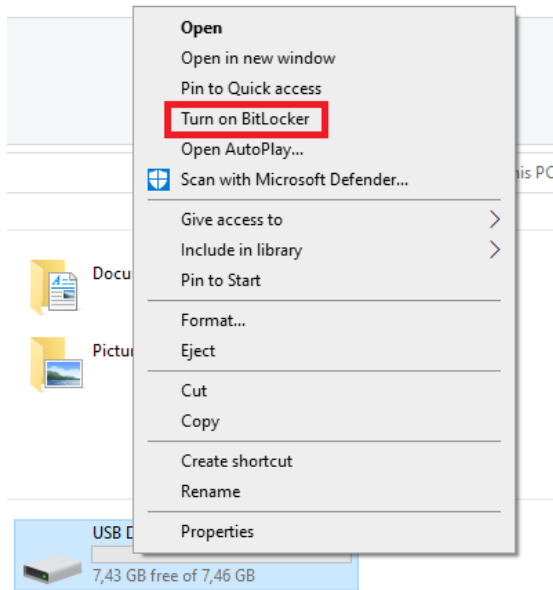
Enable Bitlocker for one storage medium

After this step, you have prepared an external flash drive or (internal) data-disk for Bitlocker. All the present data will be encrypted retroactively and all future data stored there will be encrypted. Note that this does not work with boot drives, but only with data disks.

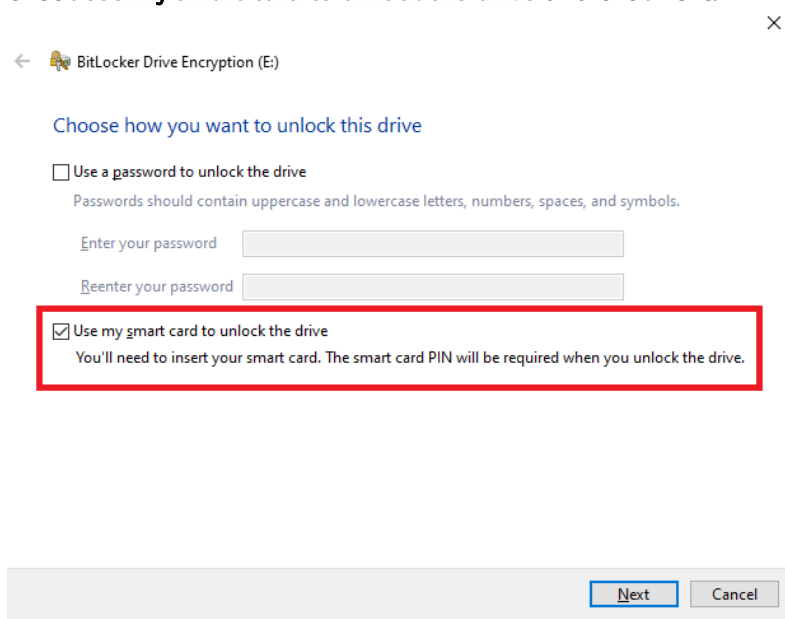
Note: For this step, you do not need to install the OpenSC Minidriver and iShield PIV module anymore. As soon as there is a valid certificate on the iShield Key Pro, the pre-installed Windows tools work by themselves.

Insert the device you want to encrypt in your PC and open it in the **Windows Explorer**.

- Right-click on the drive and select **Turn on BitLocker**.

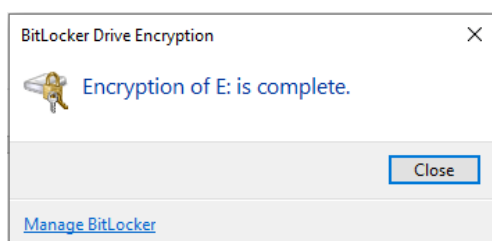


- Check **Use my smart card to unlock the drive** and click *Next*.



- Store your recovery key in some secure place. In case you were to lose your iShield Key Pro in the future, you can still recover your encrypted drive. Click *Next*.
- For the next options, you can choose whichever you want. Finally click **Start encrypting**. Depending on the options you have chosen, this might take a while.

The drive and smartcard have now been prepared to work with BitLocker on any Windows 10 Pro PC.



7.5.2 Use it to encrypt a Drive

As soon as the iShield Key Pro and external drive are prepared, the default Windows PIV driver is sufficient for usage with Bitlocker. You will not need to install the OpenSC Minidriver or iShield PIV module in this case.

- Insert your drive. You should see a message "This drive is BitLocker-protected".
- Access the drive via the Explorer. A prompt for your smart card (iShield Key Pro) or your recovery key will appear.
- Select the "smart card" option and provide your smart card PIN.
- You will find the drive unlocked and usable until you unplug and reinsert it

7.6 Use Case: Active Directory Bitlocker

In this scenario, the PC, on which Bitlocker is used, is a workstation within an **active directory domain**. The domain server will manage all the domain information. Upon request, the certificate authority on the domain server will issue and sign the used certificate. This is ideal for companies, in which the AD infrastructure already exists. You can increase your data security by following the next steps.

This guide expects a working AD setup.

7.6.1 Setup on Server

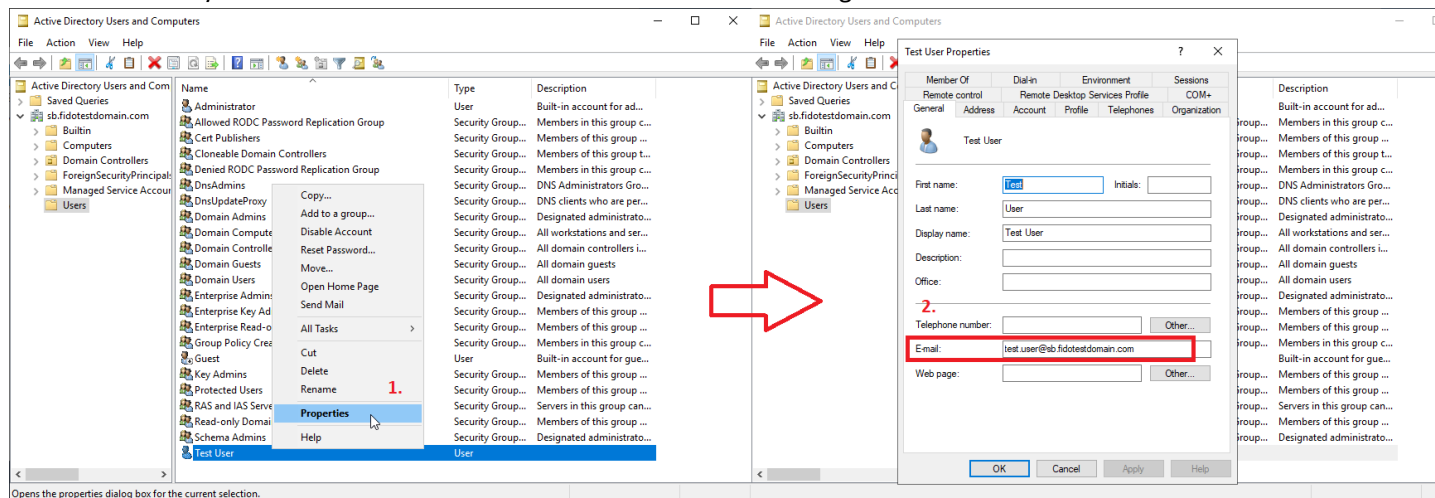
This section explains the necessary steps to configure a Microsoft Active Directory Domain Server for this use case. These instructions target the IT Administrator of a domain and you cannot conduct those on a personal computer or domain workstation.

Requirements beforehand

All users, that want to self-enroll certificates for Bitlocker, need an email address assigned to them in the user management tool.

In the Server Manager:

- Go to *Tools > Active Directory Users and Computers*
- In your domain go to *Users* and **right click the user**
- Click *Properties > General*: Make sure an email address is assigned to them



Setup Bitlocker Template

On the AD-ROOT-Server, open the **Server Manager**.

1. In the top right go to *Tools > Certification Authority*
2. In **certsrv** expand the server name
3. Right click *Certificate Templates > Manage*

Setup the Bitlocker Template:

4. In Certificate Templates Console right click Smartcard User > Duplicate Template and adjust the Properties of each of the tabs as follows:

○ General: Rename template

Properties of New Template

Subject Name Server Issuance Requirements
Superseded Templates Extensions Security
Compatibility General Request Handling Cryptography Key Attestation

Template display name:
Smartcard User Bitlocker

Template name:
SmartcardUserBitlocker

Validity period: 1 years Renewal period: 6 weeks

Publish certificate in Active Directory
 Do not automatically reenroll if a duplicate certificate exists in Active Directory

OK Cancel Apply Help

○ Request Handling: Select purpose Encryption

Properties of New Template

Subject Name Server Issuance Requirements
Superseded Templates Extensions Security
Compatibility General Request Handling Cryptography Key Attestation

Purpose: Encryption

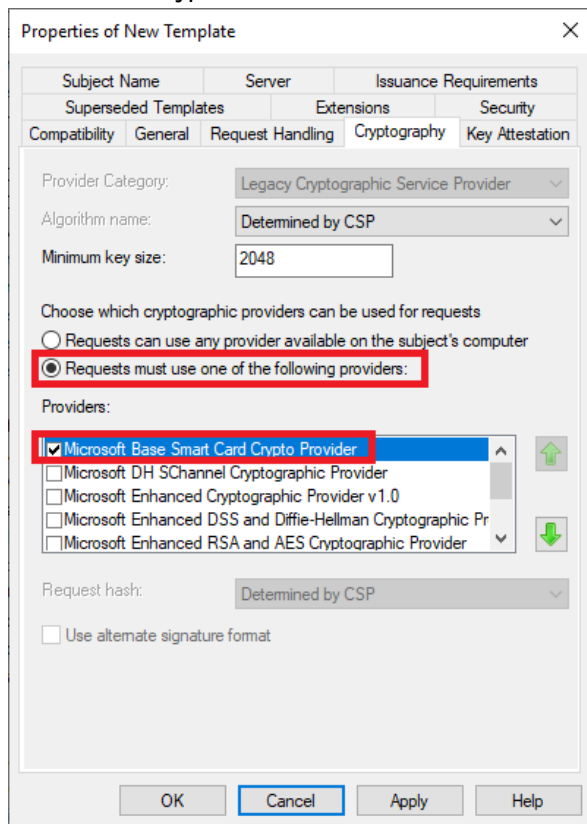
Allow private key to be exported
 Renew with the same key (*)
 For automatic renewal of smart card certificates, use the existing key if a new key cannot be created (*)

Do the following when the subject is enrolled and when the private key associated with this certificate is used:
 Enroll subject without requiring any user input
 Prompt the user during enrollment
 Prompt the user during enrollment and require user input when the private key is used

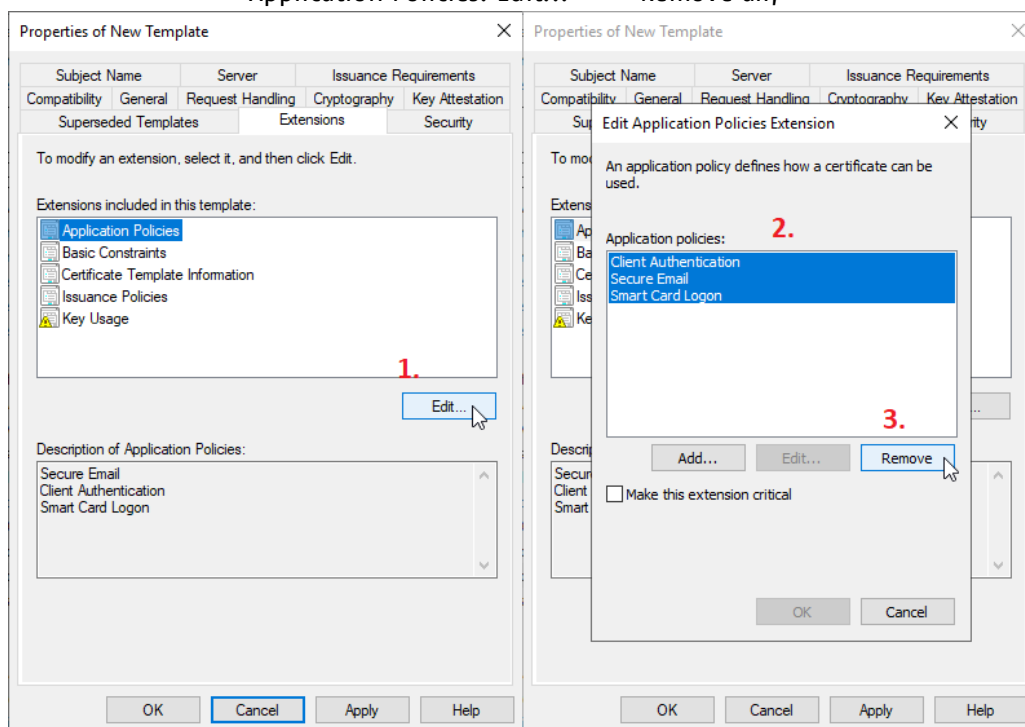
* Control is disabled due to [compatibility settings](#).

OK Cancel Apply Help

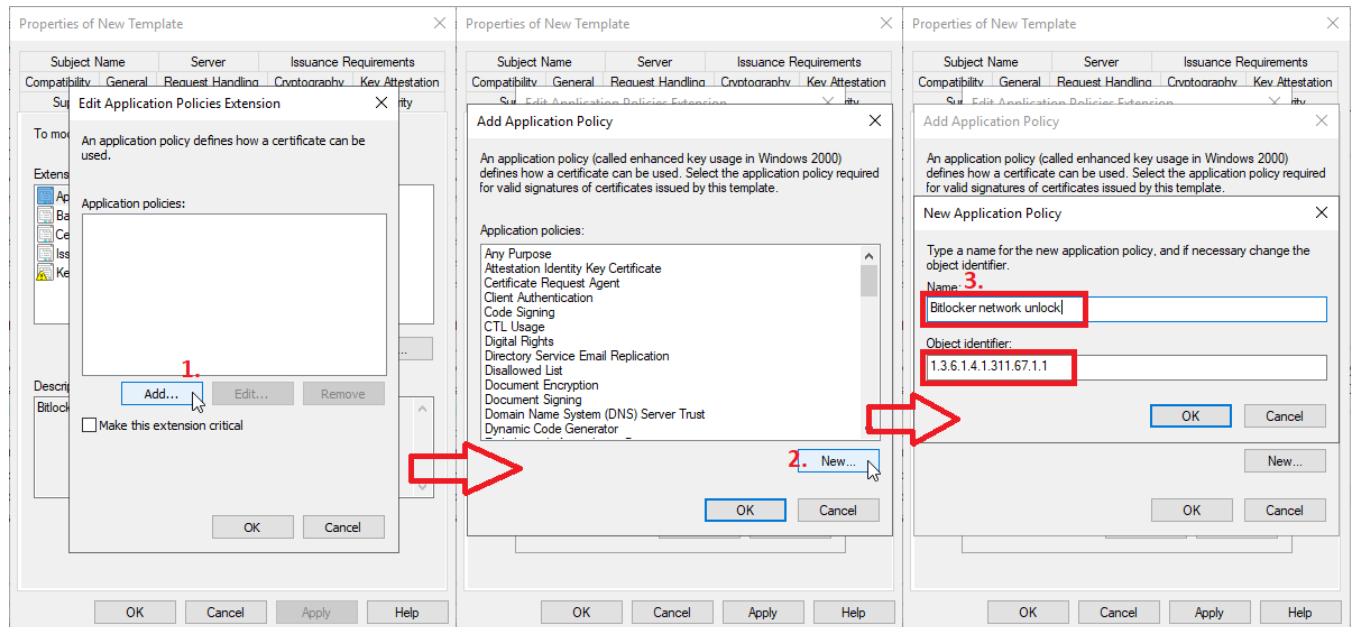
- Cryptography: Requests must use one of the following providers: Microsoft Base Smart Card Crypto Provider



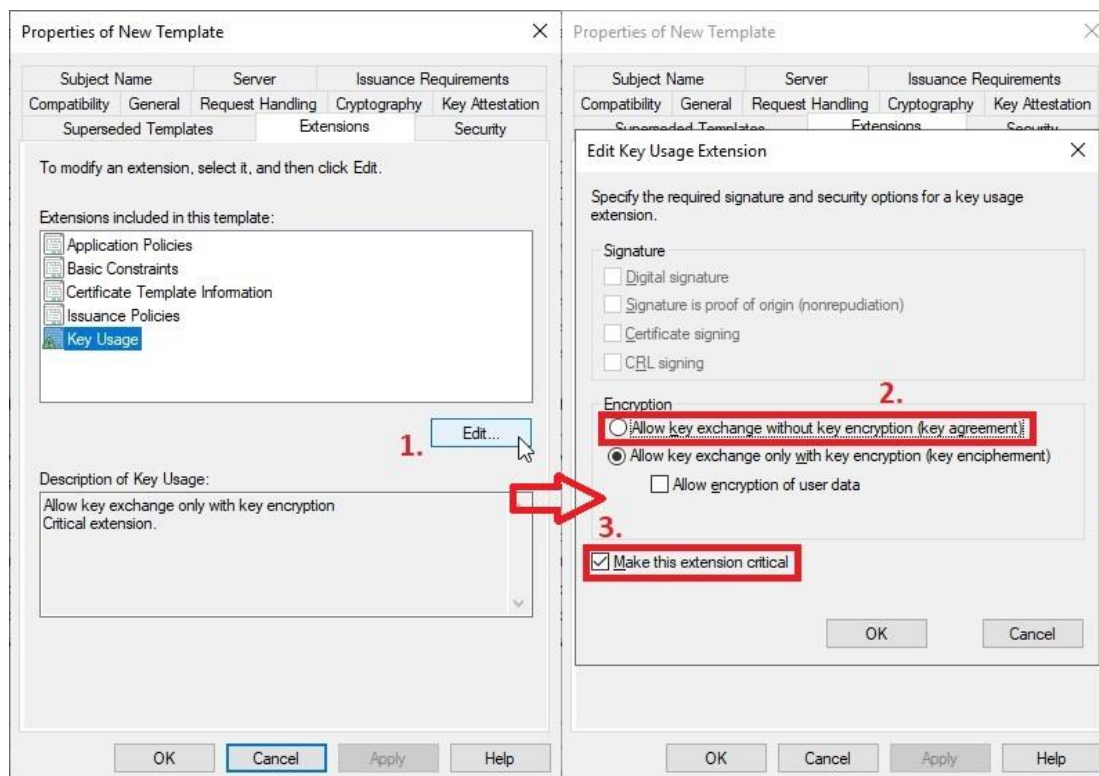
- Extensions:
 - Application Policies: Edit... -> - Remove all;



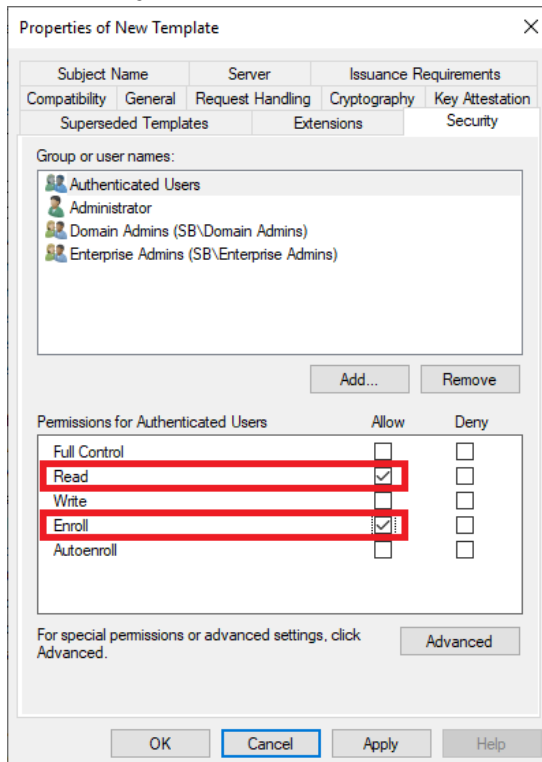
Add... -> New... -> Name: your choice (e.g. "Bitlocker network unlock"), Object identifier: 1.3.6.1.4.1.311.67.1.1 (default in Bitlocker policies, must correspond to settings on client);



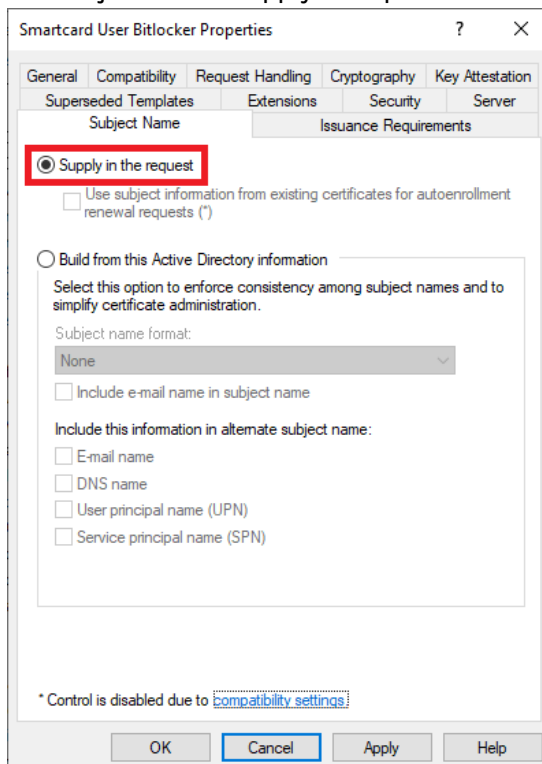
- Key Usages:
 - Allow key exchange only with key extension
 - Make this extension critical extension.



- Security: Authenticated Users > allow *Read* and *Enroll*

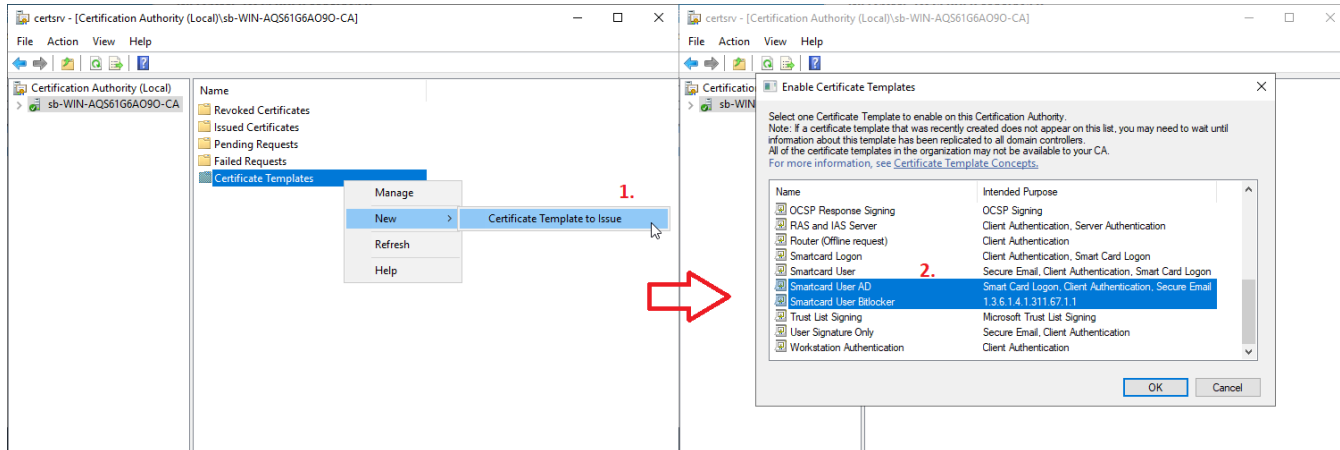


- Subject Name: Supply in request -> Accept warning

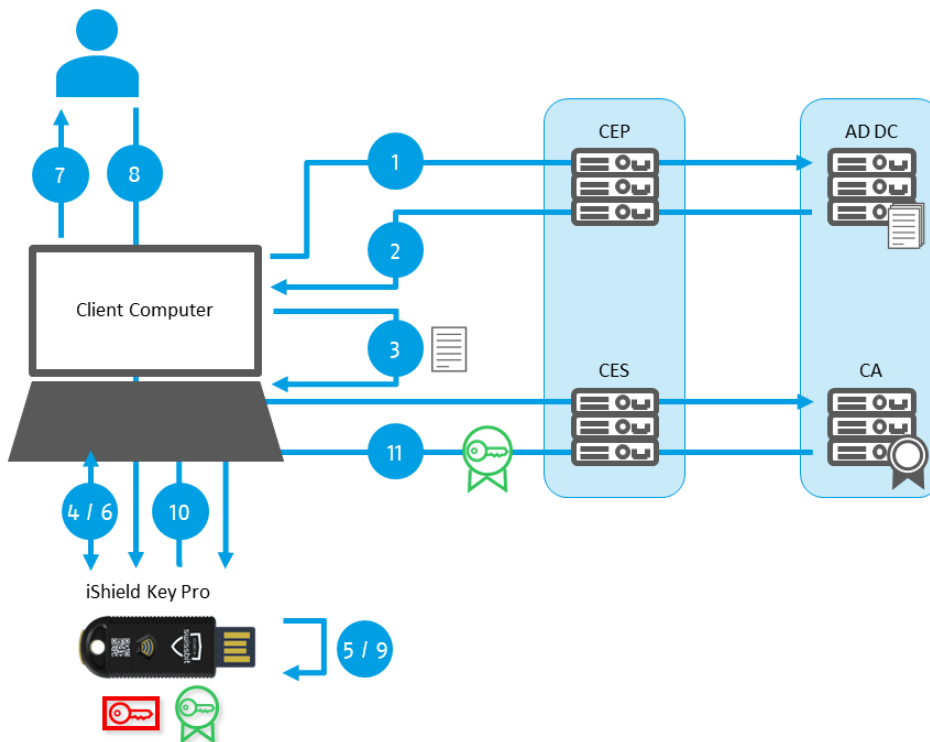


Enable the created Template:

5. Click OK to save the template.
6. In certsrv: go to your current domain > right click Certificate Templates > New > Certificate Template to Issue and select the template you have just created.



7.6.2 Self-enroll Certificate on Client PC



Certificate Enrollment with AD Certificate Services

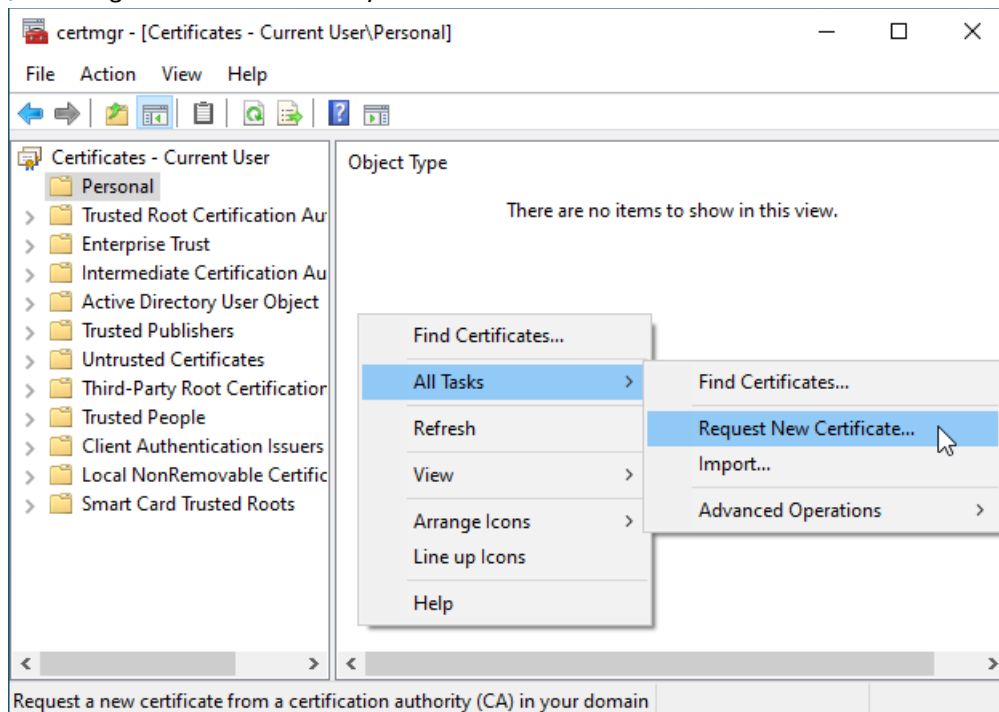
1. Request Policies from Active Directory Domain Controller (AD DC) via Certificate Enrollment Policy Service (CEP)
2. AD DC provides policies via CEP to client
3. User selects certificate template
4. User authenticates as card administrator via management key
5. New public/private key pair is generated on iShield FIDO2 key
6. Generate certificate signing request (CSR) comprising public key and certificate template information
7. Request user PIN
8. User authenticates via PIN
9. CSR is signed with private key to prove ownership to Certificate Authority (CA)
10. CSR is sent to CA via Certificate Enrollment Service (CES) for signing
11. CA signed certificate is returned to client via CES and imported into iShield Key Pro

This requires you to install the OpenSC Minidriver and the iShield PIV Module.

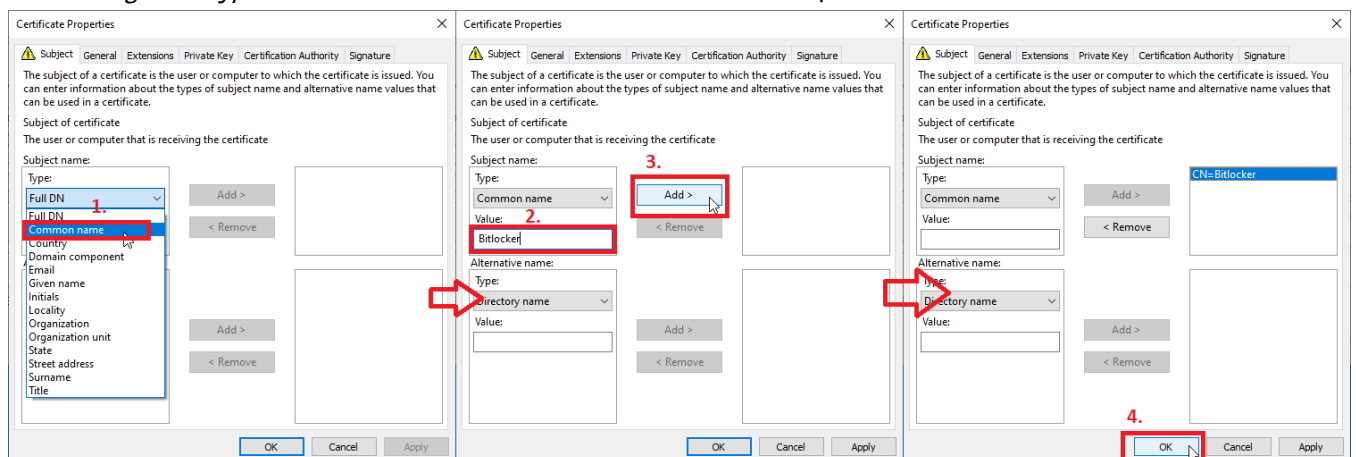
Setup Client certificates management

On the Workstation log in with the user account for which the smartcard Bitlocker encryption should be activated. Plug in your smartcard on the client PC.

1. Go to Start-Menu and search for "Manage User Certificates"
2. Open *Certificates - Current User > Personal > Certificates* and right-click in the blank space
3. Navigate to *All Tasks > Request New Certificate*. The Certificate Enrollment wizard should open



4. Click through it: Select the *Active Directory Enrollment Policy* then *Next*
5. Select the template for the **Bitlocker** use-case, then *Enroll*
6. An exclamation point should have popped up underneath the Template name *Click here to configure settings*. Set *type* to "Common name" and *value* to Bitlocker; click *Add >* and *OK*



A prompt for the smartcard's PIN will show up. After entering the PIN, Windows should display a success message.

Enable Bitlocker for one storage medium

After this step, you have prepared an external flash drive or (internal) data-disk for Bitlocker. All the present data will be encrypted retroactively and all future data stored there will be encrypted. Note that this does not work with boot drives, but only data disks.

Note: For this step, you do not need to install the OpenSC Minidriver and iShield PIV module anymore. As soon as there is a valid certificate on the iShield Key Pro, the pre-installed Windows tools work by themselves.

Insert the device you want to encrypt in your PC and open it in the **Windows Explorer**.

- Right-click on the drive and select **Turn on BitLocker**.
- Check **Use my smart card to unlock the drive** and click *Next*.
- Store your recovery key in some secure place. In case you were to lose your iShield Key Pro in the future, you can still recover your encrypted drive. Click *Next*.
- For the next options, you can chose whichever you want. Finally click **Start encrypting**. Depending on the options you have chosen, this might take a while.

The drive and smartcard have now been prepared to work with Bitlocker on any Windows 10 Pro PC.

7.6.3 Use it on Client

As soon as the iShield Key Pro and external drive are prepared, the default Windows PIV driver is sufficient for usage with Bitlocker. You will not need to install the OpenSC Minidriver or iShield PIV module in this case.

- Insert your drive. You should see a message "This drive is BitLocker-protected".
- Access the drive via the Explorer. A prompt for your smart card (iShield Key Pro) or your recovery key will appear.
- Select the "smart card" option and provide your smart card PIN.
- You will find the drive unlocked and usable until you unplug and reinsert it

7.7 Use Case: Active Directory PC logon

In this scenario, the PC, on which Bitlocker is used, is a workstation within an **active directory domain**. The domain server will manage all the domain information. Upon request, the certificate authority on the domain server will issue and sign the used certificate. This is ideal for companies, in which the AD infrastructure already exists. You can increase your data security by following the next steps.

This guide expects a working AD setup.

7.7.1 Setup on Server

This section explains the necessary steps to configure a Microsoft Active Directory Domain Server for this use case. These instructions target the IT Administrator of a domain and you cannot conduct those on a personal computer or domain workstation.

Requirements beforehand

All users, that want to self-enroll certificates for Logon, need an email address assigned to them in the user management tool.

In **Server Manager**:

- Go to *Tools > Active Directory Users and Computers*
- In your domain go to *Users* and **right click the user**
- Click *Properties > General*. Make sure an email address is assigned to them

Setup Bitlocker Template

On the AD-ROOT-Server, open the **Server Manager**.

1. In the top right go to *Tools > Certification Authority*
2. In **certsrv** expand the server name
3. Right click *Certificate Templates > Manage*

Setup the Bitlocker Template:

4. In Certificate Templates Console right click Smartcard User > Duplicate Template and adjust the Properties of each of the tabs as follows:
 - Compatibility: leave as-is
 - General: Rename (e.g. Smartcard User AD)
 - Request Handling: Purpose: *Signature and smartcard logon*, if warning box appears, click Yes
 - Cryptography: Minimum key size: 2048, *Requests must use one of the following providers*, Providers: *Microsoft Base Smart Card Crypto Provider*
 - Security: Authenticated Users > allow *Read* and *Enroll*
 - Subject Name: *Include e-mail in subject name* and *E-mail name*

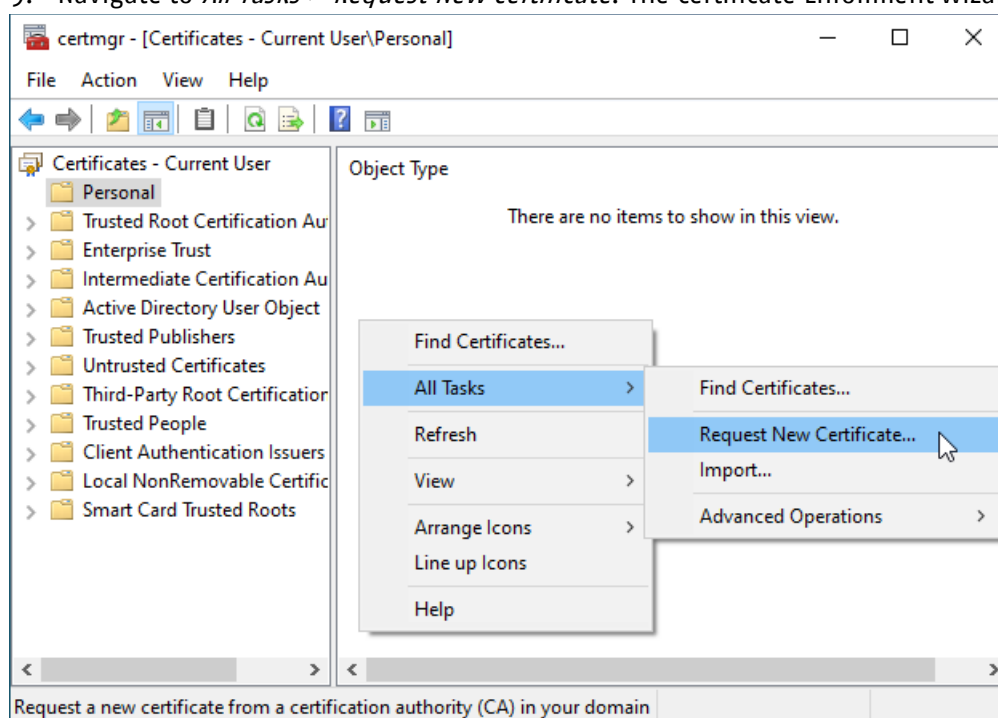
Enable the created Template:

5. Click *OK* to save the template.
6. In **certsrv**: go to your current domain > right click *Certificate Templates > New > Certificate Template to Issue* and select the template you have just created.

7.7.2 Self-enroll Certificate on Client PC

On the Workstation log in with the user account for which the smartcard logon should be activated. Plug in your smartcard on the client PC.

7. Go to Start-Menu and search for "Manage User Certificates"
8. Open *Certificates - Current User > Personal > Certificates* and right-click in the blank space
9. Navigate to *All Tasks > Request New Certificate*. The Certificate Enrollment wizard should open



10. Click through it: Select the *Active Directory Enrollment Policy* then *Next*
11. Select the template for the **Logon** use-case, then *Enroll*

A prompt for the smartcard's PIN will show up. After entering the PIN, Windows should display a success message. We recommend restarting your PC after certificate enrollment.

7.7.3 Use it on Client

From now on, whenever you lock your PC or log out of your account, you will be able to use your Swissbit iShield Key Pro to logon to your account.

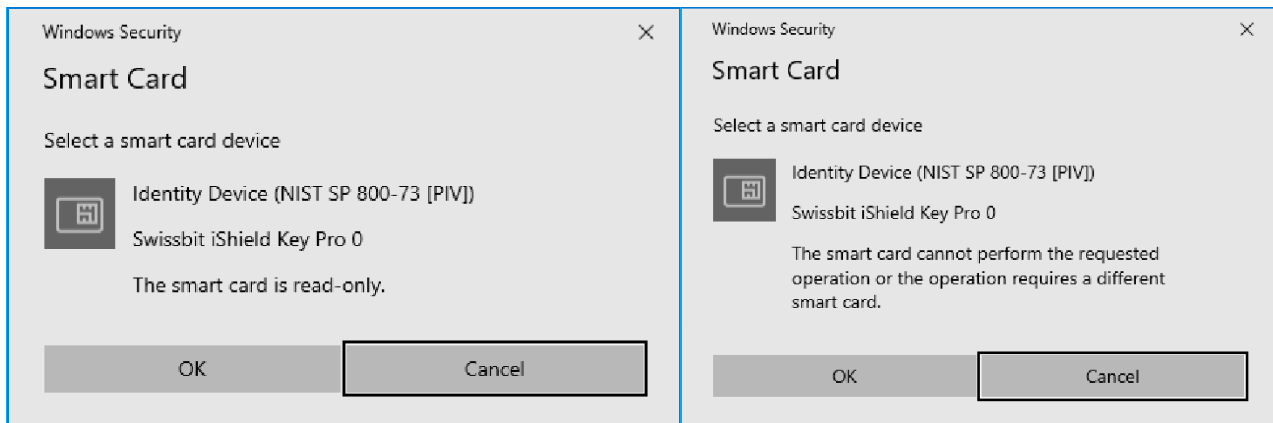
- When you see the logon screen, you should have plugged in your iShield Key Pro.
- If it does not automatically prompt you for your "Smart card PIN", you can navigate to "Sign-in options" and select the Smart card symbol
- Provide the pin of your iShield Key Pro.

If you have implemented all steps correctly, you should be logged-in by now.

7.8 Troubleshooting

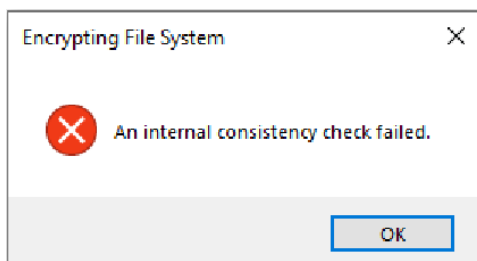
7.8.1 Troubleshooting “The smart card is read-only / cannot perform the requested operation”

If your iShield Key Pro is displayed to be read-only or to not support the requested operation, the OpenSC minidriver is not properly installed. For provisioning your key you need to use the OpenSC minidriver, see 7.2.1. Please verify that you correctly installed a compatible OpenSC version including the OpenSC minidriver.



7.8.2 Troubleshooting “An internal consistency check failed”

The error “An internal consistency check failed” is commonly caused by misconfiguration of OpenSC. Please make sure to follow all steps in section 7.4: Verify your OpenSC profiles directory, management key file, environment variables and OpenSC configuration file, in particular the iShield PIV module path and presence of required system runtime libraries.



8 Glossary

Abbreviation	Description
2FA	Two-Factor Authentication
AD	Active Directory
CA	Certificate Authority
CEP	Certificate Enrollment Policy Service
CES	Certificate Enrollment Service
CSP	Cryptographic Service Provider
CSR	Certificate Signing Request
DC	Domain Controller
EFS	Encrypting File System
FIDO	Fast Identity Online
HMAC	Hashed Message Authentication Code
HOTP	HMAC-based One-Time Password
IETF	Internet Engineering Task Force
iKM	iShield Key Manager
MFA	Multi-Factor Authentication
NFC	Near Field Communication
OIDC	OpenID Connect
OTP	One-Time Password
PIN	Personal Identification Number
PIV	Personal Identity Verification
PKCS	Public-Key Cryptography Standard
PUK	Personal Unblocking Key
SHA	Secure Hash Algorithm
SSO	Single Sign-On
TOTP	Time-based One-Time Password
U2F	Universal 2 nd Factor
VPN	Virtual Private Network

9 Document History

Version	Updated on	Updated by	Short description
1.0.0	10.03.2023	Swissbit AG	First Version
1.1.0	13.07.2023	Swissbit AG	Minor Change
1.2.0	31.08.2023	Swissbit AG	Added iShield Key Manager GUI tool
1.3.0	30.10.2023	Swissbit AG	TOTP functionality
1.4.0	20.11.2023	Swissbit AG	OpenSC Configuration for PIV over contactless interface
1.5.0	19.12.2023	Swissbit AG	iShield Key Manager for macOS
1.6.0	31.01.2024	Swissbit AG	Added iKMcli commands for PIV certificate enrollment
1.7.0	28.02.2024	Swissbit AG	iShield Key Manager for Linux

Disclaimer:

No part of this document may be copied or reproduced in any form or by any means, or transferred to any third party, without the prior written consent of an authorized representative of Swissbit AG ("SWISSBIT"). The information in this document is subject to change without notice. SWISSBIT assumes no responsibility for any errors or omissions that may appear in this document, and disclaims responsibility for any consequences resulting from the use of the information set forth herein. SWISSBIT makes no commitments to update or to keep current information contained in this document. The products listed in this document are not suitable for use in applications such as, but not limited to, aircraft control systems, aerospace equipment, submarine cables, nuclear reactor control systems and life support systems. Moreover, SWISSBIT does not recommend or approve the use of any of its products in life support devices or systems or in any application where failure could result in injury or death. If a customer wishes to use SWISSBIT products in applications not intended by SWISSBIT, said customer must contact an authorized SWISSBIT representative to determine SWISSBIT willingness to support a given application. The information set forth in this document does not convey any license under the copyrights, patent rights, trademarks or other intellectual property rights claimed and owned by SWISSBIT. The information set forth in this document is considered to be "Proprietary" and "Confidential" property owned by SWISSBIT.

ALL PRODUCTS SOLD BY SWISSBIT ARE COVERED BY THE PROVISIONS APPEARING IN SWISSBIT'S TERMS AND CONDITIONS OF SALE ONLY, INCLUDING THE LIMITATIONS OF LIABILITY, WARRANTY AND INFRINGEMENT PROVISIONS. SWISSBIT MAKES NO WARRANTIES OF ANY KIND, EXPRESS, STATUTORY, IMPLIED OR OTHERWISE, REGARDING INFORMATION SET FORTH HEREIN OR REGARDING THE FREEDOM OF THE DESCRIBED PRODUCTS FROM INTELLECTUAL PROPERTY INFRINGEMENT, AND EXPRESSLY DISCLAIMS ANY SUCH WARRANTIES INCLUDING WITHOUT LIMITATION ANY EXPRESS, STATUTORY OR IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

©2023 SWISSBIT AG All rights reserved.