
YubiKey 5 Series Technical Manual

Yubico

Feb 24, 2023

CONTENTS

1	Introductions to the Different YubiKey Series	1
1.1	YubiKey 5 Series	1
1.2	YubiKey 5 FIPS Series	1
1.3	Security Key Series	4
1.4	YubiKey Bio Series	5
1.5	YubiKey 5 CSPN Series	6
2	What's New?	9
2.1	YubiKey 5Ci	9
2.2	NFC	10
2.3	USB	10
3	Firmware: Overview of Features & Capabilities	11
3.1	Overview of Capabilities	12
3.2	Secure Channel	14
3.3	PIV Enhancements	17
3.4	NFC ID: Calculation Changed	17
3.5	YubiHSM Auth	17
4	Physical Attributes	19
4.1	YubiKey 5 NFC	19
4.2	YubiKey 5 Nano	20
4.3	YubiKey 5C	20
4.4	YubiKey 5C Nano	21
4.5	YubiKey 5Ci	21
4.6	YubiKey 5C NFC	21
4.7	YubiKey Bio Series	22
4.8	FIPS-specific Marking	24
4.9	CSPN-specific Marking	24
4.10	Security Key Series Marking	25
5	Physical Interfaces: USB, NFC, Apple Lightning®	27
5.1	USB	27
5.2	Apple Lightning®	27
5.3	NFC	28
6	Understanding the USB Interfaces	29
6.1	OTP	29
6.2	FIDO	29
6.3	CCID	29

7	Protocols and Applications	31
7.1	FIDO2	31
7.2	FIDO U2F	33
7.3	OATH	33
7.4	OpenPGP	33
7.5	OTP	34
7.6	Smart Card (PIV Compatible)	35
7.7	YubiHSM Auth	41
8	Tools and Troubleshooting	47
8.1	Managing Applications	47
8.2	YubiKey Manager (ykman)	47
8.3	Yubico Authenticator	48
8.4	YubiKey Smart Card Minidriver	48
8.5	Troubleshooting	49
9	NFC ID Calculation Technical Description	51
9.1	Background to Door Access	51
9.2	Calculation of NFC ID	51
9.3	NFC ID Calculation for YubiKey v5.3.0 and Above	51
10	Secure Channel Specifics	53
10.1	Yubico Secure Channel Technical Description	53
10.2	Yubico Secure Channel Key Diversification and Programming	60
10.3	Yubico SCP03 Developer Guidance	63
11	YubiKey 5 FIPS Series Specifics	67
11.1	Deploying the YubiKey 5 FIPS Series	67
11.2	OTP: FIPS 140-2 with YubiKey 5 FIPS Series	70
11.3	OATH: FIPS 140-2 with YubiKey 5 FIPS Series	71
11.4	FIDO: FIPS 140-2 with YubiKey 5 FIPS Series	72
11.5	PIV: FIPS 140-2 with YubiKey 5 FIPS Series	73
11.6	FIPS Level 1 vs FIPS Level 2	78
12	YubiKey 5 CSPN Series Specifics	81
12.1	CSPN Mode Configuration	81
12.2	One-Time Password - OTP	82
12.3	OATH	89
12.4	FIDO U2F	91
12.5	FIDO2	92
12.6	PIV	96
13	YubiKey Bio Series Specifics	101
13.1	How the YubiKey Bio Works	101
13.2	Using Chrome to Enroll Fingerprints	104
13.3	Using Windows to Enroll Fingerprints	107
13.4	Tips	108
13.5	Troubleshooting and Tools	110
13.6	Requirements: Platform and Browser Compatibility	111
13.7	Resetting Your YubiKey Bio with the Yubico Authenticator for Desktop	112
13.8	Frequently Asked Questions	112
13.9	YubiKey Bio and FIDO2	112
13.10	YubiKey Bio and FIDO U2F	114
14	Acronyms	117

15 Copyright	119
15.1 Trademarks	119

INTRODUCTIONS TO THE DIFFERENT YUBIKEY SERIES

Throughout the YubiKey Technical Manual different YubiKeys will be referred to as e.g. “YubiKey 5 (FIPS/CSPN) Series”, indicating that a certain specification or feature is available on the YubiKey 5 Series, the YubiKey 5 FIPS Series and the YubiKey 5 CSPN Series, due to the fact that they share the same base hardware and many firmware features.

1.1 YubiKey 5 Series

1.1.1 About the YubiKey 5 Series

The YubiKey 5 Series security keys offer strong authentication with support for multiple protocols, including FIDO2, which is the new standard that enables the replacement of password-based authentication. The YubiKey strengthens security by replacing passwords with strong hardware-based authentication using public key cryptography.

- For those who just want to use a YubiKey without programming anything, the most useful part of this guide will be *Understanding the USB Interfaces*, which describes how the YubiKey connects, and indicates what it can connect to.

For an overview on setting up two-step verification in a typical case, see [Google on using a security key for 2-step verification](#).

- The full list of the services that work with YubiKeys is on Yubico’s [Works With YubiKey](#) page.
- Most of the rest of this guide targets systems integrators, IT teams, or developers who expect to integrate support for YubiKeys into their environment.

All the YubiKeys in the YubiKey 5 Series have the basic functionalities and capabilities described in this guide. However, it is the firmware version that determines which of the more specialized functionalities and capabilities are available to your YubiKey.

1.2 YubiKey 5 FIPS Series

1.2.1 Why FIPS?

Federal Information Processing Standards (FIPS) are developed by the United States government for use in computer systems to establish requirements such as ensuring computer security and interoperability. The [National Institute of Standards and Technology \(NIST\)](#) and the Canadian Centre for Cyber Security (CCCS) run the NIST Cryptographic Module Validation Program (CMVP) as a collaborative effort.

FIPS certification demonstrates that a product has gone through a rigorous audit process and adheres to a security standard that can be measured and quantified.

Many government organizations and government contractors are required to use FIPS-approved products, as are highly-regulated industries in general. Other countries also recognize FIPS 140-2. For the US government, the default is that FIPS is **required**.

1.2.2 Do You Require FIPS Keys?

If you do not have a security auditor, and/or the auditor does not have a compliance requirement, you probably do not need FIPS. The standard line of YubiKeys offers the same security, algorithms and functionality. The standard line also evolves at a much more rapid pace because it does not need to go through an exhaustive validation process, which commonly takes a year or more. Yubico can release standard firmware with new features, enhancements, etc. at any time, whereas FIPS-certified products must go through the FIPS validation process every time there is a change.

1.2.3 About the YubiKey 5 FIPS Series

The YubiKey 5 FIPS Series is FIPS 140-2 certified. It offers strong authentication with support for multiple protocols - including FIDO2, which is the new standard that enables the replacement of password-based authentication. The YubiKey strengthens security by replacing passwords with strong hardware-based authentication using public key cryptography.

The cryptographic functionality of the YubiKey 5 FIPS Series devices is powered by the FIPS 140-2 certified YubiKey 5 cryptographic module, a single-chip cryptographic processor with a non-extractable key store that handles all of the cryptographic operations. The YubiKey 5 cryptographic module is FIPS 140-2 certified, both Level 1 and Level 2 (Physical Security Level 3).

The YubiKey 5 FIPS Series cryptographic module is a secure element that supports multiple protocols designed to be embedded in USB security tokens. The module can generate, store, and perform cryptographic operations for sensitive data and can be utilized via an external touch-button for Test of User Presence in addition to PIN for smart card authentication. The module implements the following major functions, depending on the firmware version you have:

Function	Firmware Versions	
	5.4.2	5.4.3
Yubico One Time Password (OTP)	yes	yes
OATH OTP authentication	yes	yes
OpenPGP (version 3.4)	•	yes
PIV-compatible smart card	yes	yes
FIDO Universal 2nd Factor (U2F)	yes	yes
FIDO2 WebAuthn	yes	yes
YubiHSM Auth	•	yes
SCP03	yes	yes

The YubiKey 5 FIPS Series hardware with the 5.4 firmware is certified as an authenticator under both FIPS 140-2 Level 1 and Level 2. It meets the highest authenticator assurance level 3 (AAL3) of NIST SP800-63B guidance. To use security keys from the YubiKey 5 FIPS Series as a Level 2, more stringent initialization is required than for Level 1. Guidance for Level 2 is set out in detail in the following.

1.2.4 FIPS-specific Aspects of the YubiKey 5 FIPS Series

Distinguishing the YubiKey 5 FIPS Series from the YubiKey 5 Series with the 5.4 firmware are the following configuration changes, set at programming:

Configuration Change	Description
Functional	Enforce power-up self-test (firmware integrity and algorithm testing)
Minimum PIN length for FIDO2	6 alphanumeric characters
Identification (FIDO)	Unique AAGUIDs for the FIDO Attestation (see <i>AAGUID Values</i>)
Attestation (FIDO)	Attestation certificates for FIDO include a FIPS OID (1.3.6.1.4.1.41482.12)
FIDO GETINFO	Command returns a listing of FIPS, as well as the FIPS-specific OIDs in the PIV and FIDO attestation certificates.*
Attestation (PIV)	Attestation certificates for PIV include the FIPS Form Factor identifier** in the Form Factor OID (1.3.6.1.4.1.41482.3.9)
YubiKey Manager	Form factor identifies FIPS Series devices.**

* The certifications that are supported by a FIDO authenticator can be returned in the `certifications` member of an `authenticatorGetInfo` response as set out in paragraph 7.3.1. *Authenticator Actions of the Client to Authenticator Protocol (CTAP) Review Draft of March 09, 2021*.

** Form factor is set during manufacturing and returned as a one-byte value. Currently defined values for this are:

Table 1: Form Factor

Form Factor	Standard YubiKey Value	Security Key Value (FW 5.4+)	FIPS YubiKey Value (FW 5.4+)
UNDEFINED	0x00	N/A	N/A
Keychain with USB-A	0x01	0x41	0x81
Nano with USB-A	0x02	N/A	0x82
Keychain with USB-C	0x03	0x43	0x83
Nano with USB-C	0x04	N/A	0x84
Keychain with Lightning and USB-C	0x05	N/A	0x85

1.2.5 Firmware

The YubiKey firmware is separate from the YubiKey itself in the sense that it is put onto each YubiKey in a process separate from the manufacture of the physical key. Nonetheless, it can be neither removed nor altered. Yubico periodically updates the YubiKey firmware to take advantage of features and capabilities introduced into operating systems such as Windows, MacOS, and Ubuntu, etc., as well as to enable new YubiKey features.

The firmware version on a YubiKey or an HSM therefore determines whether or not a feature or a capability is available to that device. The quickest and most convenient way to determine your device's firmware version is to use the YubiKey Manager tool (ykman), a lightweight software package installable on any OS. The YubiKey Manager has both a graphical user interface (GUI) and a command line interface (CLI).

- Download the YubiKey Manager tool: <https://www.yubico.com/products/services-software/download/yubikey-manager/>
- [YubiKey Manager \(ykman\) CLI & GUI Guide](#)

Yubico has submitted the same firmware - releases 5.4.2 and 5.4.3 - to NIST and it has submitted release 5.4.2 to ANSSI for certification. Both organizations have approved certification.

1.3 Security Key Series

1.3.1 Overview

The *Security Key Series* differs from a YubiKey 5 Series in that it comes only with the FIDO (FIDO2/FIDO U2F) protocol and does not have a serial number. It is only available in USB-A + NFC and USB-C + NFC form factors.

However, the *Security Key Series - Enterprise Edition* is the same as a *Security Key Series* but with a serial number to allow for asset tracking. The serial number can be read visually on the back of the key and programatically through the FIDO HID interface. It is only available in USB-A + NFC and USB-C + NFC form factors.

[Get started with Security Key Series \(video tutorial\)](#)

1.4 YubiKey Bio Series

The YubiKey Bio Series offers the familiar YubiKey experience users have come to know and trust, but adds the convenience of a new biometric touch feature.

The series is comprised of two keys:

- The YubiKey Bio - FIDO Edition (USB-A form factor)
- The YubiKey C Bio - FIDO Edition (USB-C form factor)

1.4.1 Protocols Supported

Both keys in the YubiKey Bio Series support the FIDO authentication protocols, and will work with sites and applications that support the FIDO2 and FIDO U2F protocols (for more information, see [YubiKey Bio and FIDO2](#) and [u2f-label](#)). FIDO2 (sometimes referred to as WebAuthn) builds upon FIDO U2F, and is the standard which enables the replacement of password-based authentication.

The YubiKey Bio Series provides firmware applications to support two modes of authentication via the FIDO2 and U2F protocols (see [YubiKey Bio and FIDO2](#) and [u2f-label](#)). Even though the firmware applications are separate from one another, they both share the same PIN and FIDO reset capability. In fact, a FIDO reset will reset both applications (to manage these applications, see [bio-tools-label](#)).

1.4.2 Using the YubiKey Bio

To just start using the keys in the YubiKey Bio Series without going into any details, refer to [Yubico's setup page](#), which functions as a **quick start guide**.

The current guide, however, gives:

- An explanation of the way the YubiKey Bio works and descriptions of the different user experiences with the various protocols
- Full instructions for enrolling fingerprints using platform support:
 - [Using Chrome to Enroll Fingerprints](#) and
 - [Using Windows to Enroll Fingerprints](#)
- Brief descriptions of the protocols supported in [YubiKey Bio and FIDO2](#) and [u2f-label](#)
- A brief explanation of the role the Yubico Authenticator for Desktop plays in managing the YubiKey Bio, plus links for downloading it and to its documentation.

1.4.3 Usage Notes

The YubiKey Bio implements biometrics as outlined in the [CTAP 2.1 specification](#). The best user experiences are provided by the YubiKey Bio with client applications and browsers that also implement CTAP 2.1. Applications and browsers that implement CTAP 1 or CTAP 2.0 will also work with the YubiKey Bio; however, the UI on client devices will not be as intuitive, and there may be some limitations.

1.4.4 Interfaces and Applications

Interfaces

Like all other YubiKeys, the YubiKey Bio Series are USB 2.0 devices.

Note: Developers: The USB PID and iProduct string are `0x0402` and `YubiKey FIDO` respectively (see [YubiKey USB ID Values](#)).

Applications

All keys in the YubiKey Bio Series support WebAuthn sites and applications that support the FIDO2 and FIDO U2F protocols (for more information, see [YubiKey Bio and FIDO2](#) and `u2f-label`). FIDO2 (also sometimes referred to as WebAuthn) is also the standard that enables the replacement of password-based authentication.

Each application can be enabled and disabled independently. Up to five fingerprints can be stored on a YubiKey Bio. For management, see `tools-label`.

1.5 YubiKey 5 CSPN Series

1.5.1 Scope

The aim of this document is to describe how to configure and use the YubiKey 5 in a mode such that it is compliant with CSPN (“Certificat de Sécurité de Premier Niveau” [RD1]).

For each YubiKey application which will require specific configuration, there will be a short introduction, followed by the required settings to achieve the target, and finally, a technical description of the configuration itself.

1.5.2 References

Code	Document title	Reference
[RD1]	Certification de sécurité de premier niveau des technologies de l’information	https://www.ssi.gouv.fr/administration/produits-certifies/cspn/
[RD2]	Certification Report BSI-DSZ-CC-0879-V4-2020	https://www.bsi.bund.de/SharedDocs/Zertifikate_CC/CC/SmartCards_IC_Cryptolib/0879_0879V2_0879V3_0879V4.html
[RD3]	FIDO2: WebAuthn & CTAP	https://fidoalliance.org/fido2/
[RD4]	NIST Special Publication 800-73 (PIV)	https://csrc.nist.gov/publications/detail/sp/800-73/4/final
[RD5]	RFC 4226, An HMAC-Based One-Time Password Algorithm	https://tools.ietf.org/html/rfc4226
[RD6]	T/Key: Second-Factor Authentication From Secure Hash Chains	https://arxiv.org/pdf/1708.08424.pdf
[RD7]	Universal 2nd Factor (U2F) Overview	https://fidoalliance.org/specs/fido-u2f-v1.2-ps-20170411/fido-u2f-overview-v1.2-ps-20170411.html
[RD8]	W3C WebAuthn standard	https://www.w3.org/TR/webauthn-2/
[RD9]	YubiKey CSPN security target	https://www.ssi.gouv.fr/uploads/2021/09/anssi-cible-cspn-2021_18en.pdf

1.5.3 Acronyms

Acronym	Description
2FA	Two-Factor Authentication
AES	Advanced Encryption Standard
BSI	Bundesamt für Sicherheit in der Informationstechnik
CC	Common Criteria
CCID	Chip Card Interface Device
CSPN	Certificat de Sécurité de Premier Niveau
CTAP2	Client to Authenticator Protocol v2
DES	Data Encryption Standard
FIDO	Fast Identity Online
HMAC	Hash-Based Message Authentication Code
HOTP	HMAC-Based One Time Password
NIST	National Institute of Standards and Technology
OATH	Open AuTHentication
OTP	One Time Password
PIV	Personal Identity Verification
PBKDF2	Password Based Key Derivation Function
PIN	Personal Identification Number
PIV	Personal Identity Verification
PUK	PIN Unblocking Key
SHA	Secure Hash Algorithm
TOTP	Time-Based One Time Password
U2F	Universal Second Factor
RFC	Request For Comments
W3C	World Wide Web Consortium

To get in touch with Yubico Support, [click here](#).

WHAT'S NEW?

The capabilities of the YubiKey 5 Series are dependent on the different combinations of firmware + connector type + protocol. This section covers connector types (form factors). Capabilities brought to the YubiKey 5 Series by the firmware are covered in *Firmware: Overview of Features & Capabilities* and in *Protocols and Applications*.

2.1 YubiKey 5Ci

The YubiKey 5Ci is the first hardware authenticator of its kind with both USB-C and Lightning® connectors on a single security key. With multi-protocol capabilities that support OTP, U2F, *FIDO2*/WebAuthn, and Smart Card requirements, the YubiKey 5Ci provides a unified solution for secure logins on mobile and computing devices.

2.1.1 Lightning® Connector

The YubiKey 5Ci introduced support for Apple's Lightning® connector. All features of the YubiKey 5 are supported over Lightning®, including FIDO2, PIV, OpenPGP, OATH and OTP. The YubiKey 5Ci is the first YubiKey to roll out new feature enhancements to FIDO2 and OpenPGP. Details on the new functionality can be found in the [Enhancements to FIDO 2 Support](#) and in the [Enhancements to OpenPGP Support](#).

Like the USB interface, the YubiKey 5Ci's Lightning® interface also uses a variety of channels for communication between the YubiKey and iOS.

The YubiKey 5Ci presents itself as an Apple iOS peripheral. It is able to interact with:

- Any iOS app using the Yubico YubiKey iOS SDK
- Any app input data field via touch-triggered OTP.
- Any WebAuthn-compliant application (starting in iOS 13). This includes the Safari browser.

When connecting the YubiKey 5Ci via Lightning®, the **interfaces enabled** setting is common to both USB-C and Lightning®. Enabling or disabling an interface will apply to both connections.

Note: Developers: for apps within iOS to be able to use advanced protocols that send and receive information from the YubiKey 5Ci, the Yubico SDK is required (the Yubico iOS SDK can be accessed at <https://github.com/YubicoLabs/yubikit-ios>) and the app registered with Yubico. This can be done via the [Yubico iOS SDK App submission page](#).

The USB and iProduct strings that show up when connecting via Lightning® or USB are specific to the connection type. They are described in the [YubiKey USB ID Values guide](#).

2.1.2 iPad and iPad Pro

For users of keys in the YubiKey 5 Series, because the iPad Pro does not have a Lightning port, support depends on what you want to do. The second part of this article covers all those aspects: [Can I use my YubiKey with iPads?](#)

From the developer perspective, support for the iPad Pro has some limitations. Consult [Supporting U2F or FIDO2 Security Keys on iOS or iPadOS | Security Key Compatibility](#) for detailed instructions on working around those limitations.

2.2 NFC

Expanding the options for quick tap-n-go authentication across desktops, laptops, and mobile devices, all of the applications - including *FIDO2* - are available over NFC.

The YubiKey 5 NFC and YubiKey 5C NFC support the iPhone 7 and newer.

Background tag reading is supported in the iPhone XS and newer.

The YubiKey 5 NFC and YubiKey 5C NFC provide an NFC wireless interface in addition to USB. The YubiKey 5 NFC and YubiKey 5C NFC include the RFID standard specific to the ISO/IEC 14443-A and ISO/IEC 14443-4 NFC format; RFID implementations not included in the listed ISO standards are not supported.

The NDEF URI has been updated to a new format; an example of the new format is provided below. The <OTP> value is replaced with the OTP generated by the YubiKey.

<https://demo.yubico.com/yk/>

For operations that require a touch, all touch requests within the first 20 seconds of the operation will succeed. After a period of inactivity, a YubiKey placed on a desktop NFC reader may power down to help prevent unintended access. To regain connectivity with an NFC reader, remove the YubiKey from the reader and reposition it on the reader. Some NFC readers may power-cycle and in doing so, prevent the YubiKey from powering down.

2.3 USB

All of the models in the YubiKey 5 Series provide a USB 2.0 interface, regardless of the form factor of the USB connector. The YubiKey will present itself as a USB composite device in addition to each individual USB interface.

The USB PID and iProduct string changes depending on which of the USB interfaces are enabled. They are described in the [YubiKey USB ID Values guide](#).

See also *Understanding the USB Interfaces*.

To get in touch with Yubico Support, [click here](#).

FIRMWARE: OVERVIEW OF FEATURES & CAPABILITIES

The YubiKey firmware is separate from the YubiKey itself in the sense that it is put onto each YubiKey in a process separate from the manufacture of the physical key. Nonetheless, it can be neither removed nor altered. Yubico periodically updates the YubiKey firmware to take advantage of features and capabilities introduced into operating systems such as Windows, MacOS, and Ubuntu, etc., as well as to enable new YubiKey features.

The firmware version on a YubiKey or an HSM therefore determines whether or not a feature or a capability is available to that device. The quickest and most convenient way to determine your device's firmware version is to use the YubiKey Manager tool (ykman), a lightweight software package installable on any OS. The YubiKey Manager has both a graphical user interface (GUI) and a command line interface (CLI).

- Download the YubiKey Manager tool: <https://www.yubico.com/products/services-software/download/yubikey-manager/>
- [YubiKey Manager \(ykman\) CLI & GUI Guide](#)

The features, capabilities, and enhancements brought to the YubiKey 5 Series by the various firmware versions are **summarized** below, with the full details given in the technical description sections in this manual.

3.1 Overview of Capabilities

3.1.1 YubiKey 5 Series

Table 1: Features and Form Factors by Firmware Version

Firmware Version	5.0.x	5.1.x	5.2.x	5.3.x	5.4.x
Serial Number	Yes	Yes	Yes	Yes	Yes
OTP	Yes	Yes	Yes	Yes	Yes
OATH	Yes	Yes	Yes	Yes	Yes
OpenPGP version	2.1	2.1	3.4	3.4	3.4
PIV/Smart Card	Yes	Yes	Yes	Yes	Yes
FIDO U2F	Yes	Yes	Yes	Yes	Yes
FIDO2/WebAuthn	Yes	Yes	Yes	Yes	Yes
YubiHSM Auth					Yes
SCP03				Yes	Yes
USB-A	Yes	Yes	Yes	Yes	Yes
USB-A + NFC	Yes	Yes	Yes	Yes	Yes
USB-C	Yes	Yes	Yes	Yes	Yes
USB-C + NFC		Yes	Yes	Yes	Yes
USB-A Nano	Yes	Yes	Yes	Yes	Yes
USB-C Nano	Yes	Yes	Yes	Yes	Yes
Lightning + USB-C			Yes	Yes	Yes

3.1.2 YubiKey 5 FIPS Series

Table 2: Features and Form Factors by Firmware Version

Firmware Version	5.4.2	5.4.3
Serial Number	Yes	Yes
OTP	Yes	Yes
OATH	Yes	Yes
OpenPGP version		3.4
PIV/Smart Card	Yes	Yes
FIDO U2F	Yes	Yes
FIDO2/WebAuthn	Yes	Yes
YubiHSM Auth		Yes
SCP03	Yes	Yes
USB-A	Yes	Yes
USB-A + NFC	Yes	Yes
USB-C	Yes	Yes
USB-C + NFC	Yes	Yes
USB-A Nano	Yes	Yes
USB-C Nano	Yes	Yes
Lightning + USB-C	Yes	Yes

3.1.3 YubiKey 5 CSPN Series

Table 3: Features and Form Factors by Firmware Version

Firmware Version	5.4.2
Serial Number	Yes
OTP	Yes
OATH	Yes
OpenPGP version	
PIV/Smart Card	Yes
FIDO U2F	Yes
FIDO2/WebAuthn	Yes
YubiHSM Auth	
SCP03	Yes
USB-A	Yes
USB-A + NFC	Yes
USB-C	Yes
USB-C + NFC	Yes
USB-A Nano	Yes
USB-C Nano	Yes
Lightning + USB-C	Yes

3.1.4 YubiKey Bio Series

Table 4: Features and Form Factors by Firmware Version

Firmware Version	5.5.x
Serial Number	Yes
OTP	
OATH	
OpenPGP version	
PIV/Smart Card	
FIDO U2F	Yes
FIDO2/WebAuthn	Yes
YubiHSM Auth	
SCP03	
USB-A	Yes
USB-A + NFC	
USB-C	Yes
USB-C + NFC	
USB-A Nano	
USB-C Nano	
Lightning + USB-C	

3.1.5 Security Key Series

Table 5: Features and Form Factors by Firmware Version

Firmware Version	5.0.x - 5.2.x	5.4.x	5.4.x Enterprise Edition
Serial Number			Yes
OTP			
OATH			
OpenPGP version			
PIV/Smart Card			
FIDO U2F	Yes	Yes	Yes
FIDO2/WebAuthn	Yes	Yes	Yes
YubiHSM Auth			
SCP03			
USB-A	Yes		
USB-A + NFC	Yes	Yes	Yes
USB-C			
USB-C + NFC		Yes	Yes
USB-A Nano			
USB-C Nano			
Lightning + USB-C			

3.2 Secure Channel

Secure channel is used for establishing an authenticated and encrypted communication channel over CCID between a host and the secure element on the YubiKey. The YubiKey security domain can store three concurrent long-lived transport key sets.

SCP03 ([Secure Channel Protocol 03](#)), which is part of the GlobalPlatform standard, is a framework for mutual authentication and encrypted transport between hosts and secure elements in smart cards. With the 5.4.X firmware, Yubico has implemented a secure channel based on that specification.

For an **overview** of this implementation, see [Secure Channel](#), while **detailed descriptions** of the secure channel feature are to be found in `yk5-secure-channel-tech-desc-label`, [Yubico Secure Channel Key Diversification and Programming](#), and [Yubico SCP03 Developer Guidance](#).

Note: Applications based on PKCS #11 or Microsoft CNG do not usually use the secure channel.

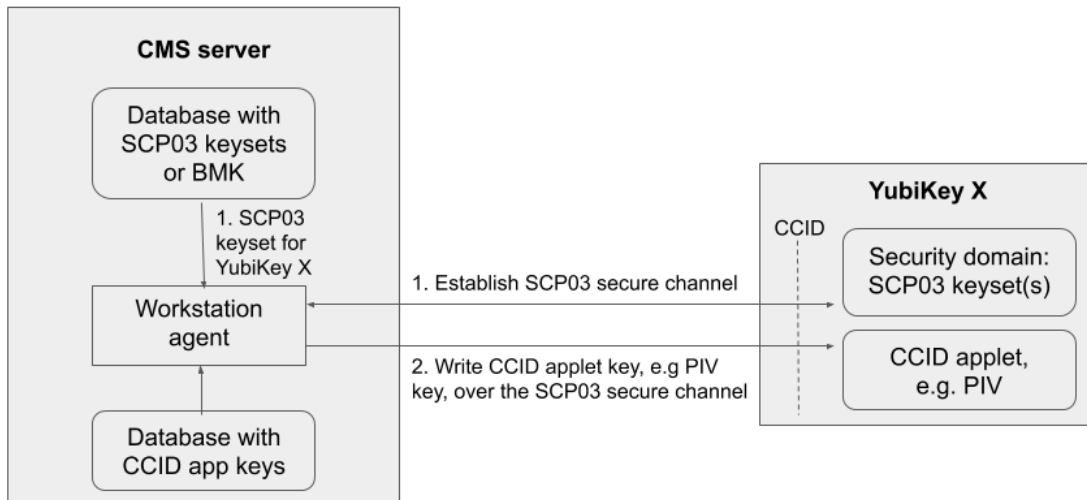
3.2.1 Security Domains & Key Diversification

The authenticated and encrypted communication channel takes place over the CCID interface between a host and the secure element on the YubiKey. This includes programming or communication from CCID functions. The secure channel feature can therefore be used to load application keys onto the YubiKey to be used with the CCID applications OATH, OpenPGP, or PIV.

Writing CCID Application Keys over SCP03

The YubiKey security domain can store three concurrent transport key sets. A transport key set contains three long-lived AES keys. When a session is established, the session AES keys are derived from the long-lived transport key set.

Writing CCID application keys over SCP03



Key diversification is the process of deriving a secure channel static transport key set from a Batch Master Key (BMK), the YubiKey identifier (part of serial number), and additional metadata. Key diversification therefore facilitates secure distribution of key sets over a secure channel. To derive the YubiKey transport key sets, the Batch Master Key (BMK) is shared with the CMS system. If the CMS vendor gives Yubico access to its BMK, Yubico can preprogram the secure channel transport key sets for the YubiKey 5 batches. The BMK could be protected by the YubiHSM2.

In order to import new transport key sets, a secure channel must be established with the security domain. This has to be done with a previously loaded transport key set or the default transport key set. Each secure channel transport key set is protected by being written to the YubiKey security domain in the secure element and stored in a server, typically a CMS system. The host that is accessing the YubiKey has an agent that connects to the CMS system to retrieve the secure channel key set. Based on the secure channel key set, both at the host and the YubiKey, a secure session is established.

Establish SCP03 Secure Channel

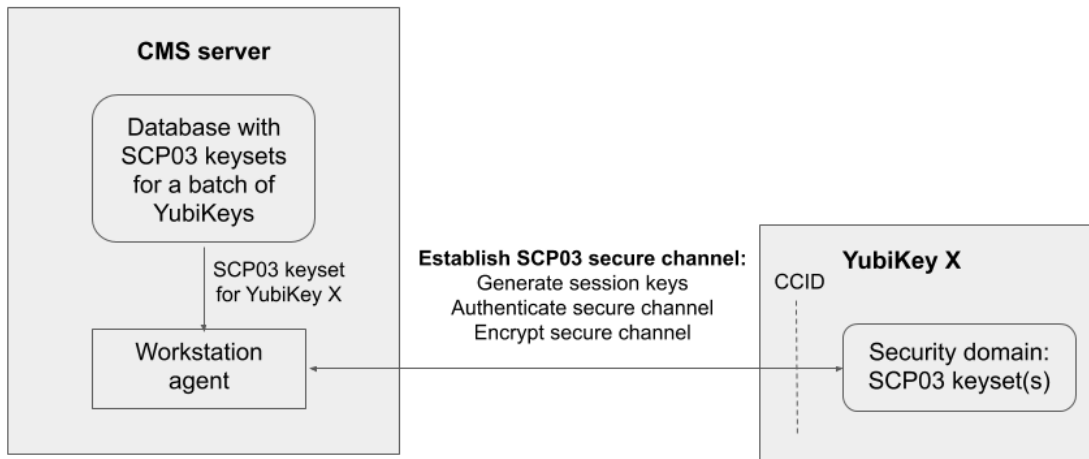
3.2.2 Benefits and Usage

- Encryption application keys can be stored on the CMS server as well as on the YubiKey. If the YubiKey is lost or compromised, the encryption key can be recovered and loaded onto a replacement YubiKey.
- Key diversification enables simplified and secured distribution of secure channel transport key sets as only the BMK must be shared with the CMS system to derive the YubiKey transport key sets.
- The secure channel transport key sets can be preprogrammed at the YubiKey batches by Yubico, if the Yubico supply chain has access to the BMK of the CMS vendor.
- The CMS system can generate the secure channel transport key sets based on the YubiKey serial numbers, the BMK, and additional metadata. The CMS can then use the initial secure channel transport key set for writing additional secure channel transport key sets to the YubiKeys.

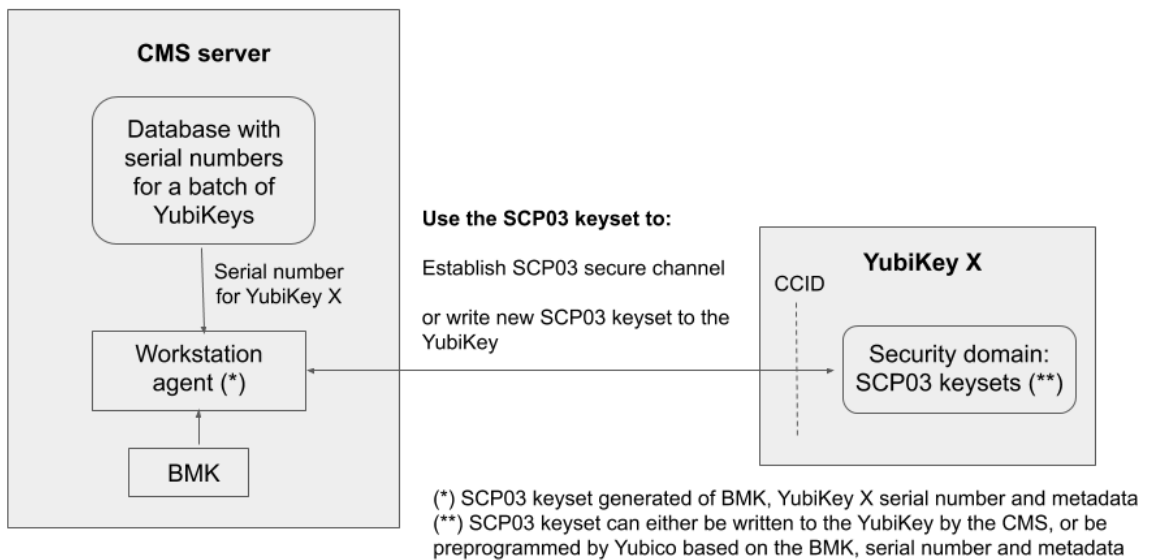
SCP03 Key Diversification

For more technical information, see *Yubico Secure Channel Key Diversification and Programming*.

Establish SCP03 secure channel



SCP03 key diversification



3.2.3 Secure Channel CPLC Data

The Card Production Life Cycle (CPLC) data object is a random dataset that is stored on each YubiKey to assure unique identification of the YubiKeys in CMS. Although it is officially deprecated from the SCP03 standard, it is still widely used to hold card data specific to CMS services or to uniquely identify smart cards. Therefore Yubico has implemented the CPLC data object to provide unique identification of YubiKeys for CMS vendors.

For a more detailed description of CPLC data object, see *Secure Channel CPLC Data*.

3.3 PIV Enhancements

3.3.1 YubiKey PIV Metadata

YubiKey 5 PIV metadata enables services and client software to obtain information about PIV keys from a central location, so that it is no longer necessary to query PIV attestation. This enables the YubiKey PIV application to report on characteristics of cryptographic keys in the specified PIV slot. YubiKey PIV metadata thereby facilitates integration with CMS vendors.

PIV metadata was introduced with the YubiKey 5.3.0 firmware. For details, see the [Get Metadata section of the PIV extensions](#) on developers.yubico.com.

3.3.2 PIV Management Key (AES)

Prior to the release of the 5.4.2 firmware, the PIV management key was a **3DES** key. This feature expands the management key type held in PIV slot 9b to include AES keys (128, 192 and 256) as defined in [SP 800-78-4 Cryptographic Algorithms and Key Sizes for Personal Identity Verification <SP800-78-4, section 5](#)). The PIV management key in AES format enables current and future FIPS-compliant CMS services.

For additional technical information, see *PIV AES Management Key*.

3.4 NFC ID: Calculation Changed

Crucial to vendors of physical access control systems and door protection systems utilizing NFC readers, the modification of the YubiKey NFC ID calculation enables NFC readers and access management systems (door locks) using the NFC ID tag to identify NFC-enabled YubiKeys, including those without serial numbers. It is now calculated so that a unique string is returned in the first part of the NFC ID. Until the release of the 5.4.2 firmware, the fact that some access control systems truncate the YubiKey NFC ID meant that YubiKey 5 NFC IDs appeared to be non-unique.

For more technical information on this, see *NFC ID Calculation Technical Description*.

3.5 YubiHSM Auth

YubiHSM Auth is a YubiKey CCID application that stores the long-lived credentials used to establish secure sessions to a YubiHSM 2. The secure session protocol is based on Secure Channel Protocol 3 (SCP03). YubiHSM Auth is supported by YubiKey firmware version 5.4.3.

For more details, see *YubiHSM Auth*.

YubiHSM Auth is a CCID application that can store long-lived credentials (AES keys) that are used to establish secure sessions to a YubiHSM 2. By providing an external challenge, a derivation scheme that yields three session keys is executed. The session keys are not stored on the YubiKey but simply output as a result. The session keys can be used

for authentication to the YubiHSM 2. The authentication scheme is based on SCP03 (see *Secure Channel* above). Each long-lived YubiHSM Auth credential is protected by a user access code that has to be provided to authenticate each session. Storing and deleting credentials requires a separate admin access code.

3.5.1 Benefits and Usage

YubiHSM Auth enables the secure storage of the long-lived credentials for accessing a YubiHSM 2. The existing authentication solution for the YubiHSM 2 is based on software credentials derived from the Password-Based Key Derivation Function 2 (PBKDF2) algorithm with a password as input.

Generating keys using PBKDF2 is just for convenience. It is preferable - and recommended - to provide AES keys directly to avoid exposing them to attack. Not only is it important to avoid losing the derivation password or the keys themselves (as those are basically the same thing), but those credentials also

- Exist outside a secure element and
- Need to be given in clear text to the program that uses them loads them into memory.

With YubiHSM Auth only the ephemeral session keys exist outside a secure environment.

For more details, see *YubiHSM Auth*.

To get in touch with Yubico Support, [click here](#).

PHYSICAL ATTRIBUTES

The serial number is printed on the back of every YubiKey in the YubiKey 5 Series, YubiKey 5 FIPS Series, YubiKey 5 CSPN Series and YubiKey Bio Series. The 2D barcode (QR code) of the serial number is also printed on the back of every YubiKey in the 5 (FIPS/CSPN) Series except the 5C Nano form factor, which is too small to accommodate the 2D barcode.

Additionally, all of the keys in the YubiKey 5 FIPS Series have the acronym “FIPS” underneath the QR code on the back, along with “v5” running up the left side of the QR code, except on the YubiKey 5C Nano, where it runs down the right side.

All current YubiKeys have been [IP68-rated](#) under the IEC standard 60529.

4.1 YubiKey 5 NFC

Important: The attributes listed below also apply to the YubiKey 5 NFC FIPS, YubiKey 5 NFC CSPN, Security Key NFC, and the Security Key NFC - Enterprise Edition.



- Dimensions: 18mm x 45mm x 3.3mm
- Weight: 3g
- Physical Interfaces: USB, NFC
- Operating Temperatures: 0 °C - 40 °C (32 °F - 104 °F)
- Storage Temperatures: -20 °C - 85 °C (-4 °F - 185 °F)

4.2 YubiKey 5 Nano

Important: The attributes listed below also apply to the YubiKey 5 Nano FIPS, and the YubiKey 5 Nano CSPN.



- Dimensions: 12mm x 13mm x 3.1mm
- Weight: 1g
- Physical Interfaces: USB
- Operating Temperatures: 0 °C - 40 °C (32 °F - 104 °F)
- Storage Temperatures: -20 °C - 85 °C (-4 °F - 185 °F)

4.3 YubiKey 5C

Important: The attributes listed below also apply to the YubiKey 5C FIPS and the YubiKey 5C CSPN.



- Dimensions: 12.5mm x 29.5mm x 5mm
- Weight: 2g
- Physical Interfaces: USB
- Operating Temperatures: 0 °C - 40 °C (32 °F - 104 °F)
- Storage Temperatures: -20 °C - 85 °C (-4 °F - 185 °F)

4.4 YubiKey 5C Nano

Important: The attributes listed below also apply to the YubiKey 5C Nano FIPS and the YubiKey 5C CSPN.



- Dimensions: 12mm x 10.1mm x 7mm
- Weight: 1g
- Physical Interfaces: USB
- Operating Temperatures: 0 °C - 40 °C (32 °F - 104 °F)
- Storage Temperatures: -20 °C - 85 °C (-4 °F - 185 °F)

4.5 YubiKey 5Ci

Important: The attributes listed below also apply to the YubiKey 5Ci FIPS and the YubiKey 5Ci CSPN.



- Dimensions: 12mm x 40.3mm x 5mm
- Weight: 2.9g
- Physical Interfaces: USB, Lightning®
- Operating Temperatures: 0 °C - 40 °C (32 °F - 104 °F)
- Storage Temperatures: -20 °C - 85 °C (-4 °F - 185 °F)

4.6 YubiKey 5C NFC

Important: The attributes listed below also apply to the YubiKey 5C FIPS, YubiKey 5C NFC CSPN, Security Key C NFC, and the Security Key C NFC - Enterprise Edition.



- Dimensions: 18mm x 45mm x 3.7mm
- Weight: 4g
- Physical Interfaces: USB, NFC
- Operating Temperatures: 0 °C - 40 °C (32 °F - 104 °F)
- Storage Temperatures: -20 °C - 85 °C (-4 °F - 185 °F)

4.7 YubiKey Bio Series

The YubiKey Bio - FIDO Edition is available in the USB-A format, while the YubiKey C Bio - FIDO Edition is available in the USB-C format, both with a maximum transfer rate of 12 Mbps.

The sensors and LEDs behave the same way in both formats.



YubiKey Bio - FIDO Edition



YubiKey C Bio - FIDO Edition

4.7.1 Sensors

The YubiKey Bio recognizes **two interactions**, one a **touch**, and the other a **fingerprint**. Its recognition of the fingerprint - or lack thereof - is communicated through the LEDs (see [LED Behavior](#)).

On the YubiKey Bio, the silver-colored bezel encircling the fingerprint sensor provides the grounding plane required to read the fingerprint.

Biometric Touch

When prompted to have the YubiKey Bio read your fingerprint from the fingerprint sensor, be sure to touch at least a tiny part of the ring. If you use your little finger to touch only the center of the fingerprint sensor, the key will not read the fingerprint.

Plain Touch

When prompted to touch the YubiKey Bio but not explicitly asked for the fingerprint, touch **both** the bezel and the fingerprint sensor, even though the fingerprint will not be read.

Tips provides detailed instructions on using the fingerprint sensor.

4.7.2 LEDs

The YubiKey Bio has a green LED and an amber LED to provide direct feedback, flashing when the key is ready for interaction or communicating something about the interaction. *LED Behavior* provides detailed descriptions.

4.7.3 Ratings

The YubiKey Bio has been IP68-rated under the IEC standard 60529.

4.7.4 Care and Cleaning

To clean the YubiKey and sensor, use only wipes impregnated with no more than 70% isopropyl alcohol.

4.7.5 Operational Data

- Dimensions
 - YubiKey A Bio: 18mm x 45mm x 3.35mm
 - YubiKey C Bio: 18mm x 45mm x 3.75mm
- Weight: YubiKey Bio A: 4.5g; YubiKey Bio C: 5.0g
- Operating Temperatures: 0 °C - 40 °C (32 °F - 104 °F)
- Storage Temperatures: -20 °C - 85 °C (-4 °F - 185 °F)

4.8 FIPS-specific Marking



4.9 CSPN-specific Marking



4.10 Security Key Series Marking

The color of the *non-Enterprise* edition of the Security Key Series was changed to black in January 2023. To distinguish the Security Key Series from the YubiKey 5 Series, “FIDO” is inscribed on the back.



Like the non-Enterprise edition of the Security Key Series, the *Enterprise* edition of the Security Key Series also has “FIDO” inscribed on the back to distinguish them from a YubiKey 5 Series. In addition, the serial number is inscribed on the back of each key in the *Enterprise* edition of the Security Key Series.



To get in touch with Yubico Support, [click here](#).

PHYSICAL INTERFACES: USB, NFC, APPLE LIGHTNING®

We refer to the ways that a computer, phone, tablet, etc. can connect with a YubiKey as the physical interfaces.

5.1 USB

All of the models in the YubiKey 5 (FIPS/CSPN) Series provide a USB 2.0 interface, regardless of the form factor of the USB connector. The YubiKey will present itself as a USB composite device in addition to each individual USB interface.

USB A and USB C connectors are supported.

The USB PID and iProduct string changes depending on which of the USB interfaces are enabled. They are described in the [YubiKey USB ID Values Guide](#).

For more information, see “Understanding the USB Interfaces” in the YubiKey 5 Series Technical Manual.

5.2 Apple Lightning®

The YubiKey 5Ci FIPS presents itself as an Apple iOS peripheral. It is able to interact with:

- Any iOS app utilizing the Yubico YubiKey iOS SDK
- Any app input data field via touch-triggered OTP.
- Any WebAuthn compliant application (starting in iOS 13). This includes the Safari browser.

All features of the YubiKey 5 FIPS are supported over Lightning®.

When connecting the YubiKey 5Ci FIPS via Lightning®, the **interfaces enabled** setting is common to both USB-C and Lightning®. Enabling or disabling an interface will apply to both connections.

The YubiKey 5Ci FIPS communication over Lightning® uses a variety of channels for communication between iOS and the YubiKey.

Note: Developers: for apps within iOS to be able to use advanced protocols that send and receive information from the YubiKey 5Ci FIPS, the [Yubico iOS SDK](#) is required and the app registered with Yubico. This can be done via the [Yubico iOS SDK App Submission page](#).

For a description of the USB and iProduct string when connecting via Lightning®, see the [YubiKey USB ID Values Guide](#).

5.3 NFC

In addition to USB, the YubiKey 5 NFC FIPS and YubiKey 5C NFC FIPS also provide an NFC wireless interface. The YubiKey 5 NFC FIPS and YubiKey 5C NFC FIPS include the RFID standard specific to the ISO/IEC 14443-A and ISO/IEC 14443-4 NFC format; RFID implementations not included in the listed ISO standards are not supported.

The NDEF URI has been updated to a new format; an example of the new format is provided below. The <OTP> value is replaced with the OTP generated by the YubiKey.

<https://demo.yubico.com/yk/>

For operations that require a touch, all touch requests within the first 20 seconds of the operation will succeed. After a period of inactivity, a YubiKey placed on a desktop NFC reader may power down to help prevent unintended access. To regain connectivity with an NFC reader, remove the YubiKey from the reader and reposition it on the reader. Some NFC readers may power-cycle and in doing so, prevent the YubiKey from powering down.

To get in touch with Yubico Support, [click here](#).

UNDERSTANDING THE USB INTERFACES

USB interfaces are the different channels that software can use to communicate with the YubiKey when it is connected via USB. Each interface enables a specific set of applications on the YubiKey; if an interface is disabled, none of the applications that use that interface will be available.

Note: With previous YubiKeys and older Yubico firmware, the USB interfaces were referred to as “modes” and the FIDO interface was called the “U2F mode”.

6.1 OTP

The OTP interface presents itself to the operating system as a USB keyboard. The OTP application is accessible over this interface. Output is sent as a series of keystrokes from a virtual keyboard. This allows for OTP to be used in any environment which can accept standard keyboard input.

The OTP interface is supported natively across all desktop OS environments (macOS, Windows, Linux) as well as on mobile OS platforms (iOS, Android). Output is sent as a series of keystrokes from a virtual keyboard, allowing the OTP application to work with any environment that supports USB keyboard input.

6.2 FIDO

The FIDO interface provides access to the [FIDO2](#) and [U2F](#) applications.

The FIDO interface presents itself as a generic human interface device (HID). The FIDO interface is supported on all desktop platforms running WebAuthn-compatible browsers or applications, as well as on Android and iOS (starting in iOS 13).

6.3 CCID

The CCID interface provides communication for the [PIV / Smart Card](#), [OATH \(HOTP and TOTP\)](#), and [OpenPGP](#) applications.

The YubiKey presents itself to the operating system as a USB smart card reader. Each of the applications presents itself as a separate smart card attached to that reader. The CCID interface is supported on Windows and MacOS, and on Linux with the [PC/SC](#) package. CCID is also supported on Android.

Note: Developers: to access the CCID interface on iOS, the Yubico iOS SDK is required.

To get in touch with Yubico Support, [click here](#).

PROTOCOLS AND APPLICATIONS

The YubiKey 5 Series provides applications for FIDO2, OATH, OpenPGP, OTP, Smart Card, U2F. The applications are all separate from each other, with separate storage for keys and credentials.

For information on managing all these applications, see *Tools and Troubleshooting*.

Note that the OTP and OATH categories overlap; technically, there are three true OTPs:

- Yubico OTP (defined by Yubico)
- OATH-HOTP (standard [RFC4226](#))
- OATH-TOTP (standard [RFC6238](#))
- We support the **Yubico OTP and OATH-HOTP directly on the touch-triggered OTP function** on the YubiKey.
- We support **OATH-HOTP and OATH-TOTP directly on the OATH function** on the YubiKey (usually called OATH and used with Yubico authenticator).
- **Touch-triggered OTP** also supports a static password and Challenge-Response. Challenge-Response can also be used with software (such as Yubico Authenticator) to act as a single OATH-TOTP credential.

All three of these OTPs are described in more detail below, under *OATH* and under *OTP*.

7.1 FIDO2

The **FIDO2** standard offers the same high level of security as FIDO U2F, since it is based on public key cryptography. In addition to providing phishing-resistant two-factor authentication, the FIDO2 application on the YubiKey allows for the storage of resident credentials, also called discoverable credentials. As these credentials can accommodate the username and other data, this enables truly passwordless authentication on sites and applications that support the WebAuthn protocol. YubiKeys in the 5 Series can hold up to 25 resident keys.

Note: FIDO2 support is available to the iPad Pro via the USB-C or Lightning® connectors of the *YubiKey 5Ci*. FIDO2/WebAuthn can be achieved over USB-C using any of the following options:

- `ASWebAuthenticationSession`
- `SFSafariViewController`
- Redirect to Safari browser

For more details on support for the iPad Pro, see *iPad and iPad Pro* below, and to see which U2F/FIDO2 security keys currently work with iOS/iPadOS 13.3+ devices using the Safari browser in combination with apps using

SFSafariViewController or ASWebAuthenticationSession - see [Supporting U2F or FIDO2 Security Keys on iOS or iPadOS | Security Key Compatibility](#).

7.1.1 Locking FIDO2 Credentials

The resident credentials can be left unlocked and used for strong single-factor authentication, or they can be protected by a PIN for two-factor authentication.

- The FIDO2 PIN must be between 4 and 63 characters in length.
- Once a FIDO2 PIN is set, it can be changed but it cannot be removed without resetting the FIDO2 application.
- If the PIN is entered incorrectly 8 times in a row, the FIDO2 application will be locked. In order to restore this functionality, the FIDO2 application must be reset.

Note: Resetting the FIDO2 application will also reset the U2F key, so the YubiKey must be re-registered not only with all the FIDO2 sites, but also with all U2F sites.

Note: The YubiKey 5Ci supports Credential Management to allow for selective deletion of resident keys. See the guide to the [Enhancements to FIDO 2 Support](#) for details.

7.1.2 Default Values

PIN: None set.

7.1.3 AAGUID Values

The [FIDO2 specification](#) states that an Authenticator Attestation GUID (AAGUID) must be provided during attestation. An AAGUID is a 128-bit identifier indicating the type of the authenticator.

New AAGUIDs will be issued for new YubiKey products which support FIDO2, or when existing YubiKey products have FIDO2 features added or removed.

For the complete list of AAGUIDs, see <https://support.yubico.com/hc/en-us/articles/360016648959-YubiKey-Hardware-FIDO2-AAGUIDs>.

7.1.4 Supported Extensions

The YubiKey 5 Series supports only the AppID extension (appid) as defined by the [W3C Web Authentication API specification](#). This extension allows U2F credentials registered using the legacy FIDO JavaScript APIs to be used with WebAuthn. That means that if you register a YubiKey in the 5 Series on a website that used U2F at that time and later upgrades to FIDO2, your U2F credentials will continue to work on the website.

7.2 FIDO U2F

FIDO U2F is an open standard that provides strong, phishing-resistant two-factor authentication for web services using public key cryptography. U2F does not require any special drivers or configuration to use, just a compatible web browser. The U2F application on the YubiKey can be associated with an unlimited number of U2F sites.

7.3 OATH

The OATH application can store up to 32 OATH credentials, either OATH-TOTP (time-based One-Time Password) or OATH-HOTP (counter-based One-Time Password). These credentials are separate from those stored in the OTP application, and can only be accessed via the CCID channel. In order to manage these credentials and read the OTPs generated by the YubiKey, the [Yubico Authenticator](#) is needed.

To restrict access to the OTPs, set a password for the OATH application.

Note: Developers: using the OATH application functions on iOS requires the Yubico iOS SDK.

7.3.1 HOTP and TOTP

Both **OATH-TOTP** and **OATH-HOTP** credentials are described in detail in the [OATH Overview](#).

7.4 OpenPGP

The OpenPGP application provides an OpenPGP-compatible smart card in compliance with version 3.4 of the specification if the YubiKey firmware is 5.2.3 or later. If the firmware is an earlier version, the OpenPGP-compatible smart card is in compliance with version 2.0 of the specification.

OpenPGP-compatible smart card can be used with compatible PGP software such as GnuPG (GPG) and can store one PGP key each for authentication, signing, and encryption. Similar to the PIV / Smart Card touch policy, the OpenPGP application can also be set to require the YubiKey's metal contact be touched to authorize an operation.

Note: Developers: using the OpenPGP functions on iOS requires the Yubico iOS SDK.

YubiKey firmware 5.2.3 - 5.2.8 and 5.3.2 - 5.4.3 in combination with OpenPGP 3.4:

- Extends existing RSA support for OpenPGP operations to ECC algorithms
- Provides the Yubico Attestation feature for verifying keys generated on a YubiKey device
- Utilizes separate x.509 cardholder certificates alongside the existing OpenPGP certificates for authentication, signature and encryption/decipher
- Bring attestation functionality to OpenPGP keys and certificates generated on a YubiKey
- Improves security by supporting Key Derivation Function (KDF) PINs. With KDF enabled, the PIN is stored as a hash on the YubiKey. The OpenPGP client will only pass the hashed value, never the PIN directly.

7.4.1 Elliptic Curve Cryptographic (ECC) Algorithms

The YubiKey 5.2.3 firmware added support for ECC algorithms. These can be used for Signature, Authentication and Decipher keys. The full list of curves supported by OpenPGP 3.4 can be found in section 4.4.3.10 of the [OpenPGP Smart Card 3.4 spec](#) (page 35).

In addition to [RSA Algorithms](#), YubiKeys support the following ECC algorithms:

- secp256r1
- secp256k1
- secp384r1
- secp521r1
- brainpoolP256r1
- brainpoolP384r1
- brainpoolP512r1
- curve25519
 - x25519 (decipher only)
 - ed25519 (sign / auth only)

For further details on the new features, including key attestation, expanded encryption algorithms and additional card-holder certificates, refer to [Enhancements to OpenPGP Support](#).

7.4.2 RSA Algorithms

- RSA 1024
- RSA 2048
- RSA 3072 (requires GnuPG version 2.0 or higher)
- RSA 4096 (requires GnuPG version 2.0 or higher)

7.4.3 Default Values

- PIN: 123456
- Admin PIN: 12345678

7.5 OTP

The OTP application provides two programmable slots, each of which can hold one of the types of credentials listed below. A Yubico OTP credential is programmed to slot 1 during manufacturing. Output is sent as a series of keystrokes from a virtual keyboard.

- Trigger the YubiKey to produce the credential in the first slot by briefly touching the metal contact of the YubiKey.
- If a credential has been programmed to the second slot, trigger the YubiKey to produce it by touching the contact for 3 seconds.

7.5.1 Yubico OTP

Yubico OTP is a strong authentication mechanism that is supported by the YubiKey 5 Series. Yubico OTP can be used as the second factor in a two-factor authentication (2FA) scheme or on its own, providing single-factor authentication.

The OTP generated by the YubiKey has two parts, with the first 12 characters being the public identity which a validation server can link to a user, while the remaining 32 characters are the unique passcode that is changed each time an OTP is generated.

The character representation of the Yubico OTP is designed to handle a variety of keyboard layouts. It is crucial that the same code is generated if a YubiKey is inserted into a German computer with a QWERTZ layout, a French one with an AZERTY layout, or a US one with a QWERTY layout. The “Modhex”, or Modified Hexadecimal coding, was invented by Yubico to use only specific characters to ensure that the YubiKey works with the maximum number of keyboard layouts. (USB keyboards send their keystrokes by means of “scan codes” rather than the actual character. The translation to keystrokes is done by the device to which the YubiKey is connected).

7.5.2 Static Password

A static password can be programmed to the YubiKey so that it will type the password for you when you touch the metal contact.

For managing multiple passwords, see the [password managers](#) that the YubiKey can secure with two-factor authentication (2FA).

7.5.3 HMAC-SHA1 Challenge-Response

This type of credential is most often used for offline authentication, as it does not require contacting a server for validation.

An HMAC-SHA1 Challenge-Response credential enables software to send a challenge to the YubiKey and verify that an expected, predetermined response is returned. This credential can also be set to require a touch on the metal contact before the response is sent to the requesting software. This type of credential must be activated by the software sending the challenge; it cannot be activated by touching the metal contact on the YubiKey.

Note: Developers: as the Challenge-Response function requires two-way communication with the YubiKey, using this feature on iOS requires the Yubico iOS SDK.

7.6 Smart Card (PIV Compatible)

The YubiKey 5 Series provides a PIV-compatible smart card application. PIV, or FIPS 201, is a US government standard. It enables RSA or ECC sign/encrypt operations using a private key stored on a smart card through common interfaces like PKCS#11.

On Windows, the smart card functionality can be extended with the [YubiKey Smart Card Minidriver](#).

Note: The YubiKey Smart Card Minidriver is not available for Android, Linux, macOS or iOS.

The YubiKey 5 Series supports extended APDUs, extended Answer To Reset (ATR), and Answer To Select (ATS). Using the PIV APDUs on iOS requires the Yubico iOS SDK.

7.6.1 Default Values

- PIN: 123456
- PUK: 12345678
- Management Key (3DES): 010203040506070801020304050607080102030405060708

7.6.2 Supported Algorithms

The YubiKey 5 Series supports the following algorithms on the PIV smart card application.

- RSA 1024
- RSA 2048
- ECC P-256
- ECC P-384

7.6.3 Policies

PIN Policy

To specify how often the PIN needs to be entered for access to the credential in a given slot, set a PIN policy for that slot. This policy must be set upon key generation or import; it cannot be changed later.

Touch Policy

In addition to requiring the PIN, the YubiKey can require a physical touch on the metal contact. Similar to the PIN policy, the touch policy must be set upon key generation or import.

7.6.4 Slot Information

The keys and certificates for the smart card application are stored in slots, which are described below. The PIN policies described below are the defaults, before they are overridden with a custom PIN policy. **These slots are separate from the programmable slots in the OTP application.**

Slot 9a: PIV Authentication

This certificate and its associated private key is used to authenticate the card and the cardholder. This slot is used for system login, etc. To perform any private key operations, the end user PIN is required. Once the correct PIN has been provided, multiple private key operations may be performed without additional cardholder consent.

Slot 9c: Digital Signature

This certificate and its associated private key is used for digital signatures for the purpose of document, email, file, and executable signing. To perform any private key operations, the end user PIN is required. The PIN must be submitted immediately before each sign operation to ensure cardholder participation for every digital signature generated.

Slot 9d: Key Management

This certificate and its associated private key is used for encryption to assure confidentiality. This slot is used for encrypting emails or files. The end user PIN is required to perform any private key operations. Once the correct PIN has been provided, multiple private key operations may be performed without additional cardholder consent.

Slot 9e: Card Authentication

This certificate and its associated private key is used to support additional physical access applications, such as providing physical access to buildings via PIV-enabled door locks. The end user PIN is NOT required to perform private key operations for this slot.

Slots 82-95: Retired Key Management

These slots are meant for previously used Key Management keys to be able to decrypt earlier encrypted documents or emails.

Slot f9: Attestation

This slot is only used for attestation of other keys generated on device with instruction f9. This slot is not cleared on reset, but can be overwritten.

7.6.5 Attestation

Attestation enables you to verify that a key on the smart card application was generated on the YubiKey and was not imported. An X.509 certificate for the key to be attested is created if the key has been generated on the YubiKey. Included in the certificate are the following extensions that provide information about the YubiKey.

- 1.3.6.1.4.1.41482.3.3: Firmware version, encoded as three bytes. For example, 050100 indicates firmware version 5.1.0.
- 1.3.6.1.4.1.41482.3.7: Serial number of the YubiKey, encoded as an integer.
- 1.3.6.1.4.1.41482.3.8: Two bytes, the first encoding the PIN policy and the second encoding the touch policy.
- PIN policy:
 - 01 - never require PIN
 - 02 - require PIN once per session
 - 03 - always require PIN.
- Touch policy:
 - 01 - never require touch
 - 02 - always require touch

– 03 - cache touch for 15 seconds.

- 1.3.6.1.4.1.41482.3.9: YubiKey's form factor, encoded as a one-byte octet-string.
- USB-A Keychain: 0x01
- USB-A Nano: 0x02
- USB-C Keychain: 0x03
- USB-C Nano: 0x04
- USB-C and Lightning®: 0x05
- Undefined: 0x00

7.6.6 PIV Metadata

Background: How PIV Attestation Works

A technical description of YubiKey PIV attestation is available at the [Yubico developer website](#).

Attestation is performed on a public key that has been *generated on the YubiKey*. For example, consider an asymmetric key-pair that is generated on the YubiKey with the following `ykman` (YubiKey Manager) command:

```
ykman piv generate-key 9c -
```

This command generates an asymmetric key-pair, and stores the private key in the specified slot (9c in this example). The public key that has been generated is returned as output.

The `ykman attestation` command can be executed for the key-pair at the slot (9c):

```
ykman piv attest 9c C:\Test\attestation-cert-9c.cer
```

The generated certificate is generated in real time at the YubiKey. The attestation certificate and private key, which are stored in slot f9, are used for signing the generated certificate for the slot (9c). The attestation certificate is used as template when creating the generated certificate for the slot (9c). In addition to the template attestation certificate, the [extensions and subject details](#) are appended to the generated certificate.

However, the generated certificate is not the same as the X.509 certificate that may be issued by an external CA or self-signed on the YubiKey. For example, the X.509 certificate could be issued by the Microsoft AD CS and written to the YubiKey. The YubiKey Manager GUI can be used to generate a key-pair and self-sign the public key at the YubiKey.

The public key at slot 9a can be attested (signed in real time by the CA attestation certificate) with the same `ykman` command as above:

```
ykman piv attest 9a C:\Test\attestation-cert-9a.cer
```

And the X.509 self-signed certificate can be exported from the YubiKey with the following `ykman` command:

```
ykman piv export-certificate 9a C:\Test\self-signed-9a.cer
```

The Shortcomings of PIV Attestation

PIV attestation only works for asymmetric keys that have been *generated on* the YubiKey; it does not work for asymmetric keys that have been *imported into* the YubiKey.

For example, the following `ykman` command imports a PKCS #12 file into the YubiKey at slot 9e:

```
ykman piv import-key 9e C:\\Test\\TestUser1.p12 -P 123456
```

```
ykman piv import-certificate 9e C:\\Test\\TestUser1.p12 -P 123456
```

These `ykman` commands unpack the PKCS #12 file, store the private key in the private key slot (9e), and store the X.509 certificate in the corresponding certificate slot.

Now, if one tries to attest the public key at slot 9e with the YkMan attestation command, the operation will fail:

```
ykman piv attest 9e C:\\Test\\attestation-9e.cer
```

```
Error: Attestation failed.
```

One more drawback with PIV attestation is performance, since generation of multiple PIV attestation certificates can be time-consuming.

When To Use PIV Metadata

PIV metadata should be used for the following cases:

- If PIV attestation cannot be used (for imported keys),
- If an attestation certificate is not required, PIV metadata can be used for achieving higher performance.

Yubico PIV Library and Metadata API

PIV metadata is supported by YubiKey v5.3.0 and above. YubiKey PIV metadata can be accessed by using the `libykpiv` library.

The [Yubico PIV Tool](#) contains the library

- `libykpiv.so` (for Linux),
- `libykpiv.dylib` (for MacOS),
- `libykpiv.dll` (for Windows).

The source code of the `libykpiv` library is published at the [Yubico GitHub repo](#).

libykpiv

The `libykpiv` library exposes a C API in the header file `ypiv.h`, which includes the functions `ypiv_get_metadata()` and `ypiv_util_parse_metadata()`. The source code of these functions is available in the file `ypiv.c`.

In particular, the function `ypiv_get_metadata()` calls the underlying function `_ypiv_transfer_data()`, which transfers APDUs to the YubiKey PIV applet over the CCID interface.

The function `_ypiv_transfer_data()` takes the input parameter `templ`, which is populated with the APDUs (*CLA*, *INS*, *P1*, *P2*) that are specified at the Yubico developer website for [PIV extensions](#) under the section **GET METADATA**. The YubiKey returns the tag length values (TLVs) (*Algorithm*, *Policy*, *Origin*, etc) that are specified in the same [section](#), and the TLV-encoded output is returned in the `ypiv_get_metadata()` parameter `data`.

Table 1: TLVs Returned

Key	TLV	Description
Algorithm	0x01	Algorithm/type of the key
Policy	0x02	PIN and Touch policy of the key (keys only)
Origin	0x03	Origin of the key: imported or generated
Public key	0x04	Public key associated with the private key
Default value	0x05	Whether the PIN/key has a default value PIN and PUK and Mgmt key only)
Retries	0x06	Number of retries left (PIN and PUK only)

It is even possible to invoke the function `ykpriv_transfer_data()` directly for low-level APDU communication with the YubiKey’s PIV applet.

The function `ykpriv_util_parse_metadata()` can be used for parsing the returned TLV-encoded object.

Therefore, the developer can integrate the `libykpiv` library for low level programming with YubiKey PIV metadata.

Using PIV Metadata with YKCS11

The `YKCS11` library is also part of the Yubico PIV Tool. `YKCS11` is a `PKCS#11` module that allows external applications to communicate with the PIV application running on a YubiKey.

When the `PKCS #11` function `C_OpenSession()` is called for a YubiKey `PKCS #11` slot (which is a YubiKey PIV application in a PC/SC reader), then the `YKCS11` library will parse out the public keys for all PIV key slots. If PIV attestation is supported, the PIV attestation certificate will be used for parsing out the public key.

If PIV attestation is not supported, i.e., if the key-pair has been imported into a YubiKey, then the `YKCS11` library calls the functions `ykpriv_get_metadata()` and `ykpriv_util_parse_metadata()` to parse out the requested public key.

If both attestation and PIV metadata fail, in that order, `YKCS11` will fall back to parse the public key from the X.509 certificate.

Note: The X.509 certificate’s public key may not match the private key in the YubiKey PIV slot.

7.6.7 Changes

Answer to Reset (ATR) and Answer to Select (ATS)

The ATR has been changed from “Yubikey 4” to “YubiKey” and adds support for ATS.

PIV Attestation Root CA

YubiKeys in the 5 Series have a PIV attestation root certificate authority different from the one previous YubiKeys had. You can download the certificate of the new root certificate authority on the [PIV attestation](#) page.

Easier Identification

The YubiKey 5 Series devices can report their form factor via the PIV application whether or not they have an NFC interface. This enables easier, programmatic identification of the physical attributes of the YubiKey. For more information about how to query this information, see the [YubiKey 5 Series Configuration Reference Guide](#).

PIV AES Management Key

Historically, the YubiKey PIV management key is a 3DES key. With the release of the YubiKey firmware version 5.4.2, the YubiKey PIV management key can also be an AES key. For more details, see the article on our Developer site, [YubiKey and PIV](#).

Technically speaking, this feature expands the management key type held in PIV slot 9b to include AES keys (128, 192 and 256) as defined in the PIV specification (SP800-78-4, section 5). PIV management key in AES format renders the YubiKey compatible with current or future FIPS-compliant CMS services.

7.7 YubiHSM Auth

7.7.1 Introduction

YubiHSM Auth is a YubiKey CCID application that stores the long-lived credentials used to establish secure sessions to a YubiHSM 2. The secure session protocol is based on Secure Channel Protocol 3 (SCP03). YubiHSM Auth is supported by YubiKey firmware version 5.4.3.

YubiHSM Auth uses hardware to protect the long-lived credentials for accessing a YubiHSM 2. This increases the security of the authentication credentials, as compared to the authentication solution for the YubiHSM 2 based on software credentials derived from the Password-Based Key Derivation Function 2 (PBKDF2) algorithm with a password as input.

7.7.2 Credentials and PIN Codes

Each YubiHSM Auth credential is comprised of two AES-128 keys which are used to derive the three session-specific AES-128 keys. The YubiHSM Auth application can store up to 32 YubiHSM Auth credentials in the YubiKey.

Each YubiHSM Auth credential is protected by a 16-byte user access code provided to the YubiKey for each YubiHSM Auth operation. The access code is used to access the YubiHSM Auth Credential to derive the session-specific AES-128 keys.

Storing or deleting YubiHSM Auth credentials requires a separate 16-byte admin access code.

Each access code has a limit of eight retries and optionally, verification of user presence (touch).

7.7.3 YubiHSM 2 Secure Channel

Use the YubiKey YubiHSM Auth application to establish an encrypted and authenticated session to a YubiHSM 2. Although the YubiHSM 2 secure channel is based on the protocol Global Platform Secure Channel Protocol '03' (SCP03), there are two important differences:

- The YubiHSM 2 secure channel protocol does not use APDUs, so the commands and possible options are not those of the complete SCP03 specification.
- SCP03 uses key sets with three long-lived AES keys, while the YubiHSM 2 secure channel uses key sets with two long-lived AES keys.

The YubiHSM 2 authentication protocol uses a set of static credentials called a long-lived key set. This consists of two AES-128 keys:

- ENC: Used for deriving keys for command and response encryption, as specified in SCP03.
- MAC: Used for deriving keys for command and response authentication, as specified in SCP03.

The identical long-lived keyset is protected in the YubiHSM 2 and in the YubiKey YubiHSM Auth application.

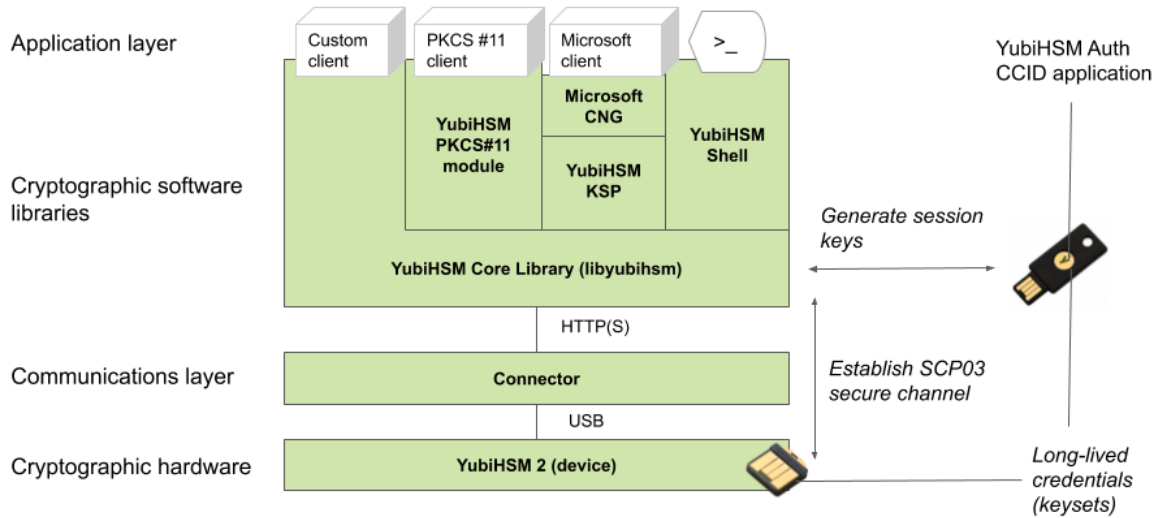
Those long-lived key sets are used by the YubiHSM Auth application to derive a set of three session-specific AES-128 keys using the challenge-response protocol as defined in SCP03:

- Session Secure Channel Encryption Key (S-ENC): Used for data confidentiality.
- Secure Channel Message Authentication Code Key for Command (S-MAC): Used for data and protocol integrity.
- Secure Channel Message Authentication Code Key for Response (S-RMAC): Used for data and protocol integrity.

The YubiHSM Auth session-specific keys are output from the YubiKey to the calling library, which uses the session keys to encrypt and authenticate commands and responses during a single session. The session keys are discarded afterwards.

7.7.4 Architecture Overview

The figure below shows how the YubiHSM Auth application fits in to the YubiHSM 2 architecture.



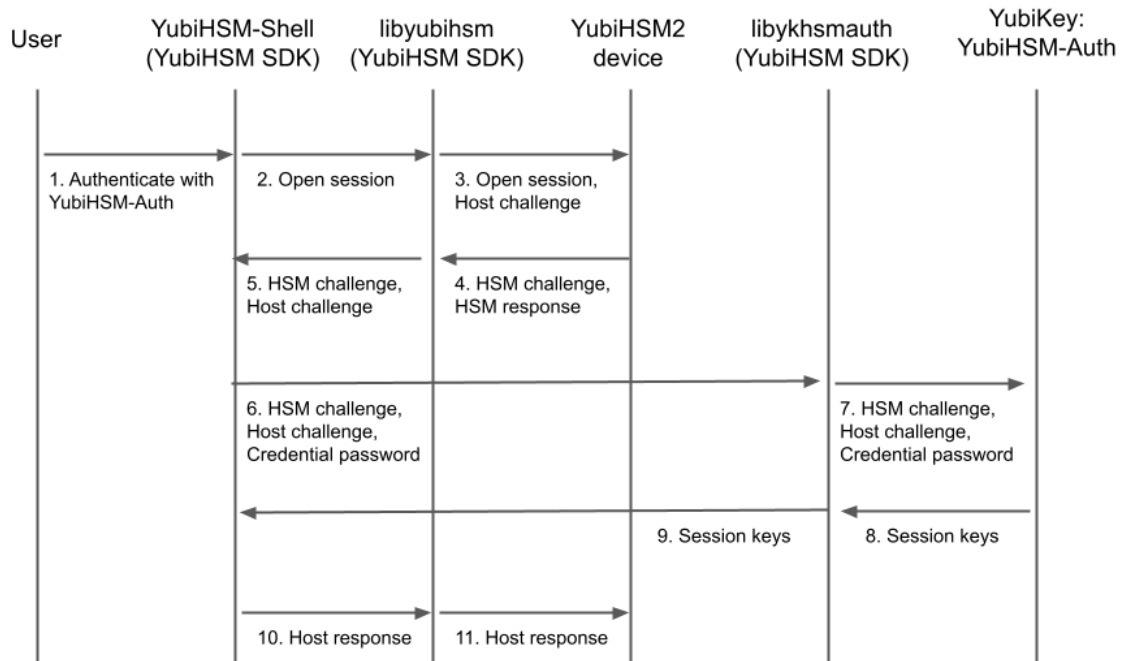
The identical long-lived credentials (key sets) are protected in both the YubiKey YubiHSM Auth application and in the YubiHSM 2. The YubiHSM-Shell software tool can be used for generating the key sets in the YubiHSM 2, and the YubiHSM-Auth software tool can be used for importing the same key sets to the YubiKey YubiHSM Auth application.

At the client, the YubiHSM authentication protocol is implemented in the `libykhsmauth` library, which derives the three session AES-keys by calling the YubiKey YubiHSM Auth CCID application. The session objects that are created can be used by the `libyubihsm` in the communication with YubiHSM.

The YubiHSM session keys are therefore generated on the basis of the long-lived credentials that are protected in the YubiHSM 2 and YubiKey YubiHSM Auth in conjunction with the SCP03 derivation scheme.

7.7.5 YubiHSM Auth Flowchart

The flowchart below illustrates the authentication protocol communication with YubiHSM using the static keys on YubiHSM Auth. It is assumed that the YubiHSM and YubiHSM Auth application share the same static keyset. The steps are explained below.



1. The user launches YubiHSM-Shell and enters the commands `connect` and `session open`, with the flag `ykopen` that indicates that the YubiKey with YubiHSM Auth shall be used.
2. The YubiHSM-Shell invokes the `libyubihsm` library, with a request to open a session to the YubiHSM 2.
3. The `libyubihsm` library generates a host challenge, and opens a session to the YubiHSM 2 device.
4. The YubiHSM 2 device generates an HSM challenge, and generates the session keys based on the HSM challenge, the host challenge, and the static key set in the YubiHSM 2 device. The YubiHSM 2 returns the HSM challenge in an HSM response to the `libyubihsm` library.
5. The `libyubihsm` library propagates the host challenge and HSM challenge to the YubiHSM Shell.
6. The user enters the Credential password for unlocking the static keyset in the YubiHSM Auth application in the YubiKey. The YubiHSM Shell invokes the `libykhsmauth` library, with a request to generate session keys.
7. The `libykhsmauth` library invokes the YubiHSM Auth application in the YubiKey with the Credential password, the HSM challenge and host challenge are used as input parameters.
8. The Credential password unlocks the static keyset in the YubiHSM Auth application, and the YubiHSM Auth application generates the session keys based on the static keys, HSM challenge, and host challenge.
9. The `libykhsmauth` library returns the session keys to YubiHSM Shell.
10. The YubiHSM Shell acknowledges the protocol handshake to `libyubihsm`.
11. The `libyubihsm` sends the host response to the YubiHSM 2 device. The session keys can now be used for secure channel communication between YubiHSM-Shell/libyubihsm in the host and the YubiHSM device.

7.7.6 Software and Tools

YubiHSM-Auth Software Tool

The YubiHSM-Auth software tool is part of the [YubiHSM Shell](#), which is installed with the [YubiHSM SDK](#). YubiHSM-Auth tool can be used for:

- Storing the YubiHSM Auth credentials on a YubiKey
- Deleting the YubiHSM Auth credentials on a YubiKey
- Listing the YubiHSM Auth credentials on a YubiKey
- Changing the YubiHSM Auth management key on a YubiKey
- Checking the number of retries of the YubiHSM Auth credential password
- Checking the version of the YubiHSM Auth application
- Calculating session keys, mainly for debugging and test purposes
- Resetting the YubiHSM Auth application on a YubiKey

First, the YubiHSM 2 device needs to be configured with an authentication key. The default authentication key password on KeyID=1 is set to “password”, and this should be changed or replaced with other authentication keys. For the examples in this section, however, it is assumed that the default authentication key is still present on the YubiHSM 2.

In order to generate and store the equivalent YubiHSM Auth credentials on the YubiKey, the `yubihsm-auth` command line tool can be used. To invoke YubiHSM-Auth simply run `yubihsm-auth` with the required commands and parameters.

To get a list of available commands, parameters and their syntax, run: `yubihsm-auth --help`.

An example of how to use `yubihsm-auth` for storing YubiHSM Auth credentials on a YubiKey is shown below:

```
$ yubihsm-auth -a put --label="default key" --derivation-password="password" --credpwd=
↪"MyPassword" --touch=on --mgmkey="00000000000000000000000000000000" --verbose=5
Credential successfully stored
```

Where:

- `-a put` is the action to insert a YubiHSM Auth credential on the YubiKey
- `--label` is the label of the YubiHSM Auth credential on the YubiKey
- `--derivation-password` is used as input to the PBKDF2 algorithm, which is used for generating the two AES-128 keys that constitute the YubiHSM Auth credentials to be stored on the YubiKey
- `--credpwd` is the password protecting the YubiHSM Auth credentials on the YubiKey
- `--touch` is set to ‘on’, which requires the user to touch the YubiKey when accessing the YubiHSM Auth credential
- `--mgmkey` is the management key that is needed for writing the YubiHSM Auth credentials on the YubiKey
- `--verbose` is used to print more information as output

Note: We recommend using an offline air-gapped computer when storing the YubiHSM Auth credentials on the YubiKey.

Now, the YubiKey YubiHSM Auth application can be used with [YubiHSM Shell](#) for authentication to the YubiHSM 2.

Using YubiHSM-Auth with YubiHSM Shell

It is now possible to authenticate to the YubiHSM 2 device with static credentials that are protected in the YubiKey application called YubiHSM Auth. For more information on this YubiKey feature and how to configure it, see Using YubiHSM Auth.

The YubiHSM Shell tool supports authentication with YubiHSM Auth credentials in both interactive mode and command-line mode.

In order to use `yubihsm-shell` with the YubiHSM Auth-enabled YubiKey in interactive mode, open a session by executing the following `yubihsm-shell` command: `yubihsm> session ykopen <authkey> <label> <password>` where, in the context of using YubiHSM-Shell with the YubiHSM Auth application, the following parameters are used:

- `authkey` is the identifier of the authentication key in the YubiHSM 2
- `label` is the label of the YubiHSM-Auth credentials stored in the YubiKey
- `password` is the password that protects the YubiHSM-Auth credentials stored in the YubiKey.

Below is an example of an interactive command with YubiHSM Shell:

```
yubihsm> session ykopen 1 "default key" "MyPassword"
trying to connect to reader 'Yubico YubiKey OTP+FIDO+CCID 0'
Created session 0
```

To use `yubihsm-shell` with YubiHSM Auth in command-line mode, add the parameter `--ykhsmauth-label` that implicitly invokes the YubiHSM Auth application at the YubiKey. Below is an example of how to use YubiHSM Shell in command-line mode:

```
$ yubihsm-shell --ykhsmauth-label "default key" -p "MyPassword" -a generate-asymmetric -
↵A rsa2048 -i 11 -c sign-pss -l Signature_Key
```

If the YubiKey is configured to require touch when accessing the YubiHSM-Auth credentials, the user needs to touch the YubiKey sensor in addition to entering the credential password.

Once the user is authenticated with YubiHSM Auth, all YubiHSM-Shell commands can be used.

To get in touch with Yubico Support, [click here](#).

TOOLS AND TROUBLESHOOTING

8.1 Managing Applications

8.1.1 Enabling/Disabling

The YubiKey Manager can be used to find out which applications are enabled on which interface and to enable or disable each application on each physical interface.

To find out which applications are enabled, select the **Interfaces** tab. A checkbox with a tick is shown next to each enabled application. To change which applications are enabled, use the checkboxes to select the ones you want enabled and click Save Interfaces.

Note: For the YubiKey 5Ci, any modifications made to the applications over the USB interface will also apply to the applications over Lightning®.

8.1.2 Locking

Once the desired applications have been selected, a lock code can be set to prevent changes to the set of enabled applications. This is done using the YubiKey Manager command line interface command `ykman config set-lock-code`. The lock code is 16 bytes presented as 32 hex characters. For more information, see the *YubiKey Manager (ykman) CLI & GUI Guide* <<https://docs.yubico.com/software/yubikey/tools/ykman/>> `.`.

8.2 YubiKey Manager (ykman)

The *YubiKey Manager* is a tool for configuring all aspects of YubiKeys in the 5 series and for determining the model of key and the firmware it runs. It has both a graphical interface and a command line interface. Being cross-platform, it runs on Windows, macOS, and Linux. Some of the more advanced options are only available through the command line. See the *YubiKey Manager (ykman) CLI Guide*.

8.2.1 Graphical User Interface (GUI)

The graphical user interface of the YubiKey Manager provides an easy-to-use method of performing basic configuration tasks of the YubiKey 5 Series, including:

- Displaying information about the YubiKey(s) connected to the computer.
- Enabling or disabling applications per physical interface.
- Setting or changing the FIDO2 PIN, as well as resetting the FIDO application.
- Managing the credentials in the OTP application.

8.2.2 Command Line Interface (CLI)

Using `ykman`'s CLI, you can do everything that the GUI can and more. This includes, but is not limited to:

- Enabling or disabling applications and prevent unauthorized changes by setting a lock code.
- Managing the credentials in the PIV / Smart Card application, including resetting them.
- Managing and generating OTPs from the credentials in the OATH application, including resetting the application.
- Resetting the OpenPGP application and setting the OpenPGP touch policy.

For usage information and examples for `ykman`, see the [YubiKey Manager \(ykman\) CLI Guide](#).

8.3 Yubico Authenticator

[Yubico Authenticator](#) is used to manage credentials on the OATH application and display the OTPs generated by the YubiKey. Yubico Authenticator is required in order to generate OTPs for OATH-TOTP credentials as the YubiKey does not contain a battery and thus cannot track time. It is open source, cross-platform, and runs on Windows, macOS, Linux, and Android. The Android version of Yubico Authenticator can communicate with YubiKeys over NFC or USB.

8.4 YubiKey Smart Card Minidriver

The YubiKey [Smart Card Minidriver](#) extends the PIV / Smart Card application on the YubiKey on Windows, facilitating deployment and management. Key benefits include:

- Enrollment of the YubiKey using standard Windows utilities.
- Auto-enrollment, enabling user self-provisioning of a YubiKey and automatic renewal.
- Multiple authentication certificates on one YubiKey.
- Changing of the PIN from the Ctrl+Alt+Del menu.
- Unblocking of the PIN using the PUK at the Windows logon screen.

To get started with the YubiKey Smart Card Minidriver, see the [deployment guide](#)

Note: For use with YubiKeys in the 5 Series, version 4.0 or later of the minidriver is required.

8.5 Troubleshooting

If you run into any issues with a key from the YubiKey 5 Series, refer to the [Knowledge Base](#) and search for your issue. If your issue is not listed in the Knowledge Base, or if you have any technical questions, you can get in touch with Yubico Support by [clicking here](#).

NFC ID CALCULATION TECHNICAL DESCRIPTION

9.1 Background to Door Access

The YubiKey 5 NFC can be used for physical access to doors. Essentially, the physical access system reads out the NFC ID from the YubiKey, truncates and parses the NFC ID in different ways, and checks if there is a match to a registered value in a database. If there is a match, the door is opened.

9.2 Calculation of NFC ID

For YubiKey 5.2.x and lower versions, the NFC ID was calculated as follows:

```
0x88 0x27 0 0 serial_3 serial_2 serial_1 serial_0
```

where `serial_0`, `serial_1`, `serial_2` and `serial_3` are the four bytes containing information about the YubiKey's serial number. In other words, `serial_x` is a byte that contains some of the digits of the serial number, however not a digit in itself.

`serial_0` is the most significant digit, ranging to `serial_3` which is the least significant digit. The least significant digit (`serial_3`) changes most frequently, while the most significant digit (`serial_0`) changes with the lowest frequency.

When a door access system reads out the NFC ID from the YubiKey, the NFC ID may be truncated and reversed in different ways before it is matched to the registered IDs in a database. In some cases, the most significant digits are parsed out and placed first, while the rest of the NFC ID is truncated. Such processing has in some cases resulted in parsed NFC ID values that consist of the most significant digits such as `serial_0` and `serial_1`, which may not be unique for a batch of YubiKeys. In other cases, only `0x27 0 0` are used, resulting in non-unique values.

9.3 NFC ID Calculation for YubiKey v5.3.0 and Above

For YubiKeys with firmware of 5.3.0 and above, the NFC ID calculation has been changed such that the NFC ID is now derived as:

```
0x88 0x27 serial_3 serial_2 serial_1 serial_0 serial_2 serial_3
```

Note that two of the four bytes in the serial number are repeated both at the beginning and at the end of the sequence.

(For the Security Key by Yubico, which does not have a serial number, the NFC ID is calculated as follows:

```
0x08 AA BB CC where AA, BB and CC are random bytes.)
```

This updated calculation of the NFC ID ensures unique values, regardless of the parsing direction of the NFC ID, whether from left to right or right to left.

To get in touch with Yubico Support, [click here](#).

SECURE CHANNEL SPECIFICS

10.1 Yubico Secure Channel Technical Description

10.1.1 Introduction to Yubico Secure Channel

Yubico has implemented a subset of the (GlobalPlatform Secure Channel Protocol 03) Secure channel specification: specifically, only the most secure implementation including command and response message authentication code (MAC) and encryption.

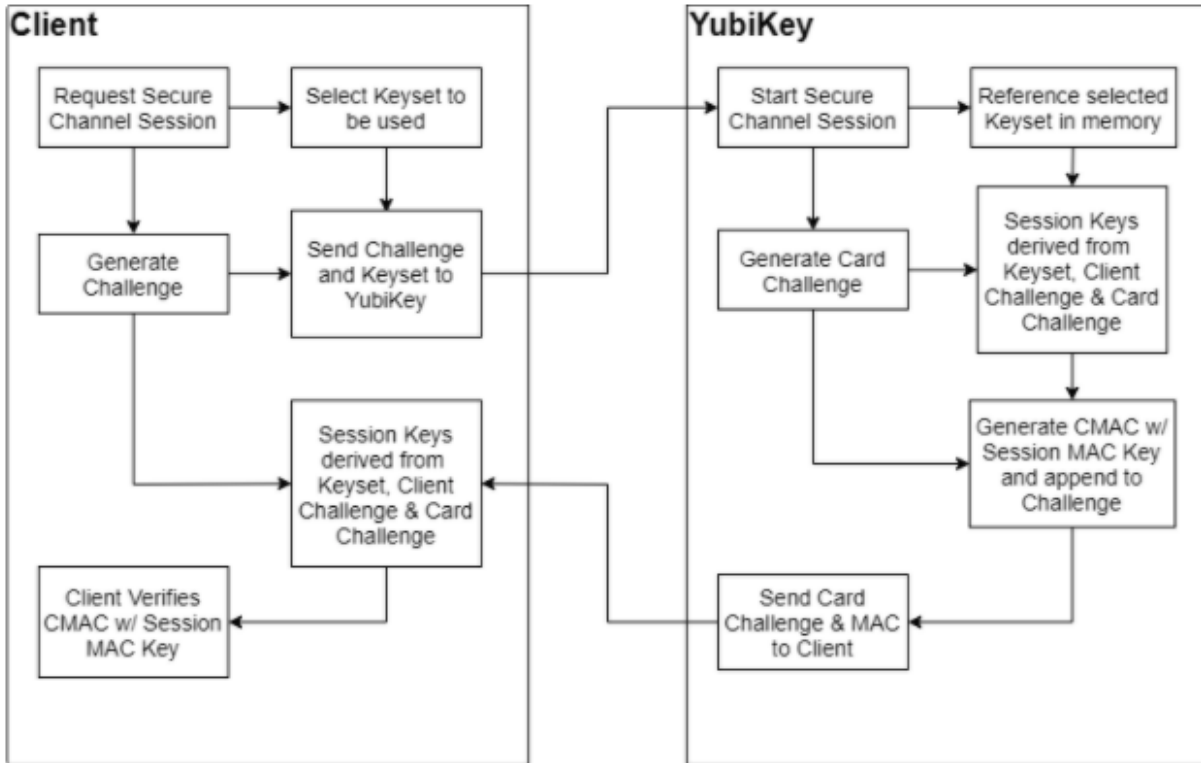
At the highest level, implementing a secure channel consists of providing overhearing and tampering resistance to information being sent between an external service, like a Credential Management Solution (CMS) and a smart card. Overhearing resistance is accomplished by the AES encryption of all commands being sent and received by use of a unique, private symmetric AES key. Tamper resistance is included by sending a securely encrypted MAC of both the commands and associated responses using an AES key unique to each session. Since these protections are applied to the data at the endpoints of the communication channel, a standard CCID interface can be used without modification, supporting native flows in Windows, Linux and other systems.

On the YubiKey, all of the secure channel operations occur within the secure cryptographic processor, with the plain text of the communication never exposed to outside observers.

Flow when Initializing a Secure Channel on a YubiKey

10.1.2 YubiKey Secure Channel Support

The YubiKey Secure Channel implementation is separate from the rest of the functionality on the YubiKey. It is only active when a secure channel is established and sits between the input of APDU commands sent into the secure element and the applications on the YubiKey. As such, any command which can be sent as an APDU over CCID can use secure channel, regardless if it is for PIV, OTP or other supported functions. The only exceptions are the FIDO protocols (U2F/WebAuthn), as they do not support communication over the CCID channel.



10.1.3 Transport Keys and Session Keys

Key	Usage	Creation
Static Secure Channel Encryption Key (Key-ENC)	Generate session key for Decryption/ Encryption (AES)	Imported from Trusted source
Static Secure Channel Message Authentication Code Key (Key-MAC)	Generate session key for Secure Channel authentication and Secure Channel MAC Verification and Generation (AES)	Imported from Trusted source
Data Encryption Key (Key-DEK)	Sensitive Data Decryption (AES) used for encryption other Transport key sets on import	Imported from Trusted source
Session Secure Channel Encryption Key (S-ENC)	Used for data confidentiality	Dynamically Created Per Session
Secure Channel Message Authentication Code Key for Command (S-MAC)	Used for data and protocol integrity	Dynamically Created Per Session
Secure Channel Message Authentication Code Key for Response (S-RMAC)	User for data and protocol integrity	Dynamically Created Per Session

The Yubico Secure Channel uses two types of AES keys as defined in the SCP03 specifications; these are organized in the static, externally sourced and imported transport keys, and the dynamic, internally generated session keys. The YubiKey can hold up to 3 transport key sets, and generates unique session keys for each session.

10.1.4 Transport Keys

A Transport Key set is made of 3 AES keys:

- Secure Channel Encryption Key (**KEY-ENC**)
- Secure Channel Message Authentication Code Key (**Key-MAC**)
- Data Encryption Key (**Key-DEK**)

The transport key sets used for establishing the secure channels are protected in the SCP03 security domain in the secure element. A transport key set contains three long-lived keys, imported from an external source. When a session is established, the session keys are derived from the long-lived transport key set.

The YubiKey security domain can store three concurrent long-lived transport key sets. In order to import new transport key sets, a secure channel must be established with the security domain. This has to be done with a previously loaded transport key set or the default transport key set.

The Secure Channel Encryption Key is used when initializing a session to generate the Session Secure Channel Encryption Key to be used during that session. Likewise, the Secure Channel MAC Key is used to generate the Session Secure Channel MAC key for Command and Session Secure Channel MAC Key for Response. The Data Encryption Key is only used when importing new transport key sets; the keys to be imported must be encrypted with a known Data Encryption key.

The Transport keys are imported from a CMS or HSM over an established secure channel. YubiKeys are shipped with either default values for the transport keys, or values derived from a Batch Master Key set at programming. Transport keys can and should be rotated on a regular basis depending on the threat model for the organization. Once overwritten on a YubiKey, Transport keys cannot be restored, so they should be archived on the CMS if necessary.

10.1.5 Session Keys

The Session Key set is made of 3 AES keys:

- Session Secure Channel Encryption Key (**S-ENC**)
- Secure Channel Message Authentication Code Key for Command (**S-MAC**)
- Secure Channel Message Authentication Code Key for Response (**S-RMAC**)

Session keys are all dynamically generated at the start of each session, using the Secure Channel Encryption and MAC Transport Keys, as well as the challenge sent from the client at the session start. For more details, refer to the GP SCP03 spec, section 4.1.5.

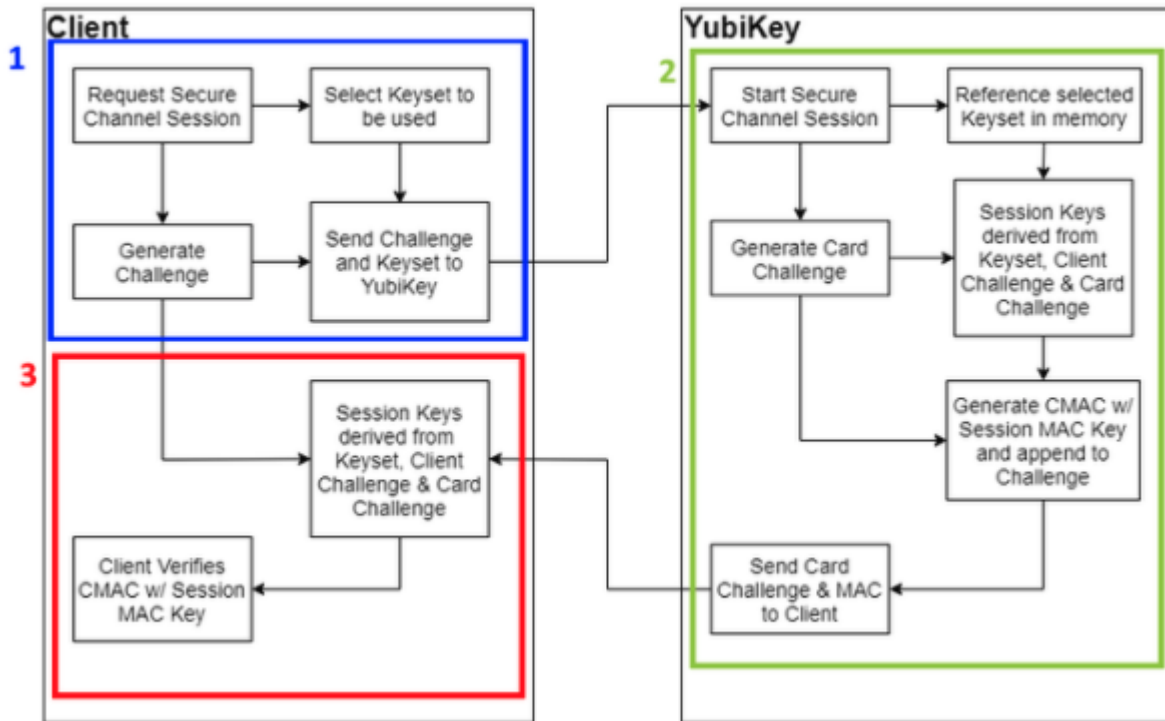
Every command sent over a secure channel between a client or CMS and a YubiKey is encrypted with the Session Secure Channel Encryption Key. Further, each command from the client has a MAC generated from the contents of the encrypted command APDU and the Session MAC key for Command. This MAC value is used to verify the authenticity of the command sent. Each command MAC value is based off the previous command MAC, enabling a chain which can be verified to ensure the data was not tampered in transit, nor is a command being replayed.

The Response MAC is generated using the encrypted response APDU from the YubiKey and the Session MAC key for Response. Each Response MAC also includes a value derived from the original command MAC sent from the client, providing a verification that the data included in the response corresponds to the last command sent.

10.1.6 Establishing a Secure Channel

A client connecting to any CCID function on the YubiKey, can establish a secure channel at the start of a session. Once a session has been started with a secure channel, all communication to and from the YubiKey over that session must be encrypted; sending a command in plain text will not be accepted and immediately end the session, removing any authorizations granted previously.

To begin, the client will identify the function they wish to communicate with and send the **Initialize Update** command.



YubiKey Secure Channel Initialize Update Flow

Step 1

When a client starts the process of establishing a secure channel with the YubiKey, it will select the Transport key set on the YubiKey to be used, as well as generating a unique challenge. This challenge will be used by the client to derive the session keys which will be utilized going forward. The Initialize Update command is sent from the client, which includes the challenge and Transport Key set identifier, to the YubiKey's CCID function they want to establish a secure channel with.

Step 2

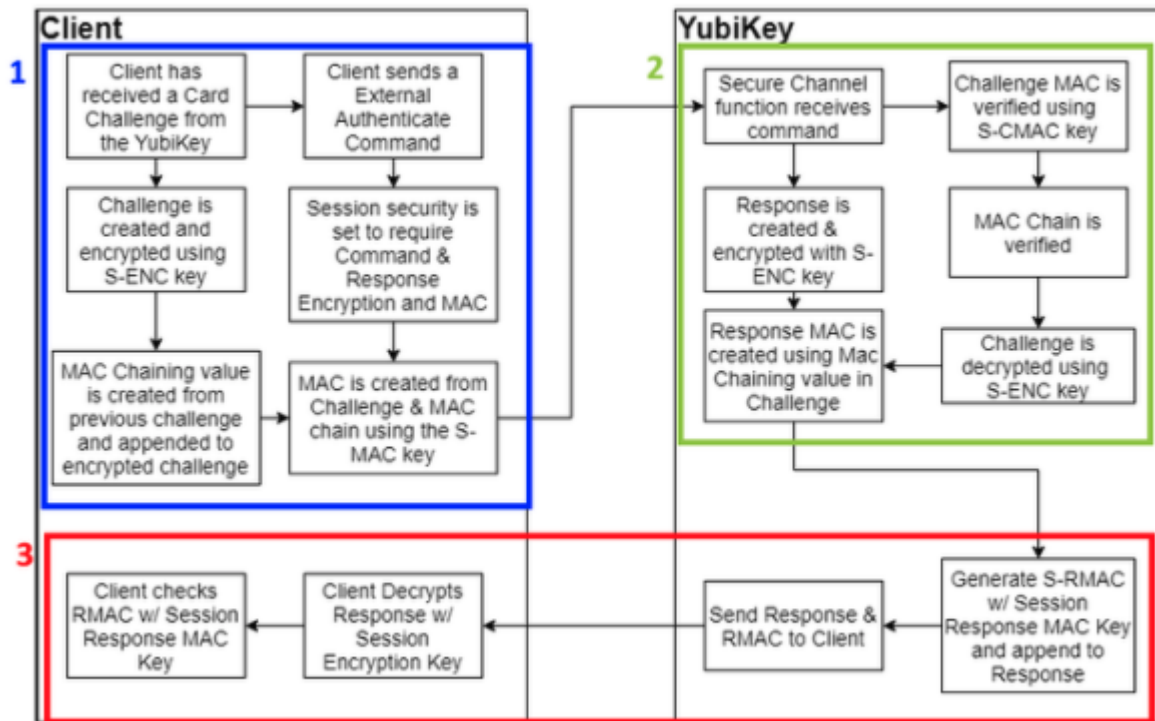
The YubiKey, upon receiving an Initialize Update command when a client is starting a session with any CCID facing function, will direct communication between a client and the YubiKey to the Secure Channel function for the remainder of the session. The Secure Channel Function will use the Transport Key Identifier to select the Transport Key set in memory to use. The YubiKey will generate a card challenge and use it, along with the challenge provided from the client, to derive the Session keys for Encryption and MAC using the selected Transport Key set.

The YubiKey will then generate a command MAC from the previously internally generated card challenge using the Session Command MAC key (S-MAC). The card challenge from the YubiKey and associated MAC are sent back to the client.

Step 3

The client, upon receiving the response from the YubiKey, derives the session keys using the transport key set, the original challenge and the card challenge from the YubiKey. The client then verifies the MAC using the session keys it had generated. Upon a successful verification, the client can be confident that the YubiKey has generated matching session keys. However, at this point, the YubiKey does not know if the client has the correct key set.

To authorize the YubiKey to accept commands from the client, the **External Authenticate** command must be run next.



YubiKey Secure Channel External Authenticate flow

Step 1

The External Authenticate flow starts with the client receiving the card challenge from the YubiKey created during the Initialize Update command. From that point, the client will define the session security settings - the YubiKey only supports the strictest option, with both commands and responses encrypted and have associated MACs generated. As with the Initialize Update flow, the client creates a challenge and encrypts it with the session encryption key. However, a MAC value from the previous challenge is also created and appended to the challenge, creating a chain of commands to be tracked. The Challenge and MAC chain value are then used to create a command MAC using the S-MAC key, and both are sent to the YubiKey.

Step 2

The YubiKey receives the External Authenticate command, and verifies the Challenge using the MAC value and S-MAC key. The MAC Chain is then verified, confirming that the client has the same session keys and a secure channel has been created. At that point, the challenge is decrypted using the S-ENC key, and a response is created. In addition, the challenge from the client is used to create a new MAC chain value, which is appended to the response.

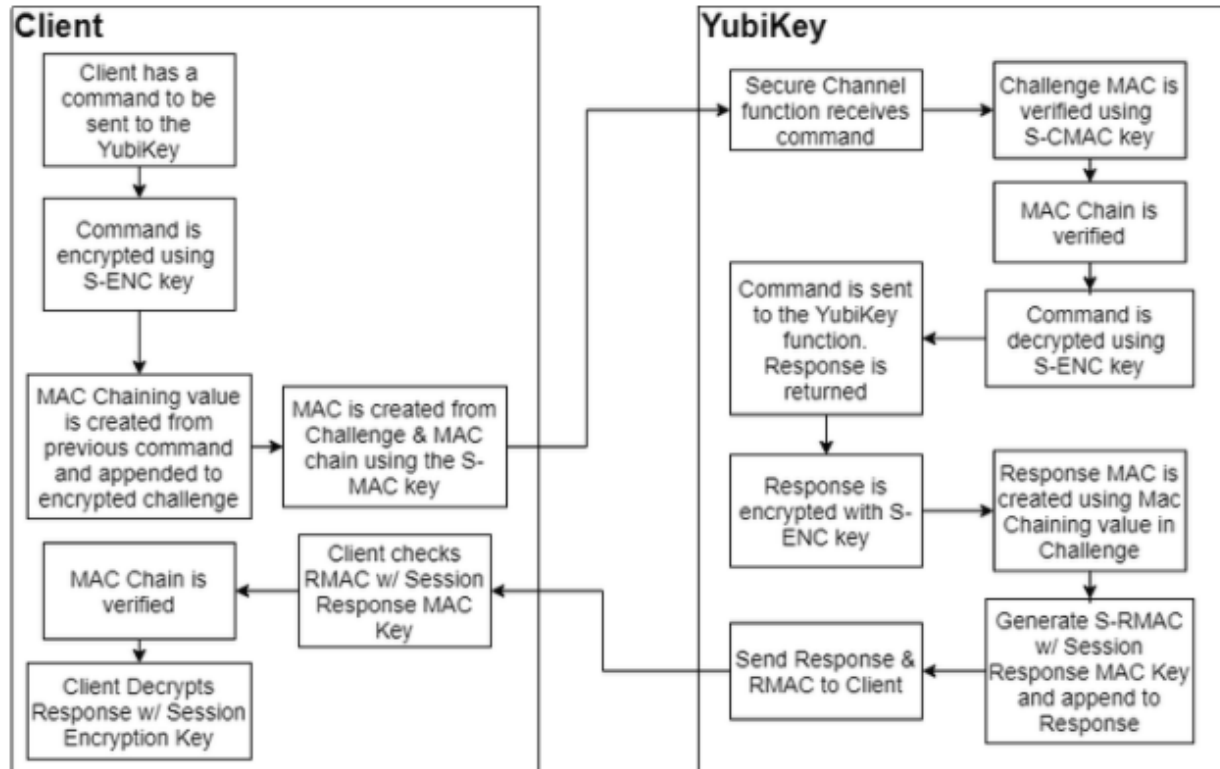
Step 3

The response and MAC Chain value are used to generate a response MAC using the S-RMAC key. Then the response and associated MAC are sent from the YubiKey back to the client. At this point, the response MAC is verified using the S-RMAC key, the MAC Chain value is verified against the

command sent previously, and the response is decrypted.

At this point, a secure channel has been established between the client and the YubiKey.

10.1.7 Communicating Over Secure Channel



Communicating Over Secure Channel Flow

When Command APDU is sent over an established secure channel, the Yubico Secure Channel follows an encrypt then MAC approach.

Step 1

The command APDU is first encrypted using the Session Encryption Key (S-ENC). It is important to note that any APDU delivering instructions or data (such as a key or certificate) to a YubiKey is considered a Command APDU. Sending an unencrypted command will end the current session and will remove any authorizations.

Step 2

A command MAC is created using the Session MAC key (S-MAC) using the encrypted APDU along with a MAC chain value created from the previous Command MAC. This is sent to the YubiKey.

Step 3

The YubiKey verifies the command MAC, then verifies the MAC Chain links to the previous command sent.

Step 4

With the MAC values verified, the command is decrypted and passed to the YubiKey functionality. A response from the function being communicated with is returned.

Step 5

The response is encrypted with the S-ENC key, then has a MAC chain value derived from the command appended to it. The response and chain MAC value are used with the Session Response MAC key to generate a response MAC. The Response and Response MAC are sent back to the client.

Step 6

The client performs the same operations, verifying the response MAC, verifying the MAC chain, then decrypting the response. The Response APDU are passed to the client.

To get in touch with Yubico Support, [click here](#).

10.2 Yubico Secure Channel Key Diversification and Programming

10.2.1 Introduction

The term “key diversification” refers to the process of deriving a secure channel static transport key set from a Batch Master Key (BMK), the YubiKey identifier (part of serial number), and additional metadata.

Benefits and Usage

Key diversification enables simplified and secured distribution of secure channel transport key sets as only the Batch Master Key must be shared with the CMS system to derive the YubiKey transport key sets.

Hence, the secure channel transport key sets can be pre-programmed by Yubico, assuming that Yubico has access to the BMK of the CMS vendor.

Another option is for the CMS system to generate the secure channel transport key sets based on the YubiKey serial number, the BMK, and additional metadata. The CMS can then use the initial secure channel transport key set for writing additional secure channel transport key sets to the YubiKeys.

SCP03 Key Diversification

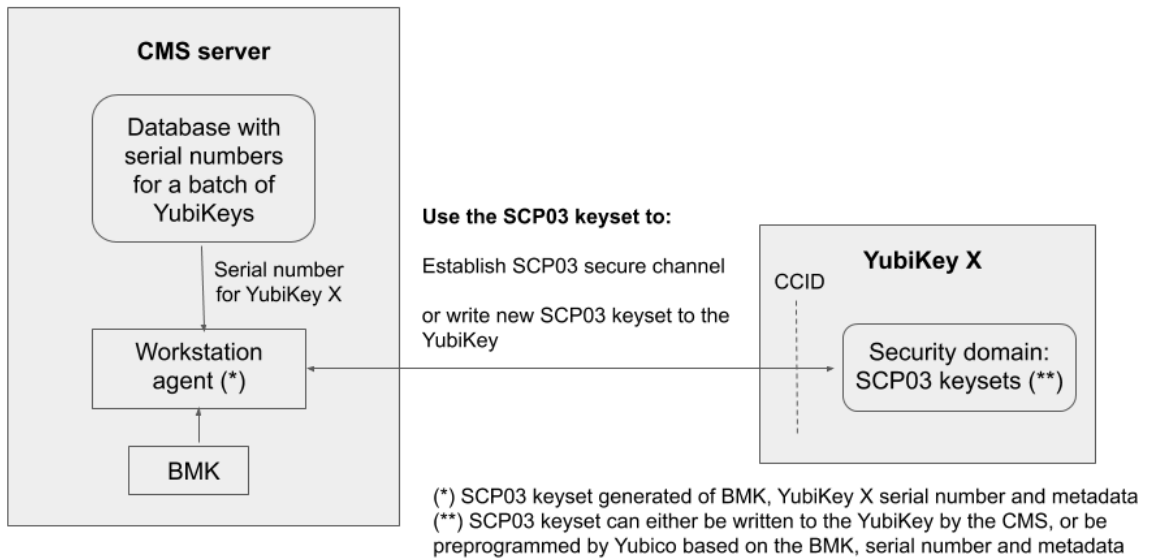
10.2.2 Secure Channel and Security Domains

The YubiKey supports up to 3 secure channel transport key sets. This is to enable more granular control over the establishment of a secure channel to a specific device. The keys in each of these key sets can be overwritten once connected to the YubiKey, allowing for a YubiKeys to be shipped with a default key set, then have the key set be changed to a random set of keys at initialization, ensuring only the CMS server has the actual keys.

10.2.3 Key Diversification Option

When purchasing YubiKeys from Yubico, there is an option to custom-configure the transport keys from the default to values derived from details specific to the hardware and a BMK. This means these keys can be distributed with locked down key sets, ensuring they cannot be connected to remotely by third parties. The BMK is generated and owned by the customer, who in turn can provide it to Yubico and their CMS deployment. The CMS can then use the BMK to establish a secure channel to a customer’s YubiKeys, and set new transport keys during initialization, limiting access to just that CMS. As with other custom configuration options, these keys can be overwritten or deleted by the customer; they are not “baked into” the YubiKey firmware.

SCP03 key diversification



Batch Master Key (BMK) Generation

Each custom order with diversified keys has a unique BMK, which is used when Yubico programs the keys. A BMK is a 32-bit AES key that Yubico recommends the customer generate in a secure manner approved by their own internal security department. The YubiHSM2 can be used to generate a random AES key with the GET PSEUDO RANDOM command, e.g. `get random 0 32`. For more information, see [this command in the HSM 2 Commands](#) reference guide.

Before every order with Key Diversification, the customer will generate and provide the BMK to Yubico by secure means. After the YubiKeys have been programmed using the BMK provided, the BMK on the Yubico programming station is destroyed, leaving the customer with the only extant BMK. The customer must maintain and securely archive their BMK if required for future orders.

Key Diversification Function (KDF)

The diversification function used is the AES-CMAC-KDF Counter Mode derivation algorithm specified in NIST SP800-108.

Scroll horizontally to see the diversified key on the next line:

```
AES CMAC of [ Counter (1 byte) || Label (4 bytes) || 00 || Context (10 bytes) || Key
↳Length in bits (2 Bytes) ]
```

Note: AES256 and 3DES keys require two rounds of KDF to generate a 32-byte key value and 24-byte key value respectively, the first one with a counter value set to 01 and the second one with a counter value set to 02.

The Key Length in the KDF input file is expressed in hexadecimal value. It is:

- 0100 for a 256-bit key,

- **00C0** for a 192-bit Key (e.g. PIV Admin Key),
- **0080** for a 128-bit key (e.g. ISD Keys & Interfaces Management Key) and
- **0040** for a 64-bit code (e.g. the PUK).

Labels for Key Diversification

The KDF function of separating keys uses the following labels as input. Note that these are example values, keeping Yubikey 5 Series implementation with ISD Keys as 16-byte values, YubiKey Interfaces Management Key as a 16-byte value, , the PIV Admin Key as a 24-byte value and the PUK as an 8-byte value.

Factory Key Codes	Key Length in bits	KDF Label
Issuer Security Domain DAK (Authentication Key)	'0080'	'00000001'
Issuer Security Domain DMK (MAC Key)	'0080'	'00000002'
Issuer Security Domain DEK (Encryption Key)	'0080'	'00000003'
PIV Application Administrative Key	'00C0'	'00000004'
PIV Application PUK	'0040'	'00000007'
Capabilities Lock Code (YubiKey Interfaces Management Key)	'0080'	'00000010'

In general the Key Length should be derived from the table below.

Key Size	Key Length in Bits
32 Bytes	'0100'
24 Bytes	'00C0'
20 Bytes	'00A0'
16 Bytes	'0080'
8 Bytes	'0040'

Context for Key Diversification

The value of the **Context** field in the KDF input data is the **Issuer Context** and is equal to the first 10 bytes of the value returned from the Global Platform INITIALIZE UPDATE command.

PUK Generation from Diversified Value

We use the trailing 8 bytes of the Diversified Value and generate the PUK using the pseudocode below. In the HTML version of this guide, you may need to scroll horizontally to see the full line.

```
for (int i = 0; i < 8; i++) { diversifiedVal[i] = (byte) (0x30 + ((diversifiedVal[i] & 0x7F) % 10));}
```

10.2.4 Global Platform: CPLC Data

Description

Although this format is officially deprecated and not part of the GlobalPlatform specification, some organizations need support for the information stored in the so-called CPLC (Card Production Life Cycle).

This consists of a static set of bytes that can be retrieved with a GET DATA command (INS 0xca) and TAG 0x9f7f after selecting the SD application.

The response is 42 bytes that can be parsed into different fields with different meanings. However, Yubico elected not to attribute any specific meaning to 40 of those bytes. Only the first two bytes are meaningful.

Example Command

To retrieve the value (scroll horizontally if necessary):

```
opencsc-tool -c default -s '00a4040008a000000151000000' -s '00ca9f7f'
```

Relevant Output

```
40 90 73 F9 53 94 C0 01 23 D8 E9 F0 68 3A 48 9A @.s.S...#...h:H.
76 30 4C D8 F6 CC 41 66 61 0F C4 F5 8C DE D6 93 v0L...Afa.....
77 32 09 82 1B EA 0C 78 3D 8B                w2.....x=.
```

Of those 42 bytes, only the first two (40 90) are meant to signify an Infineon SLE 78 chipset, the rest are random bytes generated when the SD application is (re)initialized.

To get in touch with Yubico Support, [click here](#).

10.3 Yubico SCP03 Developer Guidance

This section describes how Secure Channel Protocol 3 works in the YubiKey for developers integrating support for it.

10.3.1 Introduction

SCP03 is a protocol from Global Platform for mutual authentication and encrypted transport using smart cards. The protocol allows for the following modes of encryption and authentication of data:

- C-MAC,
- C-ENC,
- R-MAC, and
- R-ENC.

The YubiKey implements this with all of them turned on; turning anything off is not an option.

Authenticating with SCP03 does not assign any specific permissions in the YubiKey; what it does is set up a mutually authenticated and encrypted channel between the YubiKey and the host. Unencrypted commands sent over the secure channel will end the session, revoking any previously issued authorizations.

For more details on SCP03, refer to the [Global Platform SCP03 specifications](#).

10.3.2 Key Sets

A key set contains three long-lived keys, the encryption key (**Key-ENC**), the mac key (**Key-MAC**), and the data encryption key (**Key-DEK**). When a session is established, the session encryption key is derived from the ENC key, while the session mac keys are derived from the MAC key. Any new key sets transported over the session are encrypted with the DEK.

The YubiKey only allows putting or deleting a whole key set at a time, not manipulating the individual keys within the set.

Each key set is identified by the key version defined when the set is imported into the YubiKey. Each individual key also has an id, but that serves solely to identify the specific key within the set - ENC, MAC or DEK. The key version number is required for addressing the correct set. 255 is the factory default version and therefore that version number is reserved. When importing a key set, the version is set to a value in the range 1-254.

The YubiKey can store up to three key sets at a time. By default there is one key set installed with key version 255 having the value `404142434445464748494a4b4c4d4e4f` for all three keys, which are known as the test keys. When a new key set is installed, it replaces the default key set. The YubiKey supports only AES-128 for all three keys.

When authentication with a key set fails repeatedly (i.e., 32 times in a row) that key set is deleted. When the last key set is deleted, the security domain is automatically reset with the default key set installed. To delete the last key set on purpose and force a reset, the delete instruction is sent with `p2=1`.

10.3.3 Sessions

The session is established only within the scope of the currently selected applet. When a new applet is selected, the session is terminated. To manage SCP03 keys, a session needs to be established with the AID `a00000001510000000` - the Issuer Security Domain.

When a large amount of data is to be transported over the session, it is encrypted and mac'ed in its entirety. If the data exceeds the capacity of a single message, it is chunked for transport.

10.3.4 CPLC

The security domain contains an entry called CPLC which identifies a specific device. On a YubiKey, this entry is filled with random data on first boot. No significance is to be ascribed to any of the fields.

10.3.5 Software

Yubico has conformed to the Global Platform Open Standard, and as such, has developed the SCP03 support on the YubiKey to be compatible with open source offerings.

10.3.6 GlobalPlatformPro

GlobalPlatformPro is a Java library and tool for interacting with smartcards supporting the GlobalPlatform secure channel protocols.

Examples

Some of the following examples are long lines of code; for those, you might have to scroll horizontally.

Open a channel with the security domain and print information

```
$ java -jar tool/target/gp.jar --mode mac --mode enc --mode rmac --mode renc --debug --
↳ info
```

Open a channel with the security domain and install a new key set

```
$ java -jar tool/target/gp.jar --mode mac --mode enc --mode rmac --mode renc --debug --
↳ lock 000102030405060708090a0b0c0d0e0f
```

Open a channel with the PIV applet and verify the PIN over the channel

```
$ java -jar tool/target/gp.jar --mode mac --mode enc --mode rmac --mode renc --debug --
↳ sdaid a000000308000010000100 -s 0020008008313233343536ffff
```

10.3.7 gpshell

Gpshell is a C library and tool for interacting with the secure channel protocols.

Examples

Gpshell works with scripts; here is an example of opening a channel with the YubiKey. To see the last line of code in its entirety, scroll horizontally.

```
enable_trace
mode_211
establish_context
card_connect
select -AID a000000151
open_sc -enc_key 404142434445464748494a4b4c4d4e4f -mac_key_
↳ 404142434445464748494a4b4c4d4e4f -kek_key 404142434445464748494a4b4c4d4e4f -security_
↳ 51 -scp 3 -scimpl 96
```

10.3.8 References

- <https://globalplatform.org/specs-library/card-specification-v2-3-1/>
 - <https://globalplatform.org/specs-library/secure-channel-protocol-03-amendment-d-v1-2/>
 - <https://github.com/martinpaljak/GlobalPlatformPro/>
 - <https://sourceforge.net/projects/globalplatform/>
 - <https://sourceforge.net/p/globalplatform/wiki/GPShell/>
-

To get in touch with Yubico Support, [click here](#).

To get in touch with Yubico Support, [click here](#).

YUBIKEY 5 FIPS SERIES SPECIFICS

11.1 Deploying the YubiKey 5 FIPS Series

The YubiKey 5 FIPS Series keys are certified under FIPS 140-2 Level 1 and FIPS 140-2 Level 2. Keys in this series have two certificates, each corresponding to a different level of certification, but both certificates apply to the same keys. The YubiKey chipset is certified at FIPS 140-2 Physical Security Level 3, providing both tamper-evidence and tamper-resistance. This means the YubiKey 5 FIPS Series keys can be used in an Overall Security Level 1 or 2 environment without issue. Depending on which certification the YubiKey 5 FIPS Series is being deployed under, there are different requirements as to how the various functions are to be secured. To review the differences between the considerations and requirements for a FIPS 140-2 Level 1 authenticator and those for a FIPS 104-2 Level 2 authenticator, see [fips5-levels-label](#).

[NIST SP 800-63-B](#) provides guidance on the level required for your deployment.

In cases where only Level 1 is required, the end-user experience with a YubiKey 5 FIPS Series is similar to that of a user with key from the YubiKey 5 Series. The user experience with YubiKey 5 FIPS Series deployed under FIPS 140-2 Level 2 is much more onerous.

NIST classified the YubiKey 5 Series FIPS as ‘composite authenticators.’ As such, no device in that series can be taken out of the FIPS-approved mode after initialization without zeroizing the function. This means that once the YubiKey is correctly configured, it remains in the correct configuration. This is what renders the `--check-fips` command unnecessary. If the crypto officer ensures that the YubiKey 5 Series FIPS devices are correctly configured at initialization, they remain in FIPS-approved mode.

11.1.1 Configuring the YubiKey 5 FIPS Series under FIPS 140-2 Level 1

Without any configuration, the YubiKey 5 FIPS Series meets the requirements for the FIPS 140-2 Level 1 certification as an authenticator with FIPS-approved algorithms. Security Level 1 allows an authenticator to be used on a general purpose computing system using an unevaluated operating system. This can include computers or OSs that are configured in a FIPS-certified mode of operation, but which might not have extensive access controls or auditing features. Any function on the YubiKey may be used. The only non-approved algorithms are:

- RSA 1024-bit keys
- EdDSA keys
- X25519 keys

11.1.2 Configuring the YubiKey 5 FIPS Series under FIPS 140-2 Level 2

Security Level 2 includes all of the requirements for FIPS Level 1, but further enforces enhanced physical security mechanisms and a separation of functions with regard to role-based authentication. Security Level 2 allows an authenticator to be used on a general purpose computing system with an operating system that has been evaluated at EAL2 with role-based access control mechanisms and comprehensive auditing.

The role-based authentication minimum requirement is one in which a cryptographic module authenticates the authorization of an operator to assume a specific role and perform a corresponding set of services. A Security Officer role is required for services such as importing or generating new credentials or programming new OTP secrets on a YubiKey. The User role covers the actual usage of programmed credentials for authentication. The Crypto Officer role is that of “a cryptographic officer [who] is authorized to perform cryptographic initialization and management functions on a CKMS [Cryptographic Key Management System] and its cryptographic modules.” (Quote taken from SP 800-130 (DOI).)

To act in an Overall Security Level 2 environment, a YubiKey must be configured in a FIPS-approved mode of operation OR receive an exemption from the security auditor.

Note: To load key data over NFC a secure channel must be used. For more information on Secure Channel (SCP03) in connection with YubiKeys, see the [topic of that name in the YubiKey 5 Series Technical Manual](#). For more information on SCP03 requirements from NIST, see [NIST Special Publication 800-63C](#) and [NIST Special Publication 800-63B](#).

When using a security key from the YubiKey 5 FIPS Series as a FIPS 140-2 Level 2 authenticator in a FIPS environment, in order for the device to be considered as operating in a FIPS-approved mode, all of the applications must be in a FIPS-approved mode of operation.

Not all of the applications on the YubiKey 5 FIPS Series are in a FIPS mode of operation by default. The person filling the crypto officer role in deploying the YubiKey 5 FIPS Series in a secured environment must define and supervise an initialization and delivery process that ensures that each application on the YubiKey 5 FIPS Series is in a FIPS-approved mode of operation before being deployed to end-users.

Every function of the YubiKey must require permissions defined by role; in practice, this is accomplished by setting the access codes, management keys, passwords, PINs, etc. for every function on the YubiKey.

To ensure that each application is in a FIPS-approved mode of operation, use the **YubiKey Manager (ykman)** Command Line Interface (CLI).

- Download the YubiKey Manager tool: <https://www.yubico.com/products/services-software/download/yubikey-manager/>
- YubiKey Manager (ykman) CLI & GUI Guide: <https://docs.yubico.com/ykman/>

Note: It is not permissible to use U2F when the YubiKey 5 FIPS Series is deployed as a 140-2 Level 2 authenticator.

Note: Even if FIPS 140-2 Level 2 does not require that all the credentials across all the applications be changed from the default values before the YubiKey 5 FIPS Series device is deployed to the end user, it is highly recommended that these default values be changed.

Credentials and Permitted Values

The table below lists the credentials required, allowed values, and credential owner for the supported applications.

Application	Credential	Permitted Values	Credential Owner
One Time Password (OTP)	Access Code: OTP Slot 1 OTP Slot 2	6 byte access codes 6 byte access codes	Crypto Officer
OATH	Authentication Key	14-64 byte HMAC SHA1/SHA256 key	Crypto Officer
PIV Smart Card	Management Key	3-key TDES key	Crypto Officer
	PUK	6-8 byte PIN	Crypto Officer
	PIN	6-8 byte PIN	Authenticated User
OpenPGP	User Password (PW1)	6-127 byte PIN	Authenticated User
	Admin Password (PW3)	8-127 byte PIN	Crypto Officer
WebAuthn	PIN	6 to 32 byte PIN	Authenticated User

The instructions for the individual applications are provided in the following topics:

- [OTP](#)
- [OATH](#)
- [PIV](#)
- [OpenPGP](#)
- [WebAuthn](#)

11.2 OTP: FIPS 140-2 with YubiKey 5 FIPS Series

The OTP application provides two programmable slots, each of which can hold one of the types of credentials listed below. A Yubico OTP credential is programmed to slot 1 during manufacturing.

- Trigger the YubiKey to produce the credential in the first slot by briefly touching the metal contact of the YubiKey.
- If a credential has been programmed to the second slot, trigger the YubiKey to produce it by touching the contact for 3 seconds.

Output is sent as a series of keystrokes from a virtual keyboard.

11.2.1 Yubico OTP

Yubico OTP is a strong authentication mechanism that is supported by all YubiKey 5 FIPS Series. Yubico OTP can be used as the second factor in a two-factor authentication (2FA) scheme or on its own, providing single-factor authentication.

The OTP generated by the YubiKey has two parts, with the first 12 characters being the public identity which a validation server can link to a user, while the remaining 32 characters are the unique passcode that is changed each time an OTP is generated.

The character representation of the Yubico OTP is designed to handle a variety of keyboard layouts. It is crucial that the same code is generated if a YubiKey is inserted into a German computer with a QWERTZ layout, a French one with an AZERTY layout, or a US one with a QWERTY layout. The “Modhex”, or Modified Hexadecimal coding, was invented by Yubico to use only specific characters to ensure that the YubiKey works with the maximum number of keyboard layouts. (USB keyboards send their keystrokes by means of “scan codes” rather than the actual character. The translation to keystrokes is done by the device to which the YubiKey is connected).

11.2.2 OTP Deployment

The YubiKey 5 FIPS Series OTP application supports two independent OTP configurations, known as OTP slots. The OTP slots can be configured to output an OTP created with the Yubico OTP or OATH-HOTP algorithm, a HMAC-SHA1 hashed response to a provided challenge or a static password. The output of OTP slot 1 is triggered by a short touch (1~3 seconds) on the gold contact and the output of OTP slot 2 is triggered by a long touch (+3 seconds).

A 6-byte access code can be set on slot 1 and slot 2 independently. Once set, the OTP slot’s access code is required when modifying, overwriting or deleting the configuration on the respective OTP slot. By default, the YubiKey is shipped without any access code.

FIPS 140-2 Level 2: Placing the OTP Application in FIPS-approved Mode

Each OTP slot must be locked down with an access code for the YubiKey 5 FIPS Series OTP application to be in a FIPS-approved mode of operation. By default, no access codes is set for either slot.

- An access code must be applied to each OTP slot, either:
 - When writing a new configuration or
 - By updating an existing configuration in an OTP slot.
- An access code cannot be applied to an empty OTP slot.
- To secure an unused OTP slot, use a blank OTP configuration with an access code.
- YubiKey 5 FIPS Series devices must either be deployed with

- The OTP slots already set with an access code, or
- An OTP application or service which configures the access code on both slots on enrollment.
- The OTP slot access codes must be archived so that only the crypto officer alone can access them, as the access codes are used when resetting the OTP application.

Using the YubiKey Manager to Set Access Codes

The crypto officer can set an access code to the OTP slots using the YubiKey Manager Command Line Interface (CLI).

- Download the YubiKey Manager tool: <https://www.yubico.com/products/services-software/download/yubikey-manager/>
- YubiKey Manager (ykman) CLI & GUI Guide: <https://docs.yubico.com/ykman/>

To **apply an access code to a configuration** using the YubiKey Manager CLI, include the flag `--new-access-code=<access code>` in the OTP configuration string. The command must be of the format:

```
ykman otp settings --new-access-code=<access code> [OTP Slot]
```

where `<access code>` is the access code to be set, and `[OTP Slot]` is either 1 or 2 depending on if the OTP configuration is being applied to OTP slot 1 or OTP slot 2. For the characteristics of the access code, see [credential-values-label](#). For full details on setting an OTP configuration using the YubiKey Manager CLI, see the [section of that name in the YubiKey Manager CLI & GUI Guide](#).

To **fill a blank OTP slot** with a default configuration, use the command:

```
ykman otp chalresp --generate [OTP Slot]
```

where `[OTP Slot]` is either 1 or 2 depending on if the OTP configuration is being applied to OTP slot 1 or OTP slot 2.

11.3 OATH: FIPS 140-2 with YubiKey 5 FIPS Series

The YubiKey 5 FIPS OATH application can store up to 32 OATH credentials, either OATH-TOTP (time-based) or OATH-HOTP (counter-based), as defined in the [OATH specification](#). These credentials are separate from those stored in the OTP application, and can only be accessed via the CCID channel.

When an OATH-HOTP credential is programmed, the OTP is generated using the standard [RFC 4226](#) HOTP algorithm and the YubiKey will automatically type the OTP. Optionally, the OTP can be prefixed by a public identity, conforming to the [openauthentication.org Token Identifier Specification](#).

To manage the OATH credentials and read the OTPs generated by the YubiKey, the [Yubico Authenticator](#) is required. The Yubico Authenticator is supported on Windows, Linux, macOS, Android and iOS.

11.3.1 FIPS 140-2 Level 2: Placing the OATH Application in FIPS-approved Mode

Access to the YubiKey 5 FIPS Series OATH application must be protected with an Authentication Key for the application to be in a FIPS-approved mode of operation. To get the permitted values for the following operation, see [credential-values-label](#).

The crypto officer can set the Authentication Key using the YubiKey Manager Command Line Interface (CLI).

- Download the YubiKey Manager tool: <https://www.yubico.com/products/services-software/download/yubikey-manager/>
- YubiKey Manager (ykman) CLI & GUI Guide: <https://docs.yubico.com/ykman/>

To set an Authentication Key using the YubiKey Manager CLI, use the command:

```
ykman oath access change -n <Authentication Key>
```

where <Authentication Key> is the Authentication Key to be set.

11.4 FIDO: FIPS 140-2 with YubiKey 5 FIPS Series

11.4.1 FIDO U2F

FIDO U2F is an open standard that provides strong, phishing-resistant two-factor authentication for web services using public key cryptography. U2F does not require any special drivers or configuration to use, just a compatible web browser. The U2F application on the YubiKey can be associated with an unlimited number of U2F sites.

11.4.2 FIDO2

Like FIDO U2F, the **FIDO2** standard offers the same high level of security, as it is based on public key cryptography. In addition to providing phishing resistant two-factor authentication, the FIDO2 application on the YubiKey enables the storage of resident credentials. As the resident credentials can accommodate the username and other data, this enables truly passwordless authentication. Keys in the YubiKey 5 FIPS Series can hold up to 25 resident keys.

Locking FIDO2 Credentials

The resident credentials can be protected by a PIN for two-factor authentication.

- The FIDO2 PIN must be between 6 and 63 alphanumeric characters in length.
- Once a FIDO2 PIN is set, it can be changed but it cannot be removed without resetting the FIDO2 application.
- If the PIN is entered incorrectly 8 times in a row, the FIDO2 application will be locked, and FIDO2 authentication will not be possible. After 3 incorrect PIN entries, the FIDO2 application must be power cycled. In order to restore this functionality, the FIDO2 application must be reset.

Note: Resetting the FIDO2 application will also reset the U2F key. No site you have registered the YubiKey with using U2F will work until the YubiKey is re-registered with that site. However, using U2F is not compatible with FIPS 140-2 Level 2.

Note: The YubiKey 5 FIPS Series supports FIOD2 credential management, thereby enabling selective deletion of resident keys. See the [Enhancements to FIDO 2 Support](#) for details.

The rules governing FIPS-certified environments forbid the use of the following features of the YubiKey 5 FIPS Series:

- The P-224 curve
- Credential registration over NFC.

Default Values

PIN: None set.

11.4.3 Placing the WebAuthn Application in FIPS-approved Mode

For the YubiKey WebAuthn application to be in a FIPS approved mode of operation, a WebAuthn PIN must be set. By default, no WebAuthn PIN is set.

To **set or change the WebAuthn PIN**, the YubiKey Manager Command Line Interface (CLI) must be used. To set an WebAuthn PIN using the YubiKey Manager CLI, use the command:

```
ykman fido access change-pin -n<PIN>
```

where <PIN> is the WebAuthn PIN to be set. Get the PIN requirements from credential-values-label.

U2F

The YubiKey 5 U2F FIPS application cannot be used in a FIPS 140-2 Level 2 mode. In place of the U2F functionality, use the FIDO WebAuthn application. FIPS-certified services should not call the U2F functionality; nonetheless, the U2F function should be disabled on the YubiKey to ensure it is not used.

To disable U2F over USB and NFC, use the commands:

```
ykman config usb -dU2F ykman config nfc -dU2F
```

To **ensure users cannot enable U2F**, access to it can be secured with a management lock code. To set this code, use the command:

```
ykman config set-lock-code -n<lock code>
```

where <lock code> is a 16 byte (32 character) hex value.

Note: The lock code prevents anyone without it from changing which functions are accessible over NFC or USB. The lock code cannot be recovered if lost, which would result in a YubiKey with features permanently inaccessible.

11.5 PIV: FIPS 140-2 with YubiKey 5 FIPS Series

The YubiKey 5 FIPS Series provides a PIV-compatible smart card application. PIV, or FIPS 201, is a US government standard that enables RSA or ECC sign/encrypt operations using a private key stored on a smart card through common interfaces like PKCS#11. On Windows, the smart card functionality can be extended with the [YubiKey Smart Card Minidriver](#). The YubiKey Smart Card Minidriver is not available for Android, Linux, macOS or iOS.

Keys in the YubiKey 5 FIPS Series support extended APDUs, extended *Answer To Reset (ATR)*, and *Answer To Select (ATS)*. Using the PIV APDUs on iOS requires the Yubico iOS SDK.

For YubiKey 5 FIPS Series, some exceptions apply:

- Do not use non-NIST-approved curves
- Do not use the following keys:
 - RSA 1,024-bit

- 3,072-bit keys.

This applies to Attestation as well.

- PIN policy = none cannot be used. Select either once or always.

11.5.1 Default Values

- PIN: 123456
- PUK: 12345678
- Management Key (3DES): 010203040506070801020304050607080102030405060708

11.5.2 Supported Algorithms

The YubiKey 5 FIPS Series supports the following algorithms on the PIV smart card application.

- RSA 1024
- RSA 2048
- ECC P-256
- ECC P-384

11.5.3 Policies

PIN Policy

To specify how often the PIN needs to be entered for access to the credential in a given slot, set a PIN policy for that slot. This policy must be set upon key generation or import; it cannot be changed later.

Touch Policy

In addition to requiring the PIN, the YubiKey can require a physical touch on the metal contact. Similar to the PIN policy, the touch policy must be set upon key generation or import.

11.5.4 Slot Information

The keys and certificates for the smart card application are stored in slots, which are described below. The PIN policies described below are the defaults, before they are overridden with a custom PIN policy. **These slots are separate from the programmable slots in the OTP application.**

Slot 9a: PIV Authentication

This certificate and its associated private key is used to authenticate the card and the cardholder. This slot is used for system login, etc. To perform any private key operations, the end user PIN is required. Once the correct PIN has been provided, multiple private key operations may be performed without additional cardholder consent.

Slot 9c: Digital Signature

This certificate and its associated private key is used for digital signatures for the purpose of document, email, file, and executable signing. To perform any private key operations, the end user PIN is required. The PIN must be submitted immediately before each sign operation to ensure cardholder participation for every digital signature generated.

Slot 9d: Key Management

This certificate and its associated private key is used for encryption to assure confidentiality. This slot is used for encrypting emails or files. The end user PIN is required to perform any private key operations. Once the correct PIN has been provided, multiple private key operations may be performed without additional cardholder consent.

Slot 9e: Card Authentication

This certificate and its associated private key is used to support additional physical access applications, such as providing physical access to buildings via PIV-enabled door locks. The end user PIN is NOT required to perform private key operations for this slot.

Slots 82-95: Retired Key Management

These slots are meant for previously used key management keys to be able to decrypt earlier encrypted documents or emails.

Slot f9: Attestation

This slot is used only for attestation of other keys generated on device with instruction f9. This slot is not cleared on reset, but can be overwritten.

11.5.5 Attestation

Attestation enables you to verify that a key on the smart card application was generated on the YubiKey rather than being imported. An X.509 certificate for the key to be attested is created if the key has been generated on the YubiKey. Included in the certificate are the following extensions that provide information about the YubiKey.

Firmware

1.3.6.1.4.1.41482.3.3: Firmware version, encoded as three bytes. For example, 050100 indicates firmware version 5.1.0.

Serial Number

- 1.3.6.1.4.1.41482.3.7: Serial number of the YubiKey, encoded as an integer.
- 1.3.6.1.4.1.41482.3.8: Two bytes, the first encoding the PIN policy and the second encoding the touch policy.

PIN Policy

- 01 - never require PIN
- 02 - require PIN once per session
- 03 - always require PIN.

Touch Policy

- 01 - never require touch
- 02 - always require touch
- 03 - cache touch for 15 seconds.

Form Factor

1.3.6.1.4.1.41482.3.9: YubiKey's form factor, encoded as a one-byte octet-string.

- USB-A Keychain: 0x01
- USB-A Nano: 0x02
- USB-C Keychain: 0x03
- USB-C Nano: 0x04
- USB-C and Lightning®: 0x05
- Undefined: 0x00

11.5.6 New in YubiKey 5 FIPS Series

ATR and ATS

The ATR has been changed from “Yubikey 4” to “YubiKey” and adds support for ATS.

PIV Attestation Root CA

There are no changes in PIV attestation between the YubiKey 5 Series and the YubiKey 5 FIPS Series. You can find the root certification authority on the [PIV attestation](#) page.

11.5.7 PIV/Smart Card Deployment

The YubiKey 5 FIPS Series PIV application implements a PIV-compatible standard as defined in the [NIST SP 800-73-4](#) publication. Access to functions on the YubiKey 5 FIPS Series PIV application is restricted by the management key, the PIN and the PUK.

The management key is used for:

- Importing or generating asymmetric key pairs
- Importing x.509 certificates and associated information
- Setting the retry counters for PIN (also requires PIN) and PUK

The PIN is used to:

- Perform cryptographic operations using private keys
- Change the PIN

The PUK is used to:

- Unblock and set a new PIN for a blocked PIN
- Change the PUK

The YubiKey 5 FIPS Series PIV application has the default values:

- Management Key (010203040506070801020304050607080102030405060708)
- PIN (123456)
- PUK (12345678)

FIPS 140-2 Level 2: Placing the PIV Application in FIPS-approved Mode

To place the YubiKey 5 FIPS Series PIV application in the FIPS-approved mode of operation, change the default management key, PIN and PUK.

YubiKey 5 FIPS Series devices should be deployed using a credential management tool like Microsoft AD CS with YubiKey minidriver or a third party tool. The credential management tool will replace the default values by automatically setting a random value for the management key and PUK, allowing the end user to define the PIN.

If the YubiKey 5 FIPS Series PIV application is not being managed with a credential management tool, the management key, PIN and PUK must be changed by the crypto officer. To do so, the YubiKey Manager (ykman) can be used.

- Download the YubiKey Manager tool: <https://www.yubico.com/products/services-software/download/yubikey-manager/>
- YubiKey Manager (ykman) CLI & GUI Guide: <https://docs.yubico.com/ykman/>

To **change the management key**, use the command:

```
ykman piv access change-management-key -m010203040506070801020304050607080102030405060708 / -a<algorithm> -n<management key>
```

where `<management key>` is the new management key and `<algorithm>` is the key type [Triple-DES, AES-128, AES-192 or AES-256].

To **change the PIN**, use the command:

```
ykman piv access change-pin -P123456 -n<PIN>
```

where `<PIN>` is the new PIN.

To **change the PUK**, use the command:

```
ykman piv access change-puk -p12345678 -n<PUK>
```

where `<PUK>` is the new PUK.

11.6 FIPS Level 1 vs FIPS Level 2

The YubiKey 5 FIPS Series is certified in two modes of operations - one configuration which meets the requirements for FIPS Level 1, and a second, more restricted configuration that meets the requirements for FIPS Level 2.

The FIPS Level 2 configuration renders keys in the YubiKey 5 FIPS Series capable of being a component in a framework meeting the highest levels of authentication assurance. However, not every deployment requires this level of security. In cases where a FIPS-certified device is required, but a lower level of assurance is acceptable, the FIPS Level 1 configuration can be used. This provides a user experience like the standard YubiKey 5 Series user experience.

11.6.1 FIPS Initialization Comparison: Level 1 vs Level 2

The FIPS Level 2 requirements include all the those for Level 1. Therefore the FIPS Level 2 column in the table below lists only the differences.

YubiKey Function	FIPS Level 1	FIPS Level 2
Touch-Triggered OTP	If writing a configuration to a slot over NFC, use a secure channel.	Set Access code for both OTP slots. If updating a configuration of either OTP slot or the NDEF behavior, use a secure channel.
OATH	If writing a credential over NFC, use a secure channel.	Set the Management key. When setting the Management key over USB or NFC, use a secure channel. When writing a credential over USB or NFC, use a secure channel.
PIV	If importing a key or setting the management key, use a secure channel.	Change Management key, PIN and PUK from default values. For any operation with the PIV function over NFC, use a secure channel.
U2F	No additional requirements	Must be not be used. Recommendation Disable and use the FIDO2 function instead.
FIDO2	No additional requirements	Set a PIN. Set Credential Protection to level 2 for all discoverable credentials. Credential Registration is not allowed over NFC.
Secure Channel	Change the default transport keys from default	No additional requirements

For more information on secure channel requirements from NIST, see NIST SP 800-63-C and NIST SP 800-63B.

To get in touch with Yubico Support, [click here](#).

YUBIKEY 5 CSPN SERIES SPECIFICS

12.1 CSPN Mode Configuration

The YubiKey 5 Series supports a variety of applications, modes and operations. Technical descriptions of all of these are available from the [Yubico website](#).

Additionally, as described in the YubiKey 5 CSPN security target [RD9], the YubiKey can also be used in a CSPN approved mode of operation.

The specific configurations required in order to achieve a CSPN approved mode are described in the sections below, divided by application.

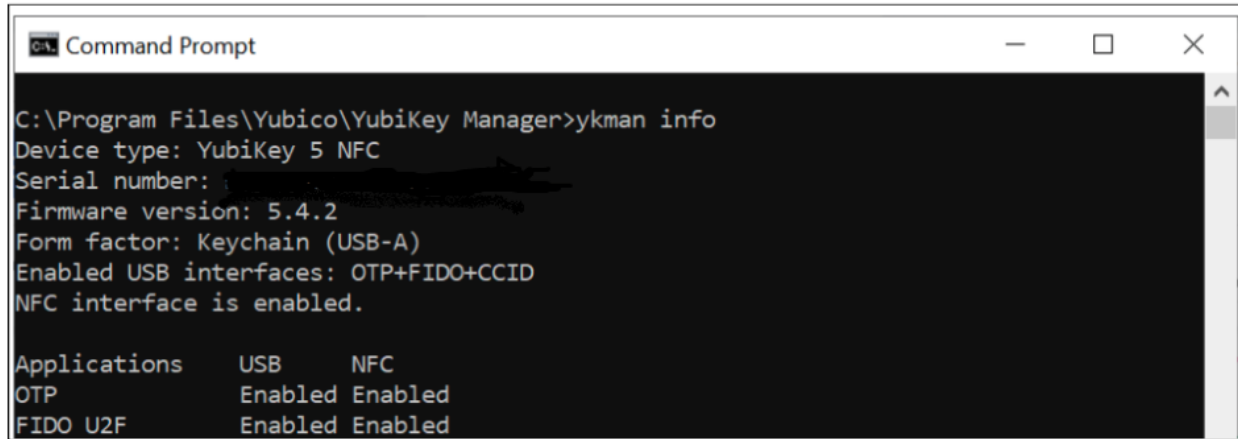
- *One-Time Password - OTP*
 - *Yubico OTP*
 - *Challenge-Response*
 - *Static Password*
 - *OATH-HOTP*
- *OATH*
- *FIDO U2F*
- *FIDO2*
- *PIV*

For each section there is a summary of the YubiKey application, how to operate it in a CSPN approved mode, and how the application can be technically configured.

12.1.1 Listing the Applications on the YubiKey 5

To obtain a list of all applications on the YubiKey 5, you can use ykman's command line [YubiKey Manager](#). To do so, in a command prompt, execute the command `ykman info`.

The output will contain general information about the YubiKey 5, such as the current firmware version, but also all of the available applications, both enabled and disabled. (The Security Domain application is hidden for the user and therefore not listed by YKMan.) An example of this command is shown in the screenshot below.



```
C:\Program Files\Yubico\YubiKey Manager>ykman info
Device type: YubiKey 5 NFC
Serial number:
Firmware version: 5.4.2
Form factor: Keychain (USB-A)
Enabled USB interfaces: OTP+FIDO+CCID
NFC interface is enabled.

Applications      USB      NFC
OTP               Enabled Enabled
FIDO U2F          Enabled Enabled
```

Fig. 1: **Figure 1** - Example of listing the applications on a YubiKey 5

12.1.2 Password Strength

It is highly recommended to adhere to [ANSI's guidelines](#) on password strength whenever applicable, as it pertains to any of the YubiKey 5 applications.

12.1.3 Configuration Environment

With regards to the configuration of the YubiKey, it can be performed in two different areas:

- If the keys of an application are generated by the secured microcontroller, the YubiKey 5 is considered as placed in a public area.
- If the keys of an application are loaded into the secured microcontroller, the YubiKey 5 is considered as placed in a secure area with restricted access.

12.2 One-Time Password - OTP

The YubiKey 5 Series OTP application supports four protocols:

- *Yubico OTP*
- *Challenge-Response*
- *Static Password*
- *OATH-HOTP*

The configuration required to achieve CSPN-approved mode is described in the sections below.

12.2.1 Yubico OTP

Feature Summary

The Yubico OTP scheme is a proprietary algorithm based on symmetric AES encryption. To generate a Yubico OTP, the following parameters must be set:

- Public ID (1-16 bytes modhex)
- Private ID (6 bytes hexadecimal)
- Secret Key (16 bytes)

The Public ID generally represents the serial number of the YubiKey, but may be set to a different value. The Private ID is an optional secret field that may be included as an input parameter to the OTP generation algorithm. By default, when this parameter is not configured, its value is set to zero. The Secret Key is an AES-128 key which must be shared between the YubiKey 5 and the verification server by the user, during the configuration of the protocol's credentials.

The touch sensor is always used when generating a Yubico OTP, and is considered part of the standard operating procedure.

For more information about Yubico OTP, see [Yubico's website](#).

CSPN Approved Mode

To operate the YubiKey 5 application Yubico OTP in a CSPN approved mode, the user must first be identified by a first factor authentication scheme (e.g. username/password). The details for such an authentication scheme go beyond the scope of this document however.

Once a Yubico OTP application has been configured, an access code must be set in order to protect the key material and configuration. More details for such a configuration are described in the section below.

Technical Configuration

In order to protect the Yubico OTP credentials, the command line [YubiKey Manager](#) (YKMan) may be used.

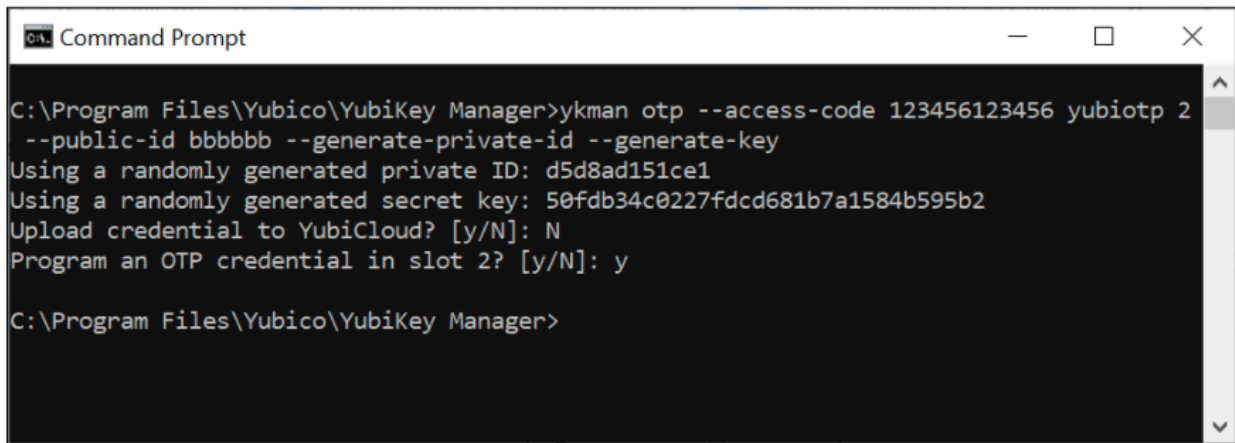
The command `ykman otp yubiotp` should be used with the option `--access-code` for protecting the credentials. The `--access-code` parameter should be set to a six byte long hex value.

An example command line interaction for creating protected Yubico OTP credentials with YKMan is depicted in the screenshot below.

A code is now required for any operations that require access to the Yubico OTP credentials:

- Delete credentials: `ykman otp --access-code <value> delete [1|2]`
- Change the settings: `ykman otp --access-code <value> settings [OPTIONS] [1|2]`

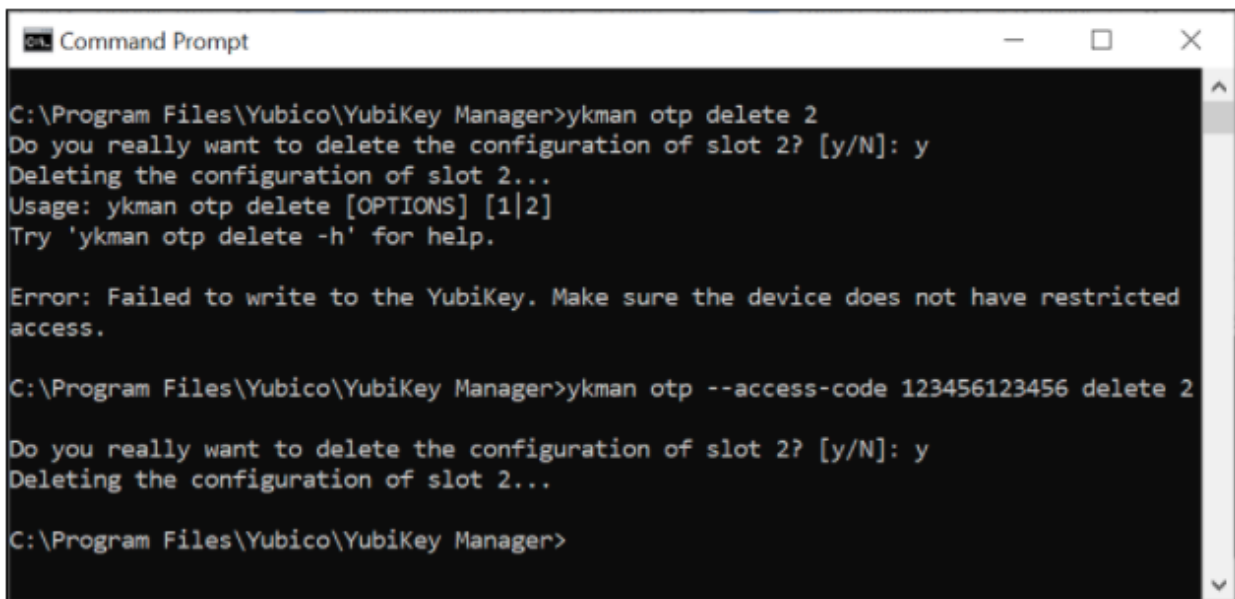
For instance, it is not possible to now delete the Yubico OTP credentials without providing the correct access code. The screenshot below is another example of how to use the YkMan command line for deleting protected Yubico OTP credentials. The first attempt fails because no `--access-code` is provided, but the second attempt succeeds when the flag `--access-code` is set.



```
Command Prompt
C:\Program Files\Yubico\YubiKey Manager>ykman otp --access-code 123456123456 yubiotp 2
--public-id bbbbbb --generate-private-id --generate-key
Using a randomly generated private ID: d5d8ad151ce1
Using a randomly generated secret key: 50fdb34c0227fdcd681b7a1584b595b2
Upload credential to YubiCloud? [y/N]: N
Program an OTP credential in slot 2? [y/N]: y

C:\Program Files\Yubico\YubiKey Manager>
```

Fig. 2: **Figure 2** - Example of configuring protected Yubico OTP credentials



```
Command Prompt
C:\Program Files\Yubico\YubiKey Manager>ykman otp delete 2
Do you really want to delete the configuration of slot 2? [y/N]: y
Deleting the configuration of slot 2...
Usage: ykman otp delete [OPTIONS] [1|2]
Try 'ykman otp delete -h' for help.

Error: Failed to write to the YubiKey. Make sure the device does not have restricted
access.

C:\Program Files\Yubico\YubiKey Manager>ykman otp --access-code 123456123456 delete 2
Do you really want to delete the configuration of slot 2? [y/N]: y
Deleting the configuration of slot 2...

C:\Program Files\Yubico\YubiKey Manager>
```

Fig. 3: **Figure 3** - Example of deleting protected Yubico OTP credentials

12.2.2 Challenge-Response

Feature Summary

The Challenge-Response protocol is based on the HMAC-SHA-1 algorithm. The relying party sends a challenge to the YubiKey 5, and the device then responds with a hash of that challenge. The secret key used in the HMAC-SHA-1 is pre-loaded by the user onto the YubiKey 5 during configuration. It is also possible to configure whether touching the sensor of the YubiKey 5 is required for each Challenge-Response request. The Challenge-Response protocol is used as a second factor in the authentication process.

For more information on the challenge-response YubiKey application, see [Yubico's website](#).

CSPN Approved Mode

To operate the YubiKey 5 in a CSPN approved mode, the user must first be identified by a first factor authentication scheme (e.g. username/password). The details for such an authentication scheme go beyond the scope of this document however.

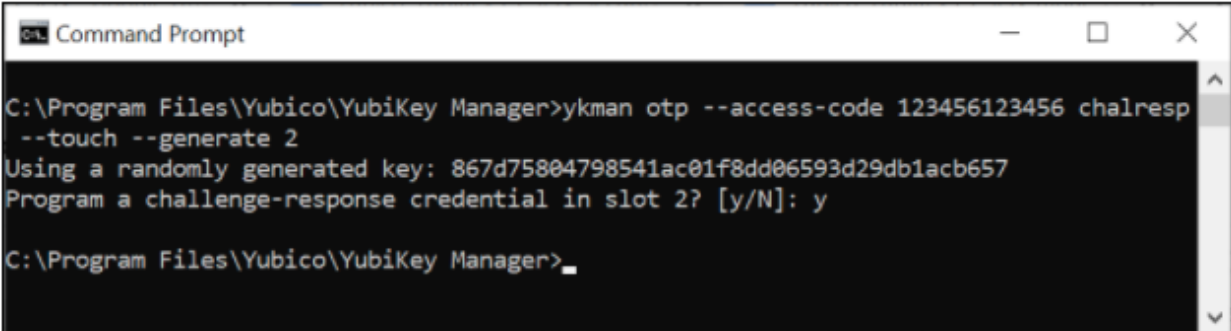
Furthermore, the usage of the YubiKey 5 touch sensor must be set to required when configuring the Challenge-Response application. Finally, when the Challenge-Response application is enabled on the YubiKey 5, an access code must be set in order to protect both the secret key and configuration. More details for such a configuration is described in the section below.

Technical Configuration

In order to protect the Challenge-Response credentials and enforce the touch sensor, the command line [YubiKey Manager](#) (YKMan) may be used.

The command `ykman otp chalresp` should be used with the option `--access-code` for protecting the credentials and `--touch` for requesting proof of user presence. The `--access-code` parameter should be set to a six byte long hex value.

An example command line interaction for creating protected Challenge-Response credentials requiring touch with YKMan is depicted in the screenshot below.



```

C:\Program Files\Yubico\YubiKey Manager>ykman otp --access-code 123456123456 chalresp
--touch --generate 2
Using a randomly generated key: 867d75804798541ac01f8dd06593d29db1acb657
Program a challenge-response credential in slot 2? [y/N]: y

C:\Program Files\Yubico\YubiKey Manager>_

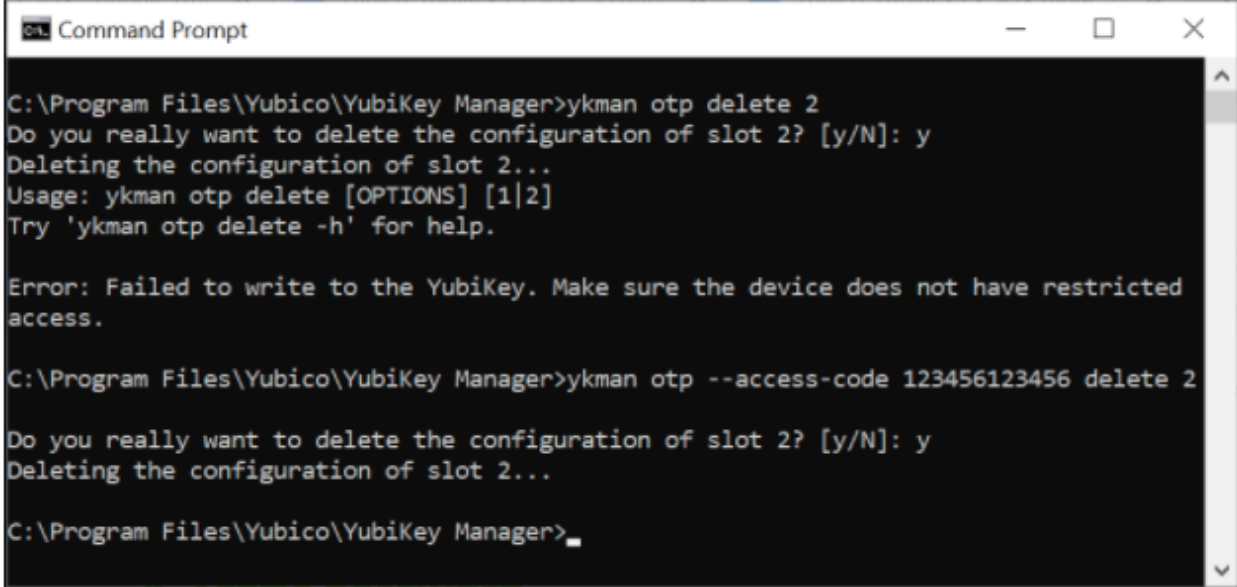
```

Fig. 4: **Figure 4** - Example of configuring protected Challenge-Response credentials with touch sensor

A code is now required for any operations that require access to the Challenge-Response credentials:

- Delete credentials: `ykman otp --access-code <value> delete [1|2]`
- Change the settings: `ykman otp --access-code <value> settings [OPTIONS] [1|2]`

For instance, it is not possible to now delete the Challenge-Response credentials without providing the access code. The screenshot below is an example of how to use the YKMan command for deleting protected Challenge-Response credentials. The first attempt fails because no `--access-code` is provided, but the second attempt succeeds when the flag `--access-code` is set.



```
C:\Program Files\Yubico\YubiKey Manager>ykman otp delete 2
Do you really want to delete the configuration of slot 2? [y/N]: y
Deleting the configuration of slot 2...
Usage: ykman otp delete [OPTIONS] [1|2]
Try 'ykman otp delete -h' for help.

Error: Failed to write to the YubiKey. Make sure the device does not have restricted
access.

C:\Program Files\Yubico\YubiKey Manager>ykman otp --access-code 123456123456 delete 2
Do you really want to delete the configuration of slot 2? [y/N]: y
Deleting the configuration of slot 2...

C:\Program Files\Yubico\YubiKey Manager>
```

Fig. 5: **Figure 5** - Example of deleting protected Challenge-Response credentials

12.2.3 Static Password

Feature Summary

The static password application allows for the storage of a complete or partial static password. The password will be replayed in the clear once the user touches the YubiKey 5 sensor. The static password is used as a second factor in the authentication process.

For more information on YubiKey application for static passwords, see [Yubico's website](#).

CSPN Approved Mode

To operate the YubiKey 5 in a CSPN approved mode, the user must only store one portion of the password within the YubiKey 5 and keep the remaining portion of the password in a different, but also secure location. The user should then reconstruct the complete password by combining the portion from the YubiKey with the other portion stored elsewhere, and then authenticate in conjunction with their username. The overall details for such a password splitting scheme go beyond the scope of this document however, as only the portion of the password to be stored within the YubiKey 5 is described.

The touch sensor is always used when displaying a portion of a static password, and is considered part of the standard operating procedure.

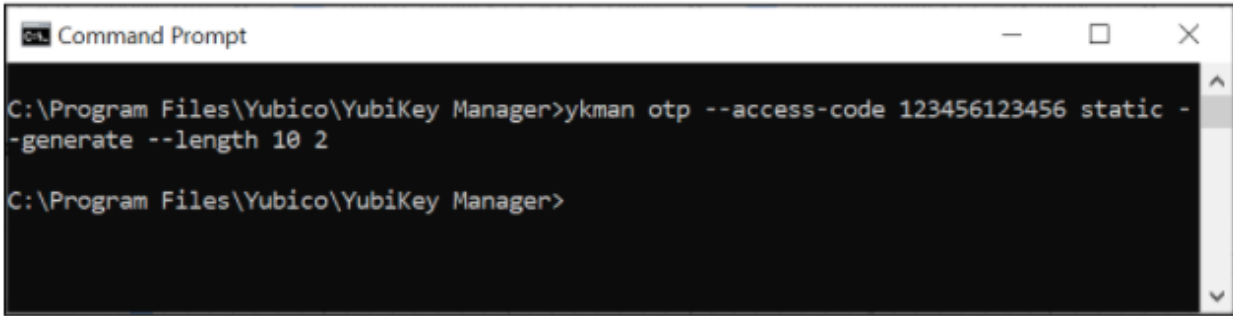
When the static password application is configured, an access code must be set in order to protect both the static password and configuration. More details for such a configuration are described in the section below.

Technical Configuration

In order to protect the static password, the command line **YubiKey Manager** (YkMan) may be used.

The command `ykman otp static`, should be used with the option `--access-code` for protecting the static password. The `--access-code` parameter should be set to a six byte long hex value.

An example command line interaction for creating a protected static password with YkMan is depicted in the screenshot below.



```
Command Prompt
C:\Program Files\Yubico\YubiKey Manager>ykman otp --access-code 123456123456 static -generate --length 10 2
C:\Program Files\Yubico\YubiKey Manager>
```

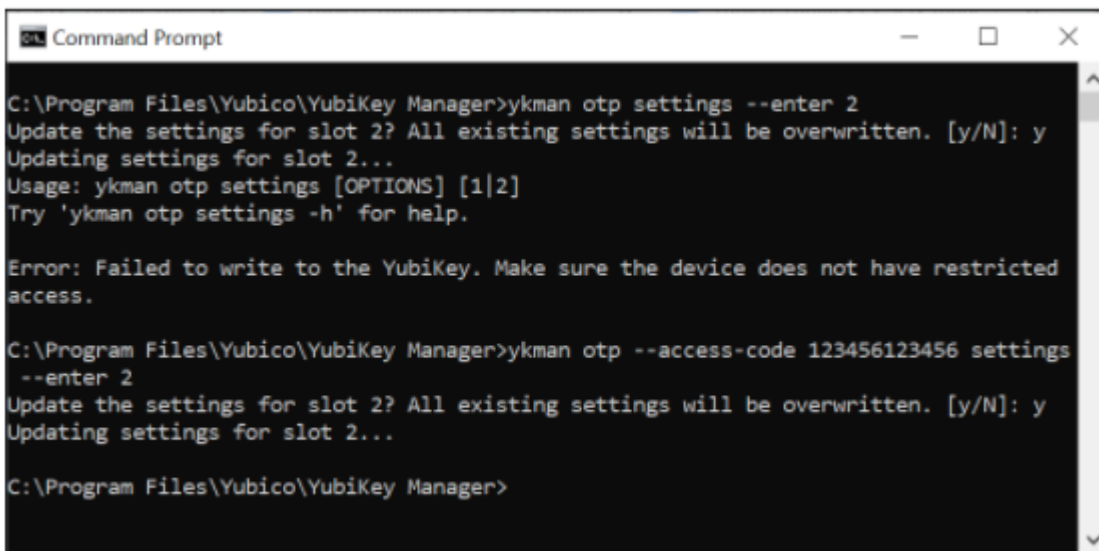
Fig. 6: **Figure 6** - Example of configuring a protected static password

A code is now required for any operations that require access to the static password:

- Delete static password: `ykman otp --access-code <value> delete [1|2]`
- Change the settings: `ykman otp --access-code <value> settings [OPTIONS] [1|2]`

For instance, it is not possible to now change the static password settings without providing the access code.

The screenshot below is an example of how to use the YkMan command line for changing the settings of a protected static password. The first attempt fails because no `--access-code` is provided, but the second attempt succeeds when the flag `--access-code` is set.



```
Command Prompt
C:\Program Files\Yubico\YubiKey Manager>ykman otp settings --enter 2
Update the settings for slot 2? All existing settings will be overwritten. [y/N]: y
Updating settings for slot 2...
Usage: ykman otp settings [OPTIONS] [1|2]
Try 'ykman otp settings -h' for help.

Error: Failed to write to the YubiKey. Make sure the device does not have restricted access.

C:\Program Files\Yubico\YubiKey Manager>ykman otp --access-code 123456123456 settings --enter 2
Update the settings for slot 2? All existing settings will be overwritten. [y/N]: y
Updating settings for slot 2...

C:\Program Files\Yubico\YubiKey Manager>
```

Fig. 7: **Figure 7** - Example of changing a protected static password

12.2.4 OATH-HOTP

Feature Summary

The OATH-HOTP protocol is implemented according to RFC 4226, i.e. “An HMAC-Based One-Time Password Algorithm”, [RD5]. The algorithm underpinning this application on the YubiKey 5 is HMAC-SHA-1. The user may choose the length of the OTP (either 6 or 8 digits) and the initial counter value. The OATH-HOTP protocol is used as a second factor in the authentication process.

The touch sensor is always used when generating the OATH-HOTP, and is considered part of the standard operating procedure.

For more information on the YubiKey application OATH-HOTP see [Yubico’s website](#).

CSPN Approved Mode

To operate the YubiKey 5 in a CSPN approved mode, the user must first be identified by a first factor authentication scheme (e.g. username/password). The details for such a first factor authentication scheme go beyond the scope of this document however.

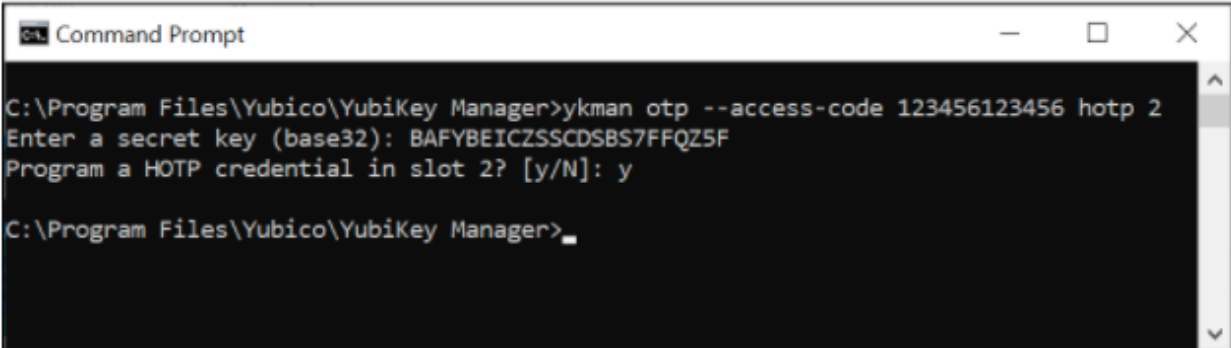
When the OATH-HOTP application is enabled on the YubiKey 5, an access code must be set to protect the initial counter value and configuration. More details for such a configuration are described in the section below.

Technical Configuration

In order to protect the OATH-HOTP credentials, the command line [YubiKey Manager](#) (YkMan) may be used.

The command `ykman otp hotp` should be used with the option `--access-code` for protecting the OATH-HOTP credentials. The `--access-code` parameter should be set to a six byte long hex value.

An example command line interaction for creating a protected OATH-HOTP with YKMan is depicted in the screenshot below.



```
Command Prompt
C:\Program Files\Yubico\YubiKey Manager>ykman otp --access-code 123456123456 hotp 2
Enter a secret key (base32): BAFYBEICZSSCDSBS7FFQZ5F
Program a HOTP credential in slot 2? [y/N]: y
C:\Program Files\Yubico\YubiKey Manager>
```

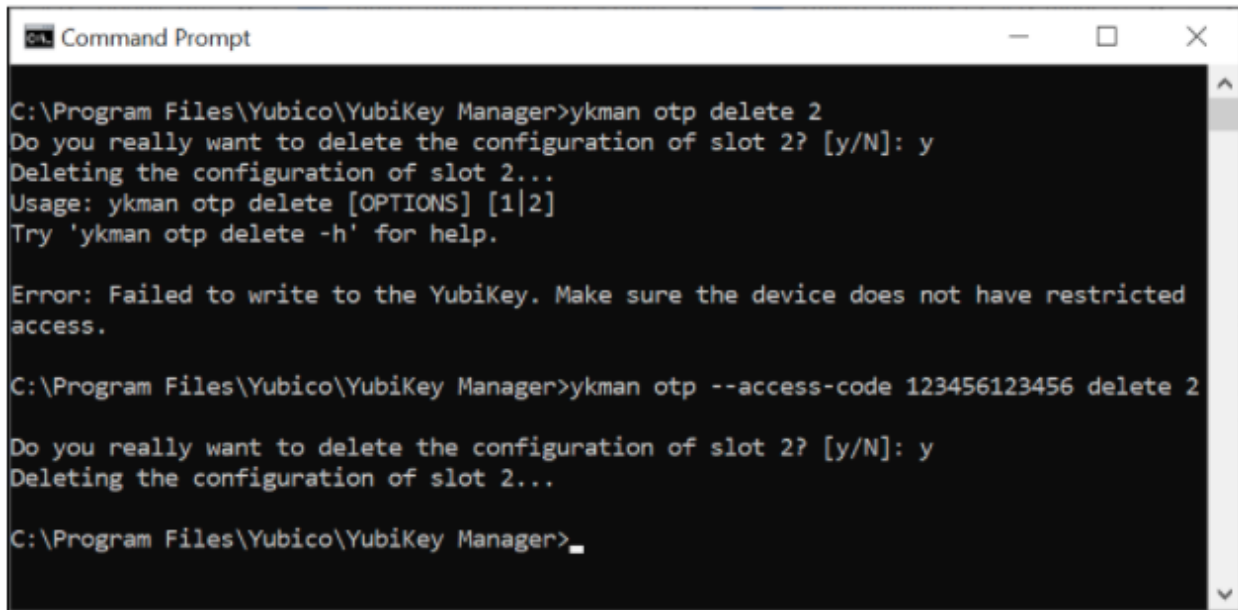
Fig. 8: **Figure 8** - Example of configuring protected OATH-HOTP credentials

A code is now required for any operations that require access to the OATH-HOTP credentials:

- Delete credentials: `ykman otp --access-code <value> delete [1|2]`
- Change the settings: `ykman otp --access-code <value> settings [OPTIONS] [1|2]`

For instance, it is not possible to now delete the OATH-HOTP credentials without providing the access code.

The screenshot below is an example of how to use the YkMan command line for deleting protected OATH-HOTP credentials. The first attempt fails because no `--access-code` is provided, but the second attempt succeeds when the flag `--access-code` is set.



```
Command Prompt
C:\Program Files\Yubico\YubiKey Manager>ykman otp delete 2
Do you really want to delete the configuration of slot 2? [y/N]: y
Deleting the configuration of slot 2...
Usage: ykman otp delete [OPTIONS] [1|2]
Try 'ykman otp delete -h' for help.

Error: Failed to write to the YubiKey. Make sure the device does not have restricted
access.

C:\Program Files\Yubico\YubiKey Manager>ykman otp --access-code 123456123456 delete 2
Do you really want to delete the configuration of slot 2? [y/N]: y
Deleting the configuration of slot 2...

C:\Program Files\Yubico\YubiKey Manager>_
```

Fig. 9: **Figure 9** - Example of deleting protected OATH-HOTP credentials

12.3 OATH

12.3.1 Feature Summary

The OATH application allows for managing two types of OTP over the CCID interface:

- HMAC-Based One Time Password (HOTP)
- Time-Based One Time Password (TOTP)

A maximum of 32 credentials¹ can be stored within the YubiKey's OATH application. The software tool [Yubico Authenticator](#) may be used to configure and use this application.

A password may also be set to protect the OATH credentials, and if this is configured, the password will be required to unlock the application, which can then be used to generate any number of OTPs for the remainder of the session (i.e. until application is deselected).

During the enrollment of credentials, it is also possible to configure whether touching the sensor of the YubiKey 5 is required for each OTP generation.

¹ A credential is a configuration of the OTP linked to a unique key.

12.3.2 CSPN Approved Mode

The OATH-HOTP/TOTP protocol is used as a second factor in the authentication process. To operate the YubiKey 5 in a CSPN approved mode, the user must first be identified by a first factor authentication scheme (e.g. username/password). The details for such a first factor authentication scheme go beyond the scope of this document however.

When the OATH-HOTP/TOTP application is enabled on the YubiKey 5, a password can also be set to protect the OATH credentials. More details for such a configuration are described in the section below.

12.3.3 Technical Configuration

In order to protect the OATH-HOTP/TOTP credentials with a password, the [Yubico Authenticator](#) should be installed and used for the configuration.

In order to set the password, launch the [Yubico Authenticator](#) application, select File from the menu and finally the option Set Password. In the dialog box that appears, enter a new password and confirm it. This configuration will protect all OATH-HOTP/TOTP credentials with the same nominated password.

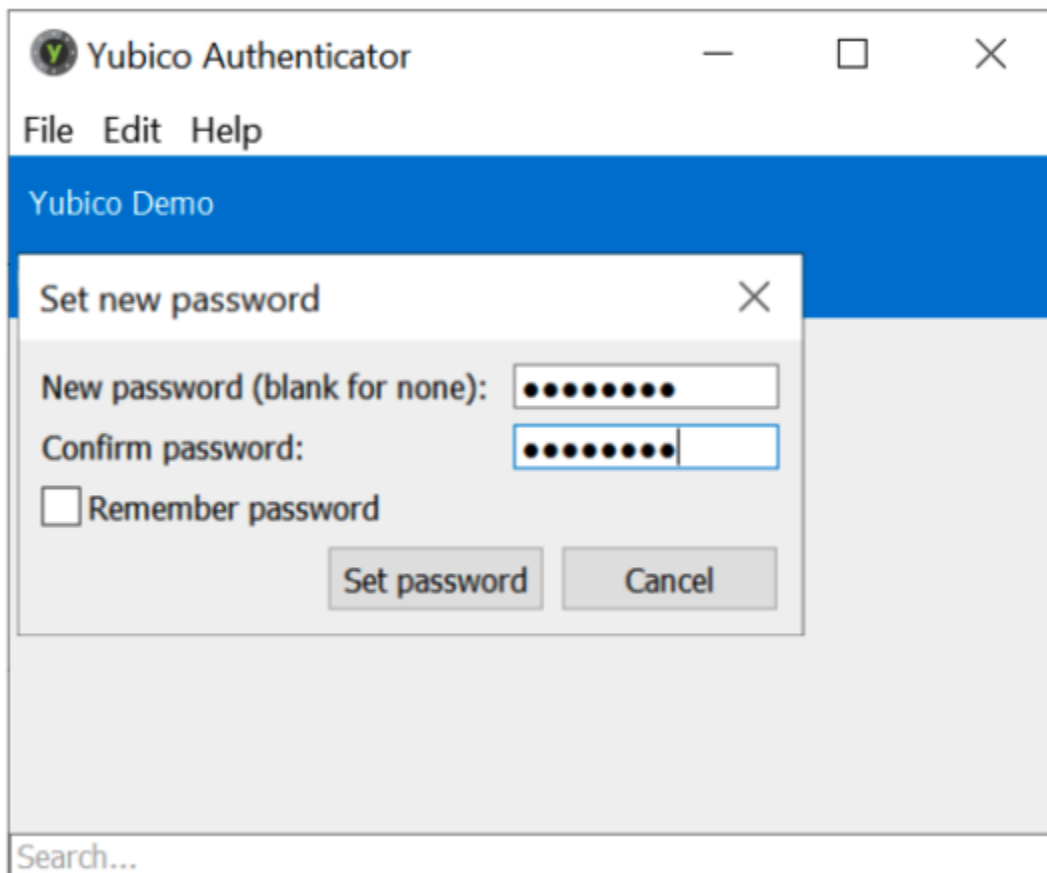


Fig. 10: **Figure 10** - Example of protecting the OATH-HOTP/TOTP credentials with a password

When [Yubico Authenticator](#) is used for generating an OATH one-time password, the user must enter the password each time in order to unlock the credentials.

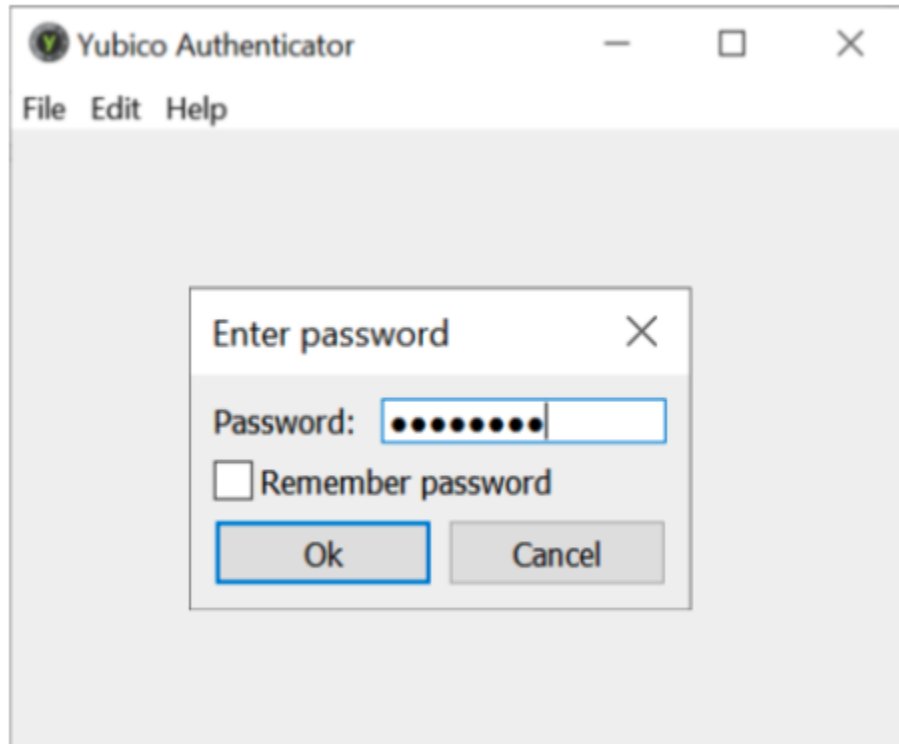


Fig. 11: **Figure 11** - Example of unlocking the OATH-HOTP/TOTP credentials

12.4 FIDO U2F

12.4.1 Feature Summary

The YubiKey 5 Series supports FIDO Universal 2nd Factor (U2F), which is defined in [RD7]. On a high level, the FIDO U2F protocol comprises both the registration and the authentication process but is only used as a second factor in the authentication process.

For more information on the YubiKey application FIDO U2F see [Yubico's website](#).

12.4.2 CSPN Approved Mode

To operate the YubiKey 5 in a CSPN approved mode, the user must first be identified with a first factor authentication scheme (e.g. username/password) according to the FIDO U2F standard [RD7]. The details for such a first factor authentication scheme go beyond the scope of this document however.

As part of the registration process, the user must touch the YubiKey 5 sensor when the browser or application prompts for it. Furthermore, the user must also touch the YubiKey 5 when the browser or application requests for it during the authentication process.

12.4.3 Technical Configuration

No additional configuration is needed to achieve a CSPN approved mode, assuming the YubiKey 5 has been correctly enrolled against a U2F compatible relying party.

12.5 FIDO2

12.5.1 Feature Summary

The FIDO2 protocol is an amalgamation of two standards: W3C WebAuthn (for the communication between the client and the relying party) and CTAP2 (for accessing the authenticator from the client). On a high level, the FIDO2 protocol comprises both the registration and the authentication process.

FIDO2 is an update of FIDO U2F and is defined in [RD3]. It takes into account PIN management, in addition to the new standardized protocols, WebAuthn [RD8] and CTAP2.

12.5.2 CSPN Approved Mode

The FIDO2 protocol can be used in two different CSPN modes of operation:

- FIDO2 with a PIN code set on the YubiKey 5 (see *FIDO2 With PIN Code*), or
- FIDO2 without a PIN code set on the YubiKey 5 (see *FIDO2 Without PIN Code*)

12.5.3 FIDO2 With PIN Code

If WebAuthn User Verification is set to 'Required' by the WebAuthn relying party when the user registers the YubiKey 5 as a FIDO2 device, it will prompt the user's client to protect the FIDO2 credentials with a PIN code during the enrollment. Alternatively, the user may also use [YubiKey Manager](#) to set a PIN code which will protect the FIDO2 credentials. In both cases, the YubiKey 5 will require the user to enter a PIN code when using it for FIDO2 authentication.

As part of the registration process, the user must touch the YubiKey 5 sensor when the browser or application prompts for it. Furthermore the user must also touch the YubiKey 5 when the browser or application requests for it during the authentication process.

12.5.4 FIDO2 Without PIN Code

If WebAuthn User Verification is not enforced as recommended above, the YubiKey 5 must then be used as a second factor authentication device. To operate the YubiKey 5 in a CSPN approved mode under such a scenario, the user must first be identified with a first factor authentication scheme (e.g. username/password). The details for such a first factor authentication scheme go beyond the scope of this document however.

The YubiKey 5 will, by default, require the sensor to be touched for this configuration. As part of the registration process, the user must touch the YubiKey 5 sensor when the browser or application prompts for it. Furthermore, the user must also touch the YubiKey 5 when the browser or application requests for it during the authentication process.

12.5.5 Technical Configuration

FIDO2 With PIN Code

There are two ways to set the PIN code for the FIDO2 application on a YubiKey 5:

- The user can set the PIN code by using the tool [YubiKey Manager](#)
- The relying party (server application) can request the user's client to set the PIN code during the WebAuthn registration

In addition to the PIN being set on the YubiKey, the touch sensor is required by default for FIDO2.

Set FIDO2 PIN Code with ykman

The [YubiKey Manager](#) may be used to set a PIN code for the FIDO2 credentials on the YubiKey 5. When a PIN code is set, all FIDO2 credentials will be protected by the same PIN code. In order to set the PIN code with [YubiKey Manager](#), select the Applications from the menu and then the FIDO2 option. In the resulting GUI which appears, press the button "Set PIN".

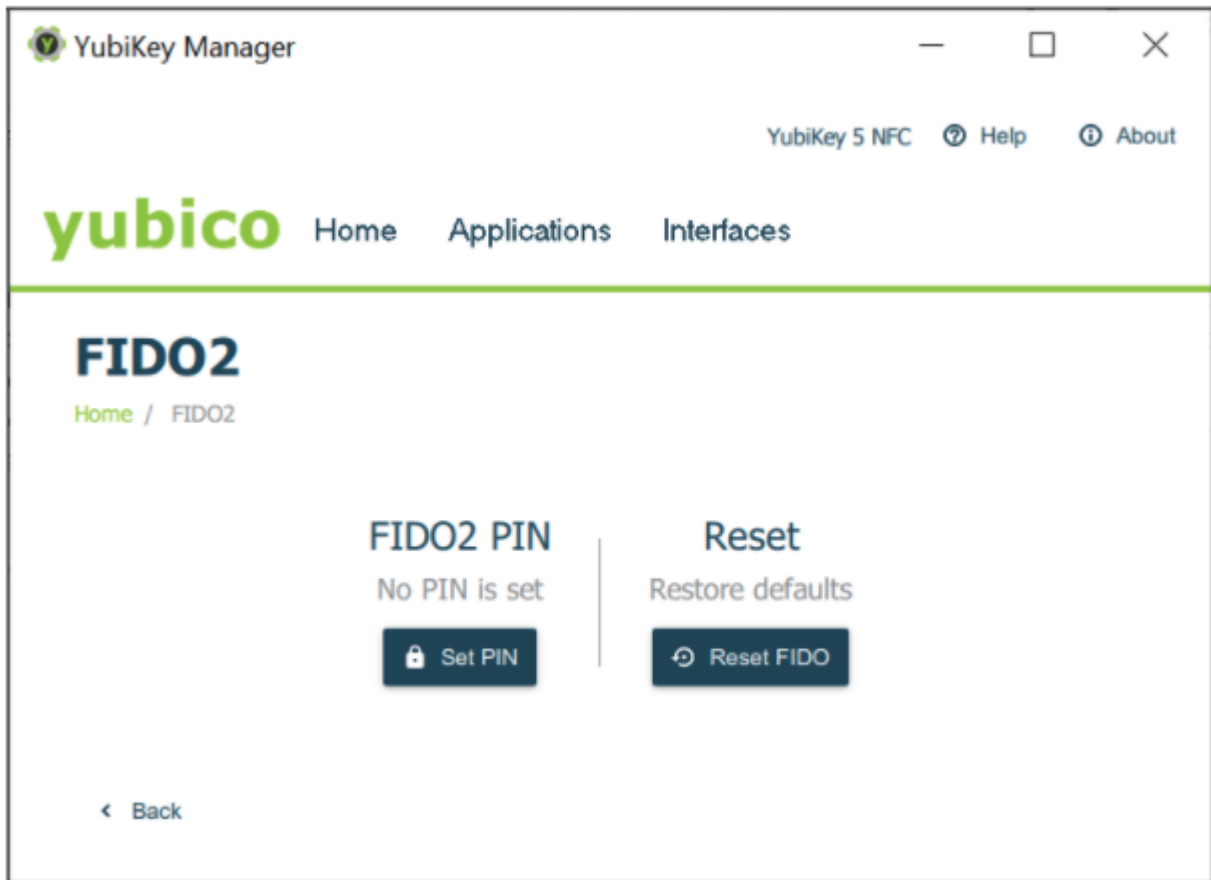


Fig. 12: **Figure 12** - Configuring the FIDO2 PIN with YubiKey Manager

In the next popup which appears, the user is prompted to set the new PIN and to confirm this PIN for the FIDO2 application.

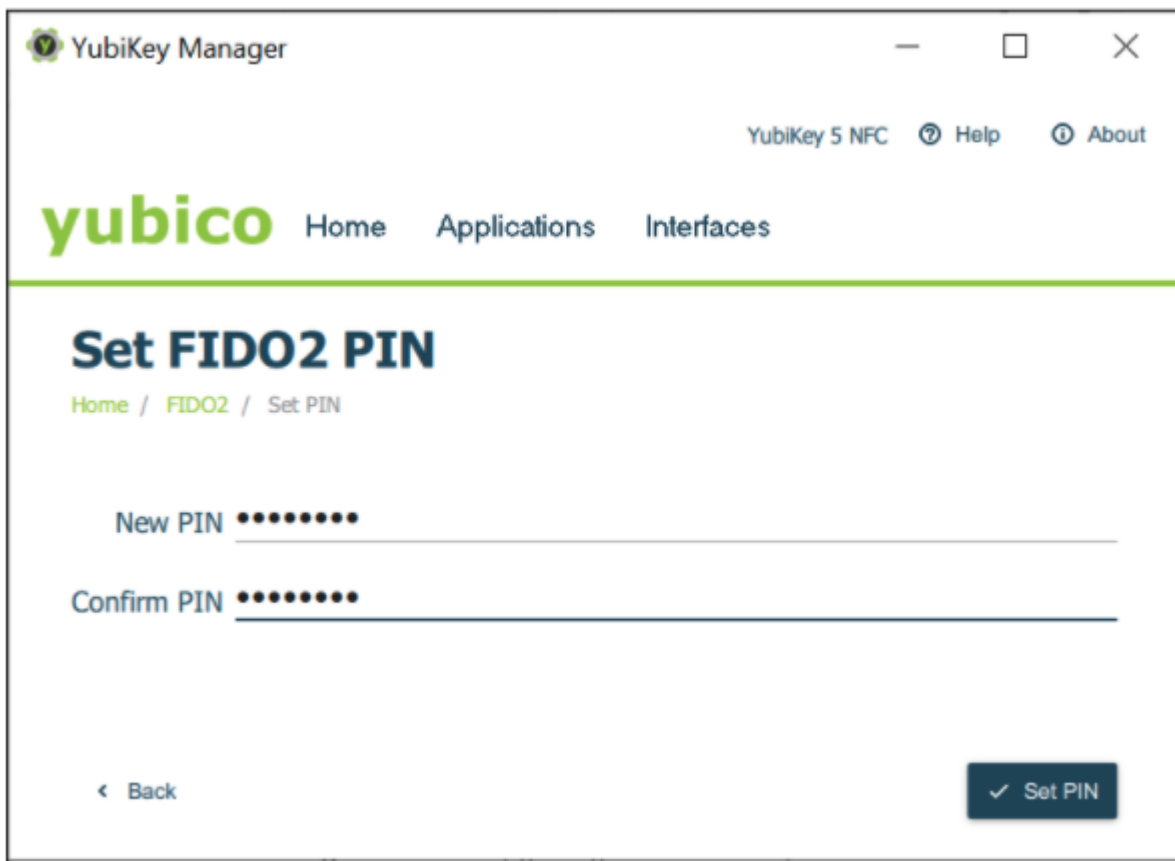


Fig. 13: **Figure 13** - Configuring the FIDO2 PIN with YubiKey Manager

Set FIDO2 PIN Code From the Relying Party

The WebAuthn relying party (authentication server) can instruct a client to set the PIN code on an authenticator during the enrollment of the FIDO2 credentials.

A client, according to the WebAuthn/FIDO2 specifications, is any user device that supports WebAuthn/FIDO2. In practice, this is a hardware device (smartphone, tablet, laptop, etc), an operating system (Microsoft Windows, Apple MacOS, Apple iOS, Android, Linux, etc) or a web browser (Google Chrome, Apple Safari, Microsoft Edge, Mozilla Firefox, etc).

If the WebAuthn MakeCredentials parameter UserVerification is set to 'Required', this will prompt the client to set the PIN code on the YubiKey 5.

The GUI for setting the FIDO2 PIN code may differ between clients. The image below is an example of using Google Chrome with Windows 10 for setting the FIDO2 PIN on a YubiKey 5.

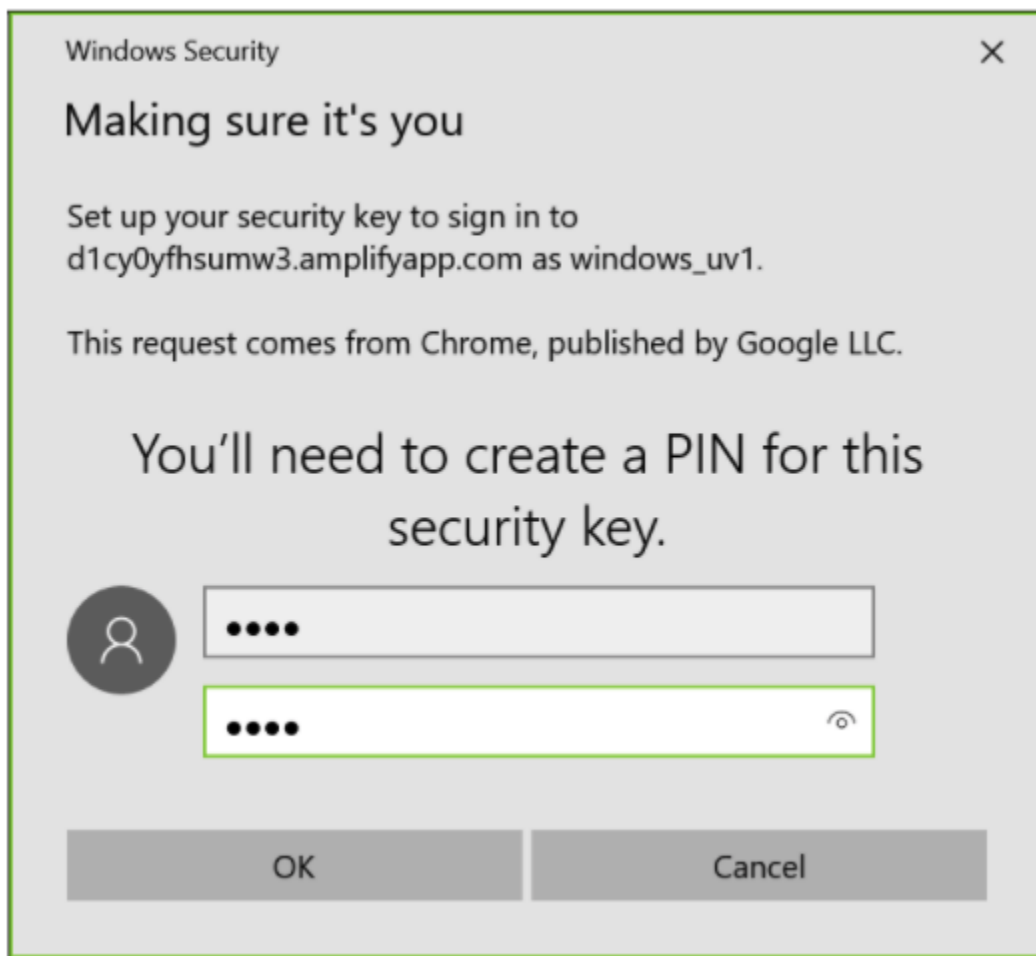


Fig. 14: **Figure 14** - Configuring the PIN code for FIDO2 with Windows 10

FIDO2 Without PIN Code

If the relying party has set the WebAuthn MakeCredentials parameter UserVerification to 'Discouraged', this will not trigger the client to set any FIDO2 PIN code on the YubiKey 5. Furthermore, if no FIDO2 PIN is set by using the [YubiKey Manager](#), then there will be no PIN set to protect the FIDO2 credentials.

However, touch will still be required, by default, for using the FIDO2 credentials during WebAuthn authentication.

When the PIN code is disabled for FIDO2 on the YubiKey 5, the CSPN approved mode is achieved by using a first factor authentication protocol in conjunction with the YubiKey 5 configured for FIDO2 and touch.

12.6 PIV

12.6.1 Feature Summary

The PIV application [RD4] can be used to authenticate, sign and decrypt. The user may, for example, use the YubiKey 5 PIV application for Windows smart card logon.

The PIV application allows for generating or importing asymmetric key-pairs (both RSA or ECC) and to store multiple X.509 certificates. In total, 24 certificate slots are available:

- Slot 9a: PIV Authentication
- Slot 9c: Digital Signature
- Slot 9d: Key Management
- Slot 9e: Card Authentication
- Slots 82-95 (hexadecimal): Retired Key Management
- Slot f9: Attestation

User verification under PIV is achieved with a PIN and a management key (Triple-DES or AES key) is used for various oversight functions. The PIN must be set to a value between 6 and 8 bytes, while the maximum number of retries must be set in the range of 1 to 255 with a default value of 3.

To specify how often the PIN needs to be entered in order to access the credentials in a given slot, a PIN policy should be set for that slot. This policy must be set upon key generation or when a key is imported, and cannot be changed at a later time.

In addition to requiring the PIN, the YubiKey 5 may also be configured to require physical contact of the touch sensor. Similar to the PIN policy, the touch policy must be set upon key generation or import.

12.6.2 CSPN Approved Mode

To operate the YubiKey's PIV application in CSPN approved mode, the PIN code, PUK code and management key must be set for the PIV application. It is imperative that the default values of these codes are also changed by the user before using the PIV application.

More details for such a configuration are described in the section below.

12.6.3 Technical Configuration

YubiKey Manager for PIN Configuration of PIV

The YubiKey Manager may be used for setting the PIN, PUK and management key on the YubiKey. In this scenario, a YubiKey 5 with default settings is assumed.

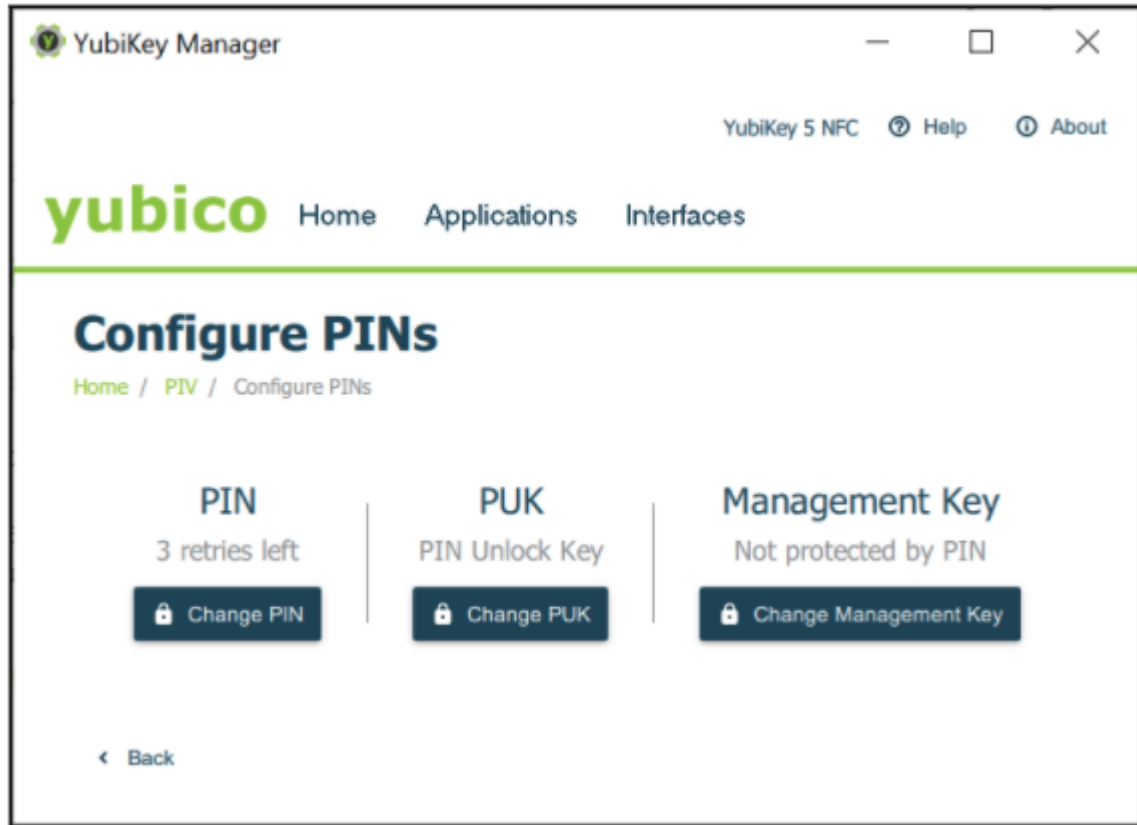


Fig. 15: **Figure 15** - Configuring the PIN, PUK and management key for PIV

Changing the PIN Code

The PIN is used during normal operation to authorize an action such as creating a digital signature with any of the stored keys. Entering an incorrect PIN too many times, which exceeds the retry counter, will cause the PIN to become blocked, thereby rendering the PIV features unusable. The PIN must be at least 6 characters and can contain any symbol, although for cross-platform portability it is recommended to only use decimal digits. There is a limit of 8 bytes for a PIN, which allows for up to 8 ASCII characters. By default the PIN code is set to “123456”.

The PIN code is changed by pressing the “Change PIN” button in the “Configure PINs” dialog box. The resulting popup which will appear in YubiKey Manager, is pictured below.

The current (default) PIN must be changed to a new PIN with a length of 6-8 digits. The user must enter the current PIN, nominate a new PIN, confirm it, and then press the “Change PIN” button.

The default PIN code mentioned above is pre-configured for slots 9a, 9c and 9d. With regards to slot 9e, the PIN policy needs to be set to enforced with the command line tool YubiKey Manager when generating or importing the key-pair on the YubiKey 5. An example of how to set the PIN policy when using the command line tool YubiKey Manager with the parameter `--pin-policy` is shown below:

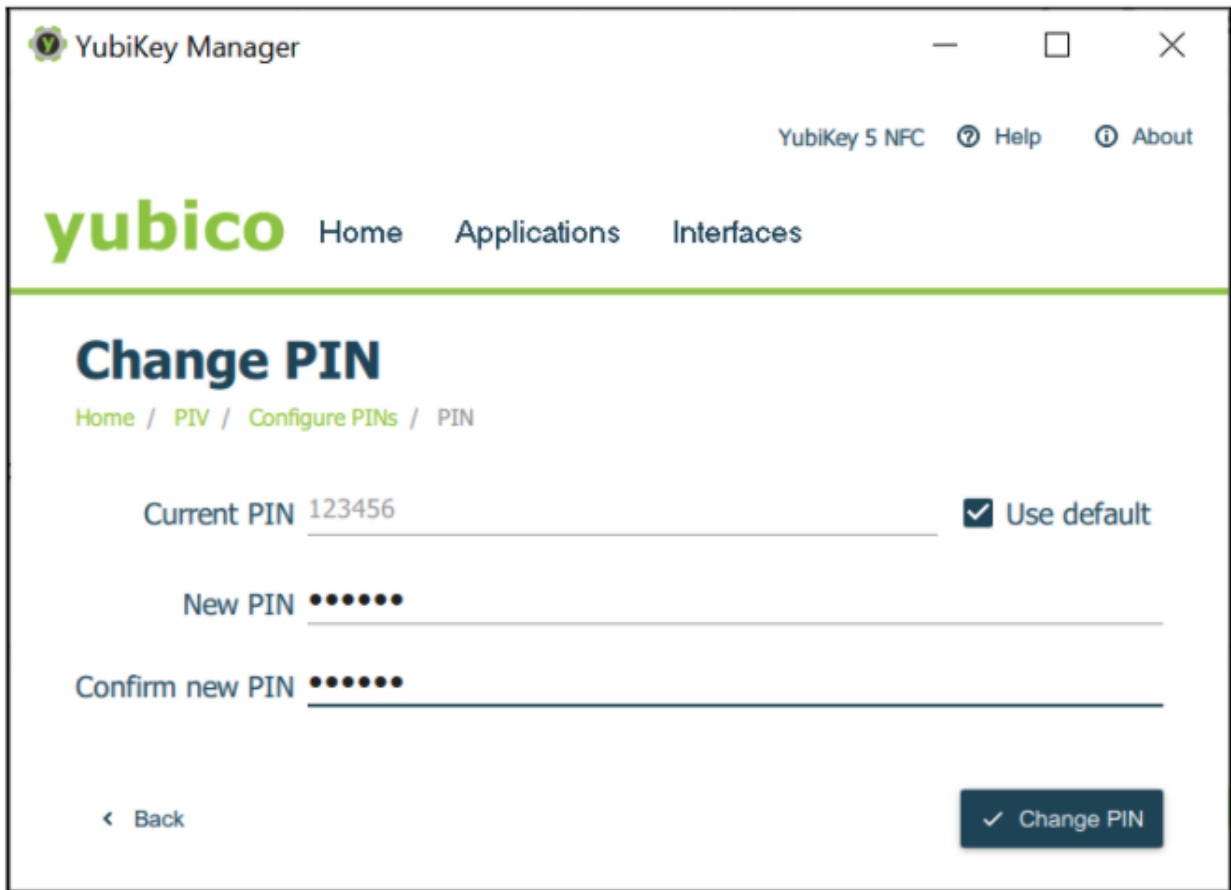


Fig. 16: **Figure 16** - Changing the PIN code for PIV


```
ykman piv generate-key --pin-policy always 9e -
```

Changing the PUK Code

The PUK can be used to reset the PIN if it is ever forgotten, lost or becomes blocked after the maximum number of incorrect attempts by the user. By default the PUK is set to “12345678”.

The PUK is changed by pressing the “Change PUK” button in the “Configure PINs” dialog box. The resulting popup which will appear in YubiKey Manager, is pictured below.

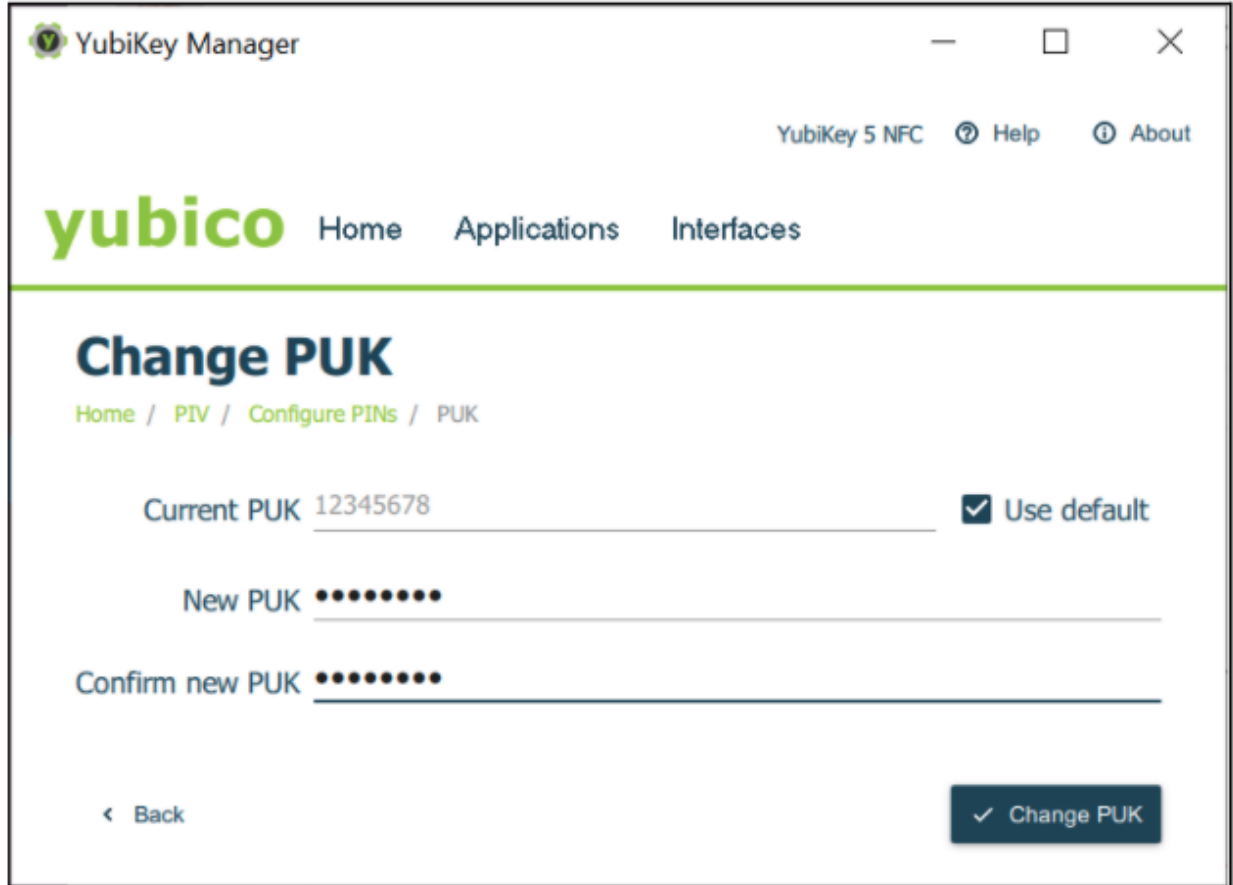
The image shows a screenshot of the YubiKey Manager application window. The title bar reads "YubiKey Manager" and includes standard window controls (minimize, maximize, close). The main content area has a header with the "yubico" logo and navigation links for "Home", "Applications", and "Interfaces". Below this, the page title is "Change PUK" with a breadcrumb trail: "Home / PIV / Configure PINs / PUK". The form contains three input fields: "Current PUK" with the value "12345678" and a checked "Use default" checkbox; "New PUK" with seven black dots; and "Confirm new PUK" with seven black dots. At the bottom left is a "Back" button with a left arrow, and at the bottom right is a "Change PUK" button with a checkmark.

Fig. 17: **Figure 17** - Changing the PUK code for PIV

The current (default) PUK must be changed to a new PUK with a length of 6-8 digits. The user must enter the current PUK, the new PUK, confirm it, and then press the “Change PUK” button.

Changing the Management Key

All PIV management operations of the YubiKey require a 24 byte 3DES or AES key, known as the management key. By default the management key is set to “010203040506070801020304050607080102030405060708”. The user should explicitly set a 24 byte key (the YubiKey PIV Manager can also generate one).

The management key is changed by pressing the “Change Management Key” button in the “Configure PINs” dialog box. The resulting popup which will appear in YubiKey Manager, is pictured below.

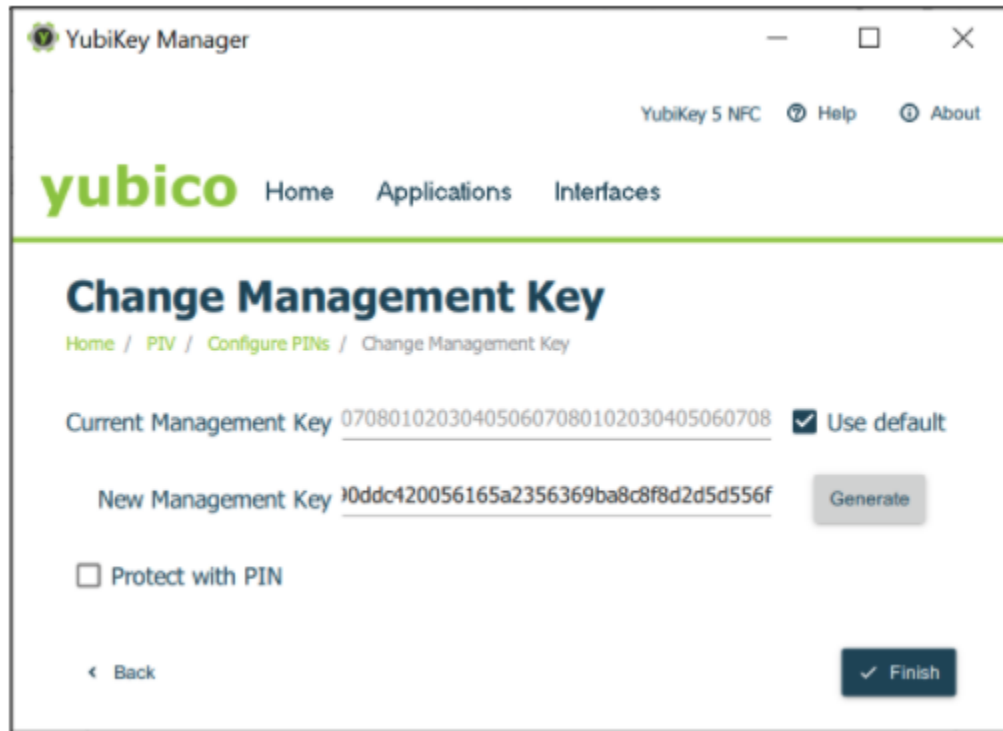


Fig. 18: **Figure 18** - Changing the management key for PIV

The current (default) management key must be changed to a new management key with a length of 48 hexadecimal digits. The user must enter the current management key, the new management key, and press the “Change management key” button.

To get in touch with Yubico Support, [click here](#).

YUBIKEY BIO SERIES SPECIFICS

13.1 How the YubiKey Bio Works

For the full technical explanation of this from a developer perspective, start with the [Yubico's WebAuthn Developer Guide](#).

Note: In the following, “credentials” will be referenced repeatedly. There are different kinds of credentials. To pursue all the distinctions, consult the [FIDO2 page on the Fido Alliance web site](#).

13.1.1 Enrollment

Before you can start using the YubiKey Bio with services and applications, you need to first set a *FIDO2 PIN* and then enroll at least one fingerprint. The YubiKey Bio needs to have the PIN as a fallback in case it cannot recognize your fingerprint.

Although there are two FIDO *applications* on the YubiKey Bio, namely FIDO2 and U2F, it is the FIDO2 PIN that is required as fallback for both. The PIN is not associated with any *site*. When the fingerprint does not work and the key falls back to the PIN, it is the *key* that needs the PIN for authentication to all sites, including U2F sites (even though U2F has no concept of PIN). With fallback to PIN, it is easy if the user is authenticating to a WebAuthn/FIDO2 site, because the browser/client app *can* prompt for the PIN. Otherwise the user must unblock biometrics by using either:

- The [YubiKey Bio start page](#)
- Yubico Authenticator for Desktop.

The “working” of the fingerprint is described in the following. For information on how and why the fingerprint might not “work”, see [Tips](#).

Risk Mitigation

To mitigate the risk of being shut out of your account or service, it is always advised to register a second YubiKey. For more information, see <https://www.yubico.com/spare/>.

Fingerprints and Templates

An enrolled fingerprint is stored on the YubiKey Bio not as an image, but in the form of a template, similar to a one-way hash. It is not possible to recreate an image of a fingerprint from a template, nor does the template ever leave the YubiKey.

After enrollment, each time you apply your fingertip to the fingerprint sensor, the key tries to match the fingerprint against the template stored on the key.

13.1.2 Parties Involved in Registration and Authentication

Closely related to *Requirements: Platform and Browser Compatibility*, registering and authenticating with a YubiKey Bio to an app or a service that supports WebAuthn or U2F involves several parties:

- The user (with their fingerprints and knowledge of the PIN)
- The YubiKey Bio
- The FIDO2 application or the U2F application on the YubiKey Bio
- The FIDO2/WebAuthn or U2F-supporting **browser** or **client**
- The service or app

All these work together. For example, if your YubiKey does not work as expected, you might be using a browser or an app that does not support FIDO2 security keys.

13.1.3 Registration

Registration of a YubiKey Bio with a site, service, or application is the same as for other YubiKeys.

13.1.4 Authentication

Depending on the protocol supported by the site or service, there are several possible user experiences (scenarios). These are described below.

User Experiences

The user experience with the YubiKey Bio is dictated by a combination of the site or service that the user is authenticating against and the browser or client. Different service and client combinations will yield different results. The user experiences are determined by the different options for developers implementing FIDO2 with the WebAuthn and CTAP protocols. Please note that the following descriptions of user scenarios are only **high-level overviews**. The experiences will change every time the various forms of support change.

Passwordless

This scenario provides the best user experience by enabling a passwordless flow backed by strong authentication. To achieve it, [discoverable credentials](#) must be used. When the user authenticates to the site or service,

1. The client or browser prompts the user to insert the YubiKey.
2. The client makes a request to the YubiKey to see if any credentials on the key have been registered for use with this site or service.
3. If the right credentials are found, the *client or browser* prompts the user to apply their fingertip to the YubiKey Bio's sensor.
 - If the fingerprint match is successful, the appropriate response is sent to the client or browser to complete authentication.
 - If the fingerprint match is unsuccessful three times in a row, the client or the browser prompts instead for the PIN. After correctly inputting the PIN, the user is then prompted to touch the key to prove presence (as opposed to verifying identity). In this situation, the YubiKey Bio behaves like any other key in the YubiKey 5 Series.

Multifactor Authentication (MFA)

When a user authenticates to the site or service,

1. The client or browser prompts the user to insert their username and password. These are what the server uses to identify the user and determine whether they have registered.
2. If username and password match the server's records, the site or service prompts the user for an additional form of identification to prove their identity. This is called **multifactor** authentication.
3. The user proves their identity *to the key* either by providing a fingerprint that the key can match to its template, or by entering the PIN.
 - If the fingerprint match is successful, the appropriate response is sent to the client or browser to complete authentication.
 - If the key is unsuccessful at matching fingerprint to template three times in a row, the YubiKey Bio goes into the biometrics blocked state, signaling this by slow constant flashing of the amber LED. The client or the browser prompts instead for the PIN and for the user to touch the key (checking for user presence). In this situation, the YubiKey Bio behaves like any other key in the YubiKey 5 Series.

U2F

This scenario only works well if the fingerprint match is successful and the user flow is the same as the multifactor flow. If the fingerprint match is unsuccessful, any prompts from the site or service are unlikely to be clear and unambiguous. The user would likely end by having to unblock the YubiKey, which can be done by visiting the [YubiKey Bio start page](#) or by using the Yubico Authenticator for Desktop.

13.1.5 Locking/Blocking

Fingerprint

If the YubiKey cannot match fingerprint to template three times in a row, fingerprint recognition is blocked. The YubiKey Bio falls back to PIN.

PIN

If you enter the wrong PIN eight times in a row, the YubiKey FIDO2 application will be **locked**, which means it cannot communicate with you or with any site or service. It indicates the blocked state by flashing its amber LED slowly and continuously. In order to restore this functionality, the FIDO2 application must be reset. For more details, see [FIDO2 PIN](#).

Unblock

Unblock the YubiKey Bio's biometric function (its ability to read fingerprints) by going to the unblocking FAQ on the [YubiKey Bio start page](#). Otherwise you can use any of the other methods given in tools-label.

Reset

You can also **reset** it, but doing so erases all the discoverable credentials on it, setting it back to factory defaults. See [Resetting Your YubiKey Bio with the Yubico Authenticator for Desktop](#).

13.1.6 Managing Credentials

If you decide to discontinue using a site or service, you can delete its discoverable credential. This frees up space on the YubiKey Bio, which can contain up to 25 such credentials.

To view the discoverable credentials on your YubiKey and delete them selectively, use the Yubico Authenticator for Desktop version 5.1.0 and above.

For more information on credentials in general, and in particular on managing them, see [Enhancements to FIDO 2 Support](#) for details.

For more **developer-oriented** information on this, see [Discoverable Credentials / Resident Keys](#) on Yubico's developer site.

13.2 Using Chrome to Enroll Fingerprints

Set a PIN and enroll the *first* fingerprint using the Chrome browser on a macOS, Linux or Chrome OS device. To enroll more fingerprints use the Chrome settings as described in [Enrolling Additional Fingerprints](#).

Note: A YubiKey is a FIDO2 *hardware* authenticator. Both Windows and Mac have *built-in* FIDO2 authenticators - i.e., software authenticators that in this case are also platform authenticators. The prompts in both Windows and Mac *might assume* you will be using their own authenticators. Therefore it is quite easy to register *their* authenticators with a site or service by mistake, without realizing that you are not registering your YubiKey. Read the prompts carefully to avoid this. And remember that the PIN is associated with the authenticator, not the site or service.

Although there are two FIDO *applications* on the YubiKey Bio, namely FIDO2 and U2F, it is the FIDO2 PIN that is required as fallback for both. The PIN is not associated with any *site*. When the fingerprint does not work and the key falls back to the PIN, it is the *key* that needs the PIN for authentication to all sites, including U2F sites (even though U2F has no concept of PIN). With fallback to PIN, it is easy if the user is authenticating to a WebAuthn/FIDO2 site, because the browser/client app *can* prompt for the PIN. Otherwise the user must unblock biometrics by using either:

- The [YubiKey Bio start page](#)

- Yubico Authenticator for Desktop.

For information on the YubiKey Bio's sensor and tips on working with fingerprints see *Tips*. For detailed information on FIDO2 PINs and their requirements, see [Understanding YubiKey PINs](#).

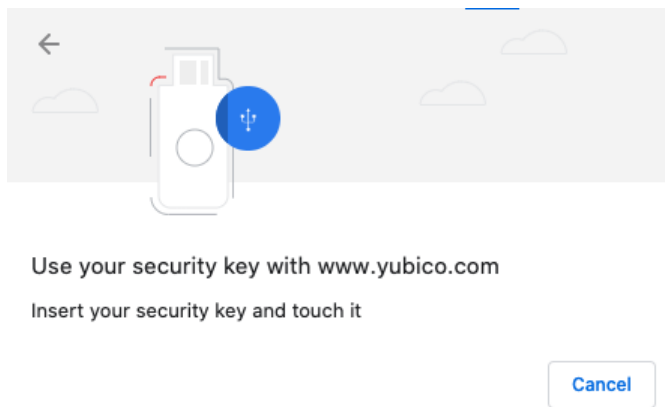
13.2.1 Enrolling the First Fingerprint

Step 1

Use an up-to-date Chrome browser to open the [YubiKey Bio Series setup](#) website. Insert your YubiKey Bio into your computer.

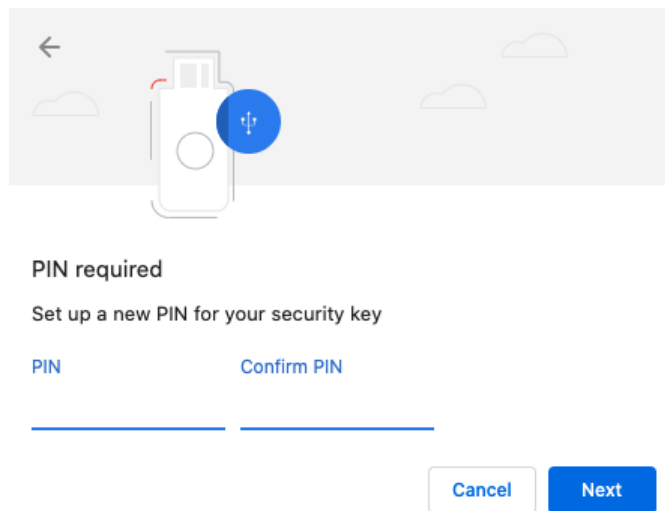
Step 2

Scroll down to the green button, **Enroll using Chrome**, and click it. The **Use your security key with Yubico.com** popup appears, taking you through the PIN setup (if no PIN is set) and later the fingerprint enrollment:



Step 3

If the amber LED flashes slowly, it means that either no fingerprint is enrolled or biometrics is blocked. If you have reason to believe biometrics is blocked, go to the appropriate link on the [YubiKey Bio Series setup](#) page or to [bio-tools-label](#). Otherwise, *touch the key*:



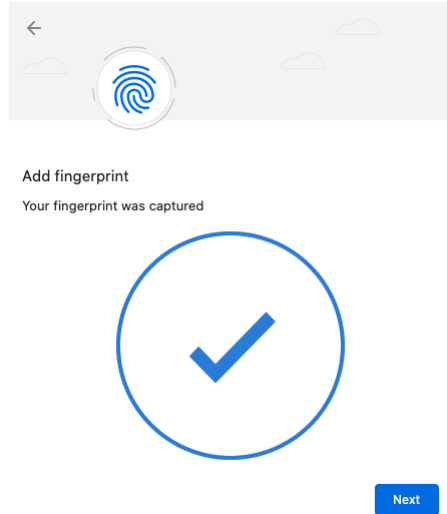
Step 4

If no PIN is set, set one by entering at least 4 digits, then confirming this PIN by re-entering it. If the YubiKey Bio already has a PIN set you are prompted to enter it.

Step 5

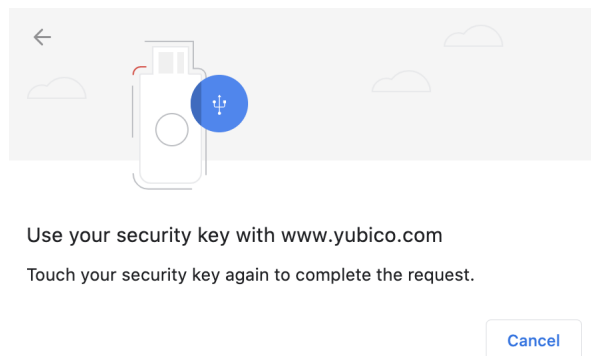
When prompted, touch the fingerprint sensor and the bezel. You are prompted to touch the sensor several times, as set out below. Change the angle of finger to sensor slightly each time.

Continue lifting and re-applying the same finger until the gray circle is entirely blue, the fingerprint icon is replaced by a tick mark, and the message in the popup reads “Your fingerprint was captured.”



Step 7

Click **Next**. The **Touch your security key again to complete the request** popup appears:



Step 8

Touch the bezel and sensor one last time. The final popup announces that enrollment was successful. The YubiKey Bio now has a template for that fingerprint.

13.2.2 Enrolling Additional Fingerprints

If the YubiKey Bio already has fingerprint(s) enrolled on it, repeating the procedure for the first fingerprint does not work for subsequent fingerprints. Instead follow these steps.

Note: You can also use this method for setting a PIN for a new YubiKey Bio and enrolling all fingerprints.

Step 1

Either paste `chrome://settings/securityKeys` into the Chrome address field or click on the

three vertical dots to the right of the URL field to navigate to **Settings->Security->Advanced->Manage security keys**.

Step 2

Click **Fingerprints** and follow the instructions in the popup.

13.3 Using Windows to Enroll Fingerprints

These are the instructions for setting a PIN on a YubiKey Bio and enrolling fingerprints on it using the Sign-in options on a Windows 10 or Windows 11 system.

Note: A YubiKey is a FIDO2 *hardware* authenticator. Both Windows and Mac have *built-in* FIDO2 authenticators - i.e., software authenticators that in this case are also platform authenticators. The prompts in both Windows and Mac *might assume* you will be using their own authenticators. Therefore it is quite easy to register *their* authenticators with a site or service by mistake, without realizing that you are not registering your YubiKey. Read the prompts carefully to avoid this. And remember that the PIN is associated with the authenticator, not the site or service.

Note: To get to the popup (prompt) for the YubiKey, you might need to *cancel* out of the pop-up for the built-in authenticator.

Although there are two FIDO *applications* on the YubiKey Bio, namely FIDO2 and U2F, it is the FIDO2 PIN that is required as fallback for both. The PIN is not associated with any *site*. When the fingerprint does not work and the key falls back to the PIN, it is the *key* that needs the PIN for authentication to all sites, including U2F sites (even though U2F has no concept of PIN). With fallback to PIN, it is easy if the user is authenticating to a WebAuthn/FIDO2 site, because the browser/client app *can* prompt for the PIN. Otherwise the user must unblock biometrics by using either:

- The [YubiKey Bio start page](#)
- Yubico Authenticator for Desktop.

For information on the YubiKey Bio's sensor and tips on working with fingerprints see [Tips](#). For detailed information on FIDO2 PINs and their requirements, see [Understanding YubiKey PINs](#).

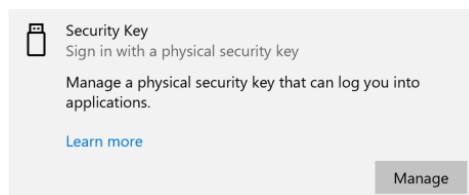
Step 1

On *Windows 10*, click **Enroll using Windows** on the YubiKey Bio setup page <<https://www.yubico.com/setup/yubikey-bio-series/>>`_`.

On *Windows 11*, click **Enroll using Windows** on the YubiKey Bio setup page <<https://www.yubico.com/setup/yubikey-bio-series/>>`_`. Then go to Step 3 below.

Step 2

On *Windows 10*, in the expanded **Security Key** field, click **Manage**.



Step 3

On both *Windows 10* and *Windows 11*, follow the Windows setup directions. Insert the YubiKey Bio into your computer's USB port and set a PIN for your YubiKey Bio if the key does not already have a PIN. In the **Security Key PIN** field, click **Add**. Enter a security key PIN and click **OK**.

Step 4

To enroll your fingerprint, in the **Security Key Fingerprint** field, click **Set up** and follow the prompts.

Touch the YubiKey Bio sensor while the green LED is still flashing, making sure to touch the ring-bezel as well.

Vary the way you touch each time to include more of the fingerprint. If the fingerprint you enroll is smaller than the sensor, apply some pressure to help ensure a good image capture.

Continue lifting and re-applying the same finger until you see the **All set!** message.

Perform this step up to five times for a total number of 5 enrolled fingerprints.

13.4 Tips

13.4.1 LED Behavior

The YubiKey Bio is not in a permanent state of readiness. It is therefore essential to wait for the key to signal its readiness by flashing the green LED before you touch it.

- If the key reacts to your touch by the flashing or blinking of the green LED, you used the right touch.
- If the amber LED flashes three times in quick succession, the attempt to match your fingerprint with the template was not successful.
- If the amber LED flashes slowly and continuously, it is in the biometrics blocked state.
- If the key does not react to your touch, you might not have touched both the bezel and the sensor. When you apply your fingerprint, always make sure you are touching the bezel at the same time. See *Tips for the Touch* below.

13.4.2 Fingerprint Enrollment Progress Indicators

The progress of reading of your fingerprint is displayed on-screen. The way it is shown depends on the client platform and browser. It is generally not under the control of the site or the service. The screenshots below show enrollment using platform support:

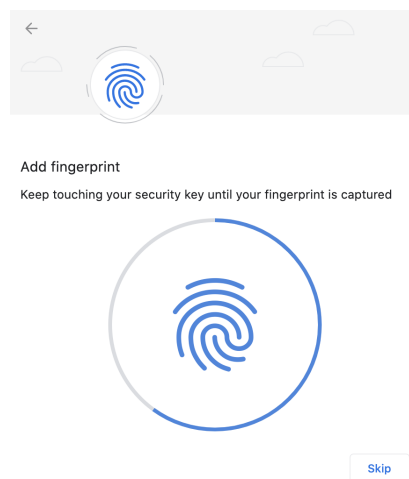


Fig. 1: Chrome on macOS, Linux, and Chrome OS: Capturing the Fingerprint

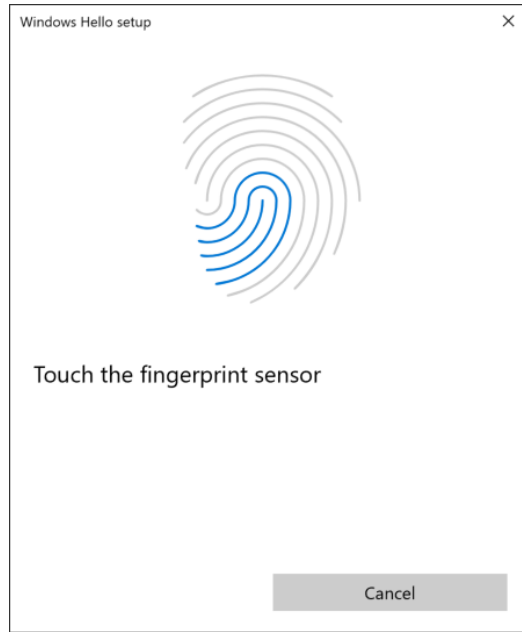


Fig. 2: Windows: Capturing the Fingerprint

13.4.3 Tips for the Touch

Because the fingerprint can be negatively affected by environmental conditions such as heat, cold, injury, etc., it is not always easy for the YubiKey Bio to interact with it. The following tips are helpful.

The YubiKey Bio recognizes **two interactions**, one a **touch**, and the other a **fingerprint**. Its recognition of the fingerprint - or lack thereof - is communicated through the LEDs (see *LED Behavior*).

On the YubiKey Bio, the silver-colored bezel encircling the fingerprint sensor provides the grounding plane required to read the fingerprint.

Biometric Touch

When prompted to have the YubiKey Bio read your fingerprint from the fingerprint sensor, be sure to touch at least a tiny part of the ring. If you use your little finger to touch only the center of the fingerprint sensor, the key will not read the fingerprint.

Plain Touch

When prompted to touch the YubiKey Bio but not explicitly asked for the fingerprint, touch **both** the bezel and the fingerprint sensor, even though the fingerprint will not be read.

Fingerprint

For enrolling, when we say 'fingertip', we actually mean the pad on the tip of the finger where the whorls of the fingerprint are. The fingerprint could equally well be a thumbprint or a toeprint; the YubiKey Bio makes no distinction between fingers, thumbs, and toes.

Quality of print

Dry or scarred skin can impede the key's ability to perform a successful fingerprint match. If your hands are dry, use moisturizer or water to enable conduction. Do not apply wet fingertips.

Repeated readings

Enrolling your fingerprint requires pressing your fingertip against sensor (and bezel) several times, usually 5 to 8 times. If an attempt to capture is unsuccessful the YubiKey Bio will need you to repeat it.

Vary the angle

When enrolling a new fingerprint, angle your finger so that different parts of the fingerprint come in contact with the sensor and bezel with each capture. This enables the YubiKey Bio sensor to collect a larger area of your finger.

Temperature

If the fingertip is too cold, the YubiKey Bio might not be able to read the fingerprint. If your hands are cold, rub them together to get the circulation going and warm them up.

Press firmly

Press the YubiKey Bio sensor and bezel with your fingertip gently but firmly and hold for a second or so. If you are using an adapter, it may be necessary to hold onto the adapter to prevent it from bending and interrupting the connection to the YubiKey.

Stabilize key

If the YubiKey Bio seems to wobble in the USB port, use your other hand to hold it steady in the port while you are applying your fingertip.

Stabilize dongle

If you are using a dongle as an adapter to your device's USB port, ensure the YubiKey Bio is stable enough for you to apply sufficient pressure with your fingertip.

Check the LEDs

When you start fingerprint enrollment, the green LED on your YubiKey Bio starts to flash. Start the fingerprint enrollment before the green LED on the YubiKey Bio stops flashing. The amber LED might flash slowly, indicating that no fingerprint is enrolled or that biometrics is in the blocked state.

Clean the sensor

If there is dust or oil residue on the YubiKey Bio sensor and bezel, clean it. See *Care and Cleaning*.

Change ports

Sometimes the USB port does not work well or the YubiKey Bio is loose in the port. Insert the YubiKey Bio in a different port on your device.

13.5 Troubleshooting and Tools

13.5.1 Troubleshooting

The primary source for troubleshooting tips is the FAQ on the [YubiKey Bio Series setup page](#).

Fingerprint

If the YubiKey cannot match fingerprint to template three times in a row, fingerprint recognition is blocked. The YubiKey Bio falls back to PIN.

PIN

If you enter the wrong PIN eight times in a row, the YubiKey FIDO2 application will be **locked**, which means it cannot communicate with you or with any site or service. It indicates the blocked state by flashing its amber LED slowly and continuously. In order to restore this functionality, the FIDO2 application must be reset. For more details, see *FIDO2 PIN*.

Unblock

Unblock the YubiKey Bio's biometric function (its ability to read fingerprints) by going to the unblocking FAQ on the [YubiKey Bio start page](#). Otherwise you can use any of the other methods given in tools-label.

Reset

You can also **reset** it, but doing so erases all the discoverable credentials on it, setting it back to

factory defaults. See *Resetting Your YubiKey Bio with the Yubico Authenticator for Desktop*.

If you run into any issues with a YubiKey Bio, you can also refer to the [Knowledge Base on Yubico's Support site](#) and search for your issue. If your issue is not listed in the Knowledge Base, or if you have any technical questions, you can open a ticket with our [Technical Support team](#).

Unblocking/Unlocking

Use the appropriate link on the [YubiKey Bio Series setup page](#) or the [Yubico Authenticator for Desktop](#).

Other Issues

If you run into any issues with a key from the YubiKey Bio Series, refer to the [Knowledge Base](#) and search for your issue. If your issue is not listed in the Knowledge Base, or if you have any technical questions, you can get in touch with Yubico Support, <http://yubi.co/support>.

13.5.2 Tools

Yubico Authenticator for Desktop

[Yubico Authenticator for Desktop](#) can be used to manage the YubiKey Bio. It is open source and cross-platform, running on Windows, macOS, and Linux. The iOS and Android versions of Yubico Authenticator cannot be used to manage the YubiKey Bio.

13.6 Requirements: Platform and Browser Compatibility

13.6.1 Desktop

The YubiKey Bio Series works with the latest versions of most browsers and desktop operating systems. Currently, the best experience can be had on macOS, Chrome OS, and Linux, running up-to-date Chromium-based browsers.

On **Windows 10**, browsers are not currently able to tell the user when the YubiKey has failed to match the fingerprint, so the user must watch out for the YubiKey's amber LED blinking to indicate when an attempt has failed. **Windows 11** does not have this problem.

On other platforms, browsers such as Firefox and Safari have not yet (at the time of writing) implemented CTAP 2.1 and therefore the user will typically be prompted to enter the PIN even if the key is not in the "biometrics blocked" state.

13.6.2 Mobile

- The YubiKey Bio does not have NFC capabilities.
- The YubiKey Bio can be used with mobile, but it is reliant on mobile operating system support as well as on browser support for the FIDO protocols. For more information, please refer to the relevant manufacturer's web sites for your mobile device.
- When the YubiKey Bio has fallen back to requiring the PIN, users may need to resort to computers (as opposed to mobile devices) to unblock biometrics .

13.7 Resetting Your YubiKey Bio with the Yubico Authenticator for Desktop

In this context, resetting means resetting the FIDO application. You can also perform a FIDO reset using the YubiKey Manager, Windows Sign-in options, or the Chrome browser settings.

The main cause for the biometric function blocking is failure to match the fingerprint three times in a row. If the YubiKey Bio was locked because the biometric function was blocked, you can just unblock it instead of resetting it: see [tools-label](#).

Resetting the key is not the same as unblocking it. Because resetting the FIDO2 and FIDO U2F applications returns the key to the factory default state, when it has neither fingerprints nor PIN nor credentials, you must enroll your fingerprints again after resetting it (see the relevant Enrolling chapter, either [Using Chrome to Enroll Fingerprints](#) or [Using Windows to Enroll Fingerprints](#)), and register your key again to your apps and services.

Note that resetting your YubiKey Bio deletes all credentials, the PIN, and stored fingerprint templates.

To review your options for tools to reset the YubiKey Bio, see [tools-label](#).

13.8 Frequently Asked Questions

The FAQs are on the [YubiKey Bio Start Page](#).

13.9 YubiKey Bio and FIDO2

The YubiKey Bio Series - FIDO Edition supports all FIDO2 scenarios supported by the YubiKey 5 Series and the Security Key Series. It can be used in both passwordless and second factor authentication scenarios. In both scenarios the fingerprint is used *in lieu of* the PIN, similar to the way biometrics is used on a smartphone. However, there are some scenarios in which the PIN is required. The PIN is required when enrolling or otherwise managing fingerprints, just as it is on a smartphone. However, the only opportunity to input the PIN is after 3 unsuccessful attempts at matching a fingerprint with an enrolled finger.

13.9.1 Discoverable Credentials

Like FIDO U2F, the FIDO2 standard offers the same high level of security, as it is based on public key cryptography. In addition to providing phishing-resistant two-factor authentication, the FIDO2 application on the YubiKey allows for the storage of discoverable credentials. (Fingerprint templates are not discoverable credentials.) Keys in the YubiKey Bio Series can hold up to 25 discoverable credentials. To manage them, see [Credential Management](#).

FIDO2 PIN

The FIDO2 PIN is necessary for:

- Enrolling fingerprints
- Managing enrolled fingerprints
- Fallback after failure to match fingerprint with template.

The FIDO2 PIN must be between 4 and 128 characters in length (for more information, see <https://support.yubico.com/hc/en-us/articles/4402836718866-Understanding-YubiKey-PINs>)

- There is no PIN set by default
- Once a FIDO2 PIN is set, it can be changed but it cannot be removed other than by resetting the FIDO2 application.
- If the FIDO2 PIN is entered incorrectly 3 times in a row, the key will need to be reinserted before it will accept additional PIN entry attempts (reinserting “reboots” the key).
- To see the number of retries remaining, use YubiKey Manager and navigate to Applications > FIDO2.
- If the PIN is entered incorrectly a total of 8 times in a row (3+3+2), the FIDO2 application will be locked, and FIDO2 authentication will not be possible.
- To restore the FIDO2 functionality, the FIDO2 application must be reset.

Note: Resetting the FIDO2 application will also reset the U2F application. No site you have registered the YubiKey with using U2F will work until the YubiKey is re-registered with that site.

FIDO2 Credentials

The discoverable credentials can be used for passwordless authentication, or they can be used for two-factor authentication. In both scenarios the credentials can be protected by the FIDO2 PIN and in the case of a YubiKey Bio, biometrics can be used in lieu of the PIN provided that fingerprints have been enrolled and that the key is not in biometrics blocked state.

13.9.2 User Verification

The YubiKey Bio implements always-on user verification, or `alwaysUV`.

The user verification requirement asks for proof that the user logging in is the same user as the one who set the PIN, enrolled fingerprints, and registered the key with the app or service (Relying Party, or RP). For more information about user verification, see [User Presence vs User Verification](#).

When `userVerification` is discouraged, the user experience is not optimal unless the platform has implemented CTAP 2.1. See [Multifactor Authentication \(MFA\)](#).

13.9.3 Credential Management

If you decide to discontinue using a site or service, you can delete its discoverable credential. This frees up space on the YubiKey Bio, which can contain up to 25 such credentials.

To view the discoverable credentials on your YubiKey and delete them selectively, use the Yubico Authenticator for Desktop version 5.1.0 and above.

For more information on credentials in general, and in particular on managing them, see [Enhancements to FIDO 2 Support](#) for details.

For more **developer-oriented** information on this, see [Discoverable Credentials / Resident Keys](#) on Yubico’s developer site.

13.9.4 Supported Extensions

The YubiKey Bio supports only the AppID extension (appid) as defined by the [W3C Web Authentication API specification](#). This extension allows U2F credentials registered using the legacy FIDO JavaScript APIs to be used with WebAuthn. In practice, that means that if you register a YubiKey Bio on a website when it used U2F and that website later upgrades to FIDO2, previously registered U2F credentials will continue to work.

Note: Developers: For AAGUID values, see [YubiKey Hardware FIDO2 AAGUIDs](#).

13.10 YubiKey Bio and FIDO U2F

The FIDO U2F protocol does not require any special drivers or configuration to use, just a compatible web browser. The U2F application on the YubiKey can be associated with an unlimited number of WebAuthn sites supporting FIDO U2F authentication.

FIDO U2F on the YubiKey Bio Series requires that the touch be a successful biometric match with an already enrolled fingerprint. This is different from FIDO U2F on other YubiKeys.

13.10.1 PIN + U2F

As the concept of PIN does not exist in FIDO U2F, after three successive failures to match the fingerprint, the key goes into the “biometrics blocked” state without first prompting for the PIN. An amber LED blinks slowly and continuously to indicate this state. Biometrics can be unblocked with a FIDO2 operation using the PIN (e.g., authentication). See [bio-tools-label](#) for full instructions and more information.

Note: Developers: With regard to computer login tools that use FIDO U2F for second-factor authentication, some software may use a YubiKey and FIDO U2F as a second factor. Since FIDO U2F has no concept of fallback to PIN, the YubiKey Bio is not likely to be a good choice for this use case. For more information about software that falls into this category, visit Yubico’s Support site and look for articles about the YubiKey Bio: <https://support.yubico.com/hc/en-us/search?query=YubiKey+Bio>

13.10.2 FIDO U2F Succeeded by FIDO2

FIDO2 is the umbrella term used to describe an amalgamation of two separate sets of specifications: WebAuthn and the Client-to-Authenticator Protocol, CTAP (currently version 2.1, and often referred to as CTAP2.1). The WebAuthn component provides a narrow scope of flexibility for developers on the service layer because it encompasses the logical interactions across a network. CTAP2.1, however, provides a much more open set of standards for the interaction between a security device and the user.

CTAP2.1 is also where biometrics such as fingerprint enrollment, management, and use were first defined. To create a cohesive user experience, adherence to this specification is required from:

- Authenticators such as the YubiKey Bio
- Clients such as the Chrome or Edge browsers
- Platforms such as Windows and macOS.

See *User Experiences*.

13.10.3 Supported Extensions

The YubiKey Bio supports only the AppID extension (appid) as defined by the [W3C Web Authentication API specification](#). This extension allows U2F credentials registered using the legacy FIDO JavaScript APIs to be used with WebAuthn. In practice, that means that if you register a YubiKey Bio on a website when it used U2F and that website later upgrades to FIDO2, previously registered U2F credentials will continue to work.

Note: Developers: For AAGUID values, see [YubiKey Hardware FIDO2 AAGUIDs](#).

To get in touch with Yubico Support, [click here](#).

ACRONYMS

3DES

Triple Data Encryption Algorithm

AES

Advanced Encryption Standard

CCC

Card Capability Container

CCID

Chip card interface device, a USB protocol for a smartcard.

CHUID

Card Holder Unique ID

CMS

Credential Management System

CN

Common name

CSR

Certificate Signing Request

ECC

Elliptic curve cryptography

FIDO

Fast Identity Online

FIPS

Federal Information Processing Standards (US government) covering codes and encryption standards.

HMAC

Hash-based message authentication code

HOTP

HMAC-based One-Time Password algorithm

KDF

Key Derivation Function

OATH

The Initiative for Open Authentication is an organization that specifies two open authentication standards, TOTP and HOTP.

OTP

One-Time Password

PBKDF2

Password-Based Key Derivation Function 2

PKCS #11

This is number eleven of the Public Key Cryptography Standards; it is also the API for creating and manipulating cryptographic tokens.

PUK

PIN Unlock Key

stdin

standard input - usually keyboard or CLI instructions

stdout

standard output - usually print to screen

TOTP

Time-based One-Time Password algorithm

X.509

The standard defining the format of a [public key certificate](#)

COPYRIGHT

© 2021-2023 Yubico AB. All rights reserved.

15.1 Trademarks

Yubico and YubiKey are registered trademarks of Yubico AB. All other trademarks are the property of their respective owners.

15.1.1 Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design, and manufacturing. Yubico shall have no liability for any error or damages of any kind resulting from the use of this document.

The Yubico Software referenced in this document is licensed to you under the terms and conditions accompanying the software or as otherwise agreed between you or the company that you are representing.

15.1.2 Contact Information

Yubico Inc.
5201 Great America Parkway
#122
Santa Clara, CA 95054
USA

<https://www.yubico.com/support/contact/>

More options for getting touch with us are available on the Contact page of Yubico's website.

15.1.3 Document Updated

2023-02-24 20:35:24 UTC