YubiKey Technical Manual

Yubico

Aug 09, 2024

CONTENTS

CHAPTER

ONE

INTRODUCTION TO THE DIFFERENT YUBIKEY SERIES

Where applicable, throughout this guide, the YubiKey 5 Series, the YubiKey Bio Series, the YubiKey 5 FIPS Series and the YubiKey 5 CSPN Series are referred to collectively as the **YubiKey 5 (FIPS/CSPN) Series**. This label indicates that a certain specification or feature is available on all these Series. This is possible because they all share the same base hardware and many firmware features.

This topic introduces:

- *YubiKey 5 Series*
- *YubiKey 5 FIPS Series*
- *YubiKey Bio Series*
- *YubiKey 5 CSPN Series*
- *Firmware*
- *Security Key Series*

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 is *Understanding the USB Interfaces*. This topic 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, with the *Protocols and Applications* chapter listing these by protocol:

- *FIDO2*
- *Smart Card (PIV Compatible)*
- *OATH*

- *OpenPGP*
- *OTP*
- *YubiHSM Auth*.

However, it is the firmware version that determines which of the more specialized functionalities and capabilities are available on your YubiKey. See *Firmware*.

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 U.S. 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 (the non-FIPS series) offers the same security, algorithms, and functionality. The standard line also evolves at a much more rapid pace because it does not need to complete an exhaustive validation process, which commonly takes a year or more. Yubico can release standard firmware with new features and enhancements at any time, whereas FIPS-certified products must go through the FIPS validation process every time there is a firmware 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 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 security feature that supports multiple protocols designed to be embedded in USB security tokens. The module can generate, store, and perform cryptographic operations for sensitive data. It is accessed through 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 on the YubiKey.

Table 1: YubiKey 5 FIPS Series Cryptographic Module Major Functions

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 detailed in *Deploying the YubiKey 5 FIPS Series*.

1.2.4 FIPS-specific Aspects of the YubiKey 5 FIPS Series

The table below lists the YubiKey 5 FIPS Series with the 5.4 firmware configuration changes that are set at programming. These are in addition to the configuration options available in the YubiKey 5 FIPS Series.

Table 2: YubiKey 5 FIPS Series 5.4 Firmware Configuration Changes

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 **AAGUID Values** in *FIDO2*)
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 (see *Footnote 1*).
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 (see *Footnote 2*).

Footnote 1 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.

Footnote 2 Form factor is set during manufacturing and returned as a one-byte value. Currently defined values for this are set out in the *Form Factor* table below:

Table 3: 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, USB-A	0x01	0x41	0x81
Nano, USB-A	0x02	N/A	0x82
Keychain, USB-C	0x03	0x43	0x83
Nano, USB-C	0x04	N/A	0x84
Keychain with Lightning, USB-C	0x05	N/A	x85

1.3 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 four keys:

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

1.3.1 Protocols Supported

All keys in the YubiKey Bio Series support the FIDO authentication protocols, and all work with sites and applications that support the FIDO2 and FIDO U2F protocols (for more information, see *YubiKey Bio and FIDO2* and *YubiKey Bio and FIDO U2F*).

FIDO2 (sometimes referred to as WebAuthn) builds upon FIDO U2F, and is the standard that enables the replacement of password-based authentication.

The YubiKey Bio Series provides firmware applications to support two modes of authentication through the FIDO2 and U2F protocols (see *YubiKey Bio and FIDO2* and *YubiKey Bio and FIDO U2F*). Even though the firmware applications are separate from one another, they both share the same PIN and FIDO reset capability, which is to say that a FIDO `reset` resets both applications. To manage these applications, see *Tools*.

The YubiKey Bio Multi-protocol Edition also supports the PIV protocol, and works with sites and applications that support PIV/SmartCard interfaces. In order to use the biometric feature on the YubiKey Bio Multi-protocol Edition, the YubiKey Smart Card Minidriver (Windows) is required (the Minidriver download is the third item on this page).

1.3.2 Using the YubiKey Bio

For a quick start to using the YubiKey Bio Series, without a lot of details, see Yubico's setup page.

This guide, the *YubiKey Technical Manual*, provides:

- An explanation of the way the YubiKey Bio works (see *How the YubiKey Bio Works*) and descriptions of the different user experiences with the various protocols (see *User Experiences*).
- 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
  - *YubiKey Bio and FIDO U2F*
- A brief explanation of the role the Yubico Authenticator plays in managing the YubiKey Bio.

1.3.3 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 also work with the YubiKey Bio. However, the UI on client devices is not as intuitive and there might be some limitations.

1.3.4 Interfaces and Applications

Interfaces

Like all 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 *YubiKey Bio and FIDO U2F*. FIDO2 (also known as WebAuthn) is 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 *Troubleshooting and Tools*.

1.4 YubiKey 5 CSPN Series

Instructions on how to configure and use the YubiKey 5 in compliance with CSPN ("Certificat de Sécurité de Premier Niveau" [RD1]) are given in *YubiKey 5 CSPN Series Specifics*.

For each YubiKey application that requires specific configuration, there is a short introduction, the required settings to achieve the target, and a technical description of the configuration.

1.4.1 References

Code	Document title	Reference (Link)
[RD1]	Certification de sécurité de premier niveau des technologies de l'information	https://cyber.gouv.fr/produits-certifies/yubikey-5-series-version-firmware-542
[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_0879V5.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://cyber.gouv.fr/produits-certifies/yubikey-5-series-version-firmware-542

1.5 Firmware

For a summary overview of the firmware features, see *Firmware Overview*, which includes the capability matrices, listing the features and form factors available per firmware version for each of the products in the YubiKey 5 Series and the Security Key Series.

For more in-depth information about:

- the most recent firmware release, see *5.7 Firmware Specifics*.
- the firmware prior to the current release, see *Firmware Specifics Prior to 5.6.x*

Note: Yubico periodically updates the firmware to take advantage of features and capabilities introduced into the ecosystem. YubiKeys are programmed in Yubico's facilities with the latest available firmware and once programmed cannot be updated to another version. The firmware cannot be altered or removed from a YubiKey.

The firmware version on a YubiKey or a Security Key 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 either the Yubico Authenticator or the YubiKey Manager tool (ykman), a lightweight software package installable on many OS. The YubiKey Manager has both a graphical user interface (GUI) and a command line interface (CLI). For more information, see the YubiKey Manager (ykman) CLI & GUI Guide.

1.5.1 NIST: FIPS

Yubico submitted the firmware for releases 5.4.2 and 5.4.3 to NIST and the organization approved the certification. The certificates can be found here. For more information about the YubiKey 5 FIPS Series, see *YubiKey 5 FIPS Series Specifics*.

1.5.2 ANSI: CSPN

Yubico submitted release 5.4.2 to ANSSI for certification and the organization approved the certification. For more information about the YubiKey 5 CSPN Series, see *YubiKey 5 CSPN Series*.

1.6 Security Key Series

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

The *Security Key Series - Enterprise Edition* is the same as the *Security Key Series* but includes a serial number to enable asset tracking. The serial number is on the back of the key and can also be read programmatically through the FIDO HID interface. It is only available in USB-A + NFC and USB-C + NFC form factors.

Table 4: Form Factor

Capability	Security Key Series 5.0.x-5.4.x	Security Key Series - Enterprise Edition 5.4.x	Security Key Series 5.7.x	Security Key Series - Enterprise Edition 5.7.x
Serial Number	No	Yes	No	Yes
Serial Number over CCID	No	No	No	Yes
PIN Complexity	No	No	No	Yes

1.6.1 Serial Number over CCID

The serial number of the Yubico Security Key is retrievable without Windows elevated privileges since the YubiKey Management Application is exposed over CCID. This change was introduced in 5.7.0.

1.6.2 Video Tutorial

Get started with Security Key Series

Click for Yubico Support.

CHAPTER

TWO

FIRMWARE OVERVIEW

2.1 YubiKey 5 Series

The new 5.7. firmware for the YubiKey 5 Series has a number of new and improved features that will be available for the first time on the multi-protocol YubiKey 5. The changes and additions are described in detail in *5.7 Firmware Specifics*.In addition to the features that are directly accessible, there are a number of features that require partner support.

Note: Yubico periodically updates the firmware to take advantage of features and capabilities introduced into the ecosystem. YubiKeys are programmed in Yubico's facilities with the latest available firmware and once programmed cannot be updated to another version. The firmware cannot be altered or removed from a YubiKey.

The firmware version on a YubiKey or a Security Key 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 either the Yubico Authenticator or the YubiKey Manager tool (ykman), a lightweight software package installable on many OS. The YubiKey Manager has both a graphical user interface (GUI) and a command line interface (CLI). For more information, see the YubiKey Manager (ykman) CLI & GUI Guide.

The features, capabilities, and enhancements of the YubiKey 5 Series that are dependent on firmware version are listed below in the *Firmware Capability Matrix* below.

2.2 Security Key Series

The Security Key Series - including Enterprise Edition - will be updated with the latest firmware, including the updates from FIDO listed above. The Enterprise Edition will have the following additional updates:

- Minimum PIN length set to 6
- PIN Complexity turned on by default (and cannot be turned off)
- Serial number retrievable by client software in Windows without requiring elevated privileges (admin rights) since the YubiKey management application is accessible via CCID, which enables use cases where client software needs to read the serial number of the authenticator.

2.3 Firmware Capability Matrix

2.3.1 YubiKey 5 Series

Table 1: **Features and Form Factors Available per Firmware Version**

Feature/Form Factor	Firmware Versions					
	5.0.x	5.1.x	5.2.x	5.3.x	5.4.x	5.7.x
Serial Number	Yes	Yes	Yes	Yes	Yes	Yes
OTP	Yes	Yes	Yes	Yes	Yes	Yes
OATH	Yes	Yes	Yes	Yes	Yes	Yes
OpenPGP version	2.1	2.1	3.4	3.4	3.4	3.4
PIV/Smart Card	Yes	Yes	Yes	Yes	Yes	Yes
FIDO U2F	Yes	Yes	Yes	Yes	Yes	Yes
FIDO2/WebAu	Yes	Yes	Yes	Yes	Yes	Yes
YubiHSM Auth					Yes	Yes
SCP03				Yes	Yes	Yes
SCP11						Yes
FIDO2 Credential Storage	25	25	25	25	25	100
OATH Credential Storage	32	32	32	32	32	64
USB-A	Yes	Yes	Yes	Yes	Yes	Yes
USB-A + NFC	Yes	Yes	Yes	Yes	Yes	Yes
USB-C	Yes	Yes	Yes	Yes	Yes	Yes
USB-C + NFC		Yes	Yes	Yes	Yes	Yes
USB-A Nano	Yes	Yes	Yes	Yes	Yes	Yes
USB-C Nano	Yes	Yes	Yes	Yes	Yes	Yes
Lightning + USB-C			Yes	Yes	Yes	Yes

2.3.2 YubiKey 5 FIPS Series

Table 2: **Features and Form Factors Available per Firmware Version**

Feature/Form Factor	Firmware Versions	
	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
SCP11		
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

2.3.3 YubiKey 5 CSPN Series

Table 3: **Features and Form Factors Available per Firmware Version**

Feature/Form Factor	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

2.3.4 YubiKey Bio Series

Table 4: **Features and Form Factors Available per Firmware Version**

Feature/Form Factor	Firmware Versions 5.5.x	 5.6.x	 5.7.x
Serial Number	Yes	Yes	Yes
OTP			
OATH			
OpenPGP version			
PIV/Smart Card			Yes
FIDO U2F	Yes	Yes	Yes
FIDO2/WebAuthn	Yes	Yes	Yes
YubiHSM Auth			
SCP03	Yes	Yes	Yes
SCP11			Yes
USB-A	Yes	Yes	Yes
USB-A + NFC			
USB-C	Yes	Yes	Yes
USB-C + NFC			
USB-A Nano			
USB-C Nano			
Lightning + USB-C			

PIV Support

Smart Card/PIV is only available on the YubiKey Bio Multi-protocol Edition.

2.3.5 Security Key Series

Table 5: **Features and Form Factors Available per Firmware Version**

Feature/Form Factor	Firmware Versions				
	5.0.x - 5.2.x	5.4.x	5.4.x Enterprise Ed.	5.7.x	5.7.x Enterprise Ed.
Serial Number			Yes		Yes
OTP					
OATH					
OpenPGP version					
PIV/Smart Card					
FIDO U2F	Yes	Yes	Yes	Yes	Yes
FIDO2/WebAuthr	Yes	Yes	Yes	Yes	Yes
YubiHSM Auth					
SCP03					
FIDO2 PIN Mgmt*				Yes	Yes
Enterprise Attestation					Yes
Blob Storage				Yes	Yes
Always UV				Yes	Yes
USB-A	Yes				
USB-A + NFC	Yes	Yes	Yes	Yes	Yes
USB-C					
USB-C + NFC		Yes	Yes	Yes	Yes
USB-A Nano					
USB-C Nano					
Lightning + USB-C					

Click for Yubico Support.

CHAPTER

THREE

5.7 FIRMWARE SPECIFICS

This section provides detailed descriptions of the features enabled by the 5.7 firmware (includes 5.6.x).

3.1 CTAP 2.1 Feature Summary

CTAP 2.1 is the evolution - and first update - of CTAP 2.0, which was introduced for FIDO2/WebAuthn several years ago. The new features enabled by CTAP 2.1 are primarily enterprise-focused, but also support new FIDO2 use cases. Yubico supported CTAP 2.1 features even before that standard was approved, for example, credential management introduced in 5.2.1 (see *Managing Credentials*) and `uvRetries` introduced in 5.5.0.

In firmware 5.7 the PIV and OpenPGP applications both support unicode PINs, as well as counting each unicode code point as a single character.

These CTAP 2.1 features are also available in the Security Key series for FIDO-only deployments (see *Security Key Series*).

CTAP 2.1 support gives organizations:

- Improved control of the CTAP 2.1 authenticators they have deployed
- Ability to list compliance requirements such as:
  - Permissible authenticators

  - PIN requirements
- More granular control of the end user experience
- Additional capabilities for CMS vendors through the storage of additional data, etc.:
  - To configure authenticators
  - To manage the authenticator lifecycle
- Management beyond the scope of FIDO2, through the ability to associate the FIDO2 credentials on the authenticator with other credentials on other protocols, such as certificates on the PIV module.

3.1.1 FIDO2 Extensions

Table 1: **FIDO2 Extensions Available per Firmware Version**

New Capabilities	YubiKey 5 Series	Security Key Series	Security Key Series Enterprise Edition
PIN Management Flexibility			
Temporary FIDO2 PIN can be set (forces user to change upon next use)	Yes	Yes	Yes
Configure FIDO2 minimum PIN length	Yes	Yes	Yes
Enhanced Asset Tracking			
Capable of Enterprise Attestation (requires custom configuration)	Yes		Yes
Serial number retrievable by client software in Windows without Admin rights	Yes Also applies to earlier firmware versions		Yes New in 5.7
Enhanced Smart Card Capabilities (PIV)			
RSA-3072 and RSA-4096 support	Yes		
Ed25519 and X25519 key types	Yes		
Future-proofed for Compliance			
Enhanced PIN complexity	Yes Requires custom config		Yes Enabled by default
Default minimum PIN length of 6			Yes
Enhanced Credential Storage			
Support for 100 passkeys	Yes	Yes	Yes
Support for 24 PIV certificates	Yes		
Support for 64 OATH credentials	Yes		
Support for 2 OTP seeds	Yes		

3.2 Enterprise Attestation

Enterprise Attestation (EA) enables Identity Providers (IdPs) to read the serial number (or other unique identifier specific to the organization) on custom-programmed keys during FIDO2 registration. This satisfies a variety of asset tracking requirements, and can aid in account recovery by enabling an end user to prove they have a specific FIDO2 device. In addition to support from the Relying Party (RP) and/or IdP, EA requires platform support (see *Current Platform/RP Support*).

EA's ability to identify individual authenticators as opposed to just the type of authenticator changes the privacy model of the FIDO protocol, making the FIDO credential behave more like a certificate. Typical use-cases are:

- Tracking of individual authenticators on registration to ensure only authenticators issued by the organization are used. This resolves a common compliance requirement that previously could only be solved by policy or by custom AAGUIDs.
- If the organization knows what serial number a user was issued but does not see it registered or did not register it on the user's behalf, the organization can take appropriate steps to help the end user register their authenticator. This will help organizations roll out phishing-resistant MFA.
- Tying the FIDO credential to a PIV certificate by matching serial numbers (or other device-specific information) between the FIDO2 EA certificate and the PIV Attestation certificate.
- Identify individual authenticators in troubleshooting scenarios. When a key is lost or broken, a user can be guided by an IT admin who knows what authenticator holds which credential. The admin can advise which key is being used and which should be de-activated. The serial number of a back-up authenticator can be identified, too.

Developers seeking more information can refer to Enterprise Attestation on developers.yubico.com.

Current Platform/RP Support

At present, few RPs support EA; however there is platform support for it in Chrome and some Chromium based browsers. Windows 11 is required on Windows platforms.

CMS vendor support for EA is currently a little sparse; however, that is rapidly changing.

3.3 Minimum PIN Length and Minimum PIN Length Extension

`minPINLength` enables the minimum PIN length to be set on an authenticator. The `minPINLength` needs to be set locally by a client tool such as YubiKey Manager/ykman. Once set, it cannot be shortened without resetting the authenticator.

`minPINLengthExtension` enables IdPs to support FIDO registration self-enrollment processes by enforcing the configured minimum PIN length. The Relying Party (RP), if configured via an allowed list configured on the YubiKey, is enabled to query the `minPINLength` of the authenticator.

This extension resolves compliance requirements for organizations that need to enforce use of specific PIN lengths. Before 5.7, this was only possible through the deploymnent of YubiKey 5 FIPS Series (authenticators certified for FIPS 140-2 has a minimum PIN length of 6) or custom configuration, in which there were no checks the RP could perform unless the authenticator had a custom AAGUID.

Current Platform/RP Support

No current RP supports this feature, however there is platform support for it in Chrome and some Chromium based browsers. Windows 11 is required on Windows platforms.

3.4 Force PIN Change

Force PIN change or `forcePINChange` (FIDO 2.1 specification) enables vendors or IT admins to prompt end-users to change their FIDO2 PIN. This is valuable in a pre-registration/enroll on behalf of scenario where the organization does not want to know their end users' PINs. End-users are prompted to set their own PIN (may be combined with minPINLength), i.e. a PIN not known by the organization.

This feature also minimizes the number of helpdesk calls due to forgotten PINs because end-users can set PINs that are meaningful to them.

Current Platform Support Force PIN change only requires support on the platform, and can only be set by communicating with the YubiKey directly. Chrome and some Chromium-based browsers support it. **Windows 11 is required on Windows platforms**.

> There is no need for explicit RP support, the force PIN change (`forcePINChange`) flag is set on the client/platform side, which is where it will trigger the flow.

3.5 Always User Validation

`alwaysuv` was introduced to prompt users for user verification (UV) each time, which provides consistency in behavior between different platforms and RPs. End-users are often confused because the setting `uv=preferred/discouraged` behaves differently depending on whether the user is on a macOS or a Windows machine.

This feature is enabled by default in the YubiKey Bio: it always asks for biometrics and never "plain touch" in a second factor flow when UV is not necessarily required. Otherwise an end user might touch the fingerprint sensor with an unenrolled finger and successfully authenticate when only performing User Presence (UP), therefore thinking they "bypassed the biometrics" or that the biometric sensor was faulty, allowing an unenrolled finger to authenticate.

An organization might want to enable it so that users always enter their PIN, ensuring they are less likely to forget it.

Current Platform/RP Support

> This setting is internal to the authenticator and requires no specific platform or RP support.

3.6 Blob Storage

There are two blob storage options available on the YubiKey 5.7. Both Credential Blobs and Large Blobs require support on the platform as well as from the RP.

3.6.1 Credential Blob

A `credBlob` is 32 bytes of unencrypted storage per credential that can be set during registration and retrieved during authentication for discoverable credentials. This feature allows for a small amount of secret data to be associated with a discoverable credential during `makeCredential`. The blob is opaque to the authenticator. This enables IdPs to include a small amount of information such as a certificate thumbprint to aid in authentication scenarios. PII can be stored in this field if it is used with `credProtect`.

There is a great variety of use cases, as the `credBlob` enables storage of arbitrary data. For example, it can:

- Be used as HPKP-like public key hash to identify, e.g., kerberos certificates to trust when using a given credential ("on prem AD").
- Provide information about the issuance of the specific credential.

3.6.2 Large Blob

authenticatorLargeBlob storage is 4096 bytes of compressed, shared storage on the authenticator. It is managed by the platform, and is always encrypted with the Large Blob Key - a per-credential symmetric encryption key that is used by the platform to read the contents of the large blob. Large blobs can be used for storing authentication certificates or other artifacts linked to the private FIDO2 key stored on an authenticator.

The large blob feature allows for a “large” amount of data to be added to a discoverable credential upon creation. The typical use case of this is a public SSH key.

Creating an SSH key using a discoverable FIDO2 credential enables the authenticator to be hardware-bound and perform SSH authentications using a key stored in the FIDO2 applet.

With the addition of large blob the user can take the authenticator to a new machine without needing to copy the public part to the new client machine.

Any other data can be associated with the key, such as linkage to a PIV certificate or details on the creation of the credential.

Note that a largeBlobKey is necessary to decrypt the data in the Large Blob.

3.7 FIDO Level 2

YubiKey 5.7 will undergo FIDO Level 2 certification for assurance of attestable hardware-bound credentials. This enables YubiKeys for use with e-government use cases (citizen-facing) and corporate compliance mandates that require FIDO L2 certification.

3.8 PIV Enhancements

3.8.1 Additional Key Types Supported

In accordance with the August 2023 Department of Defense memo on stronger public key algorithms, the 5.7.x firmware supports RSA-3072 and RSA-4096.

In addition, the 5.7.x firmware also supports the Ed25519 and X25519 key types.

3.8.2 PIV Management Key (AES)

Given that after December 31, 2023, three-key TDEA is disallowed for encryption unless specifically allowed by other NIST guidance (decryption using three-key TDEA is allowed for legacy use) the default management key with the 5.7.x firmware uses AES-192 instead of TDES. The management key uses the same default value as previous keys (TDES and AES-192 keys are the same length. If you need to know what these values actually are, go to the “General Information” section in the “Yubico PIV Tool” guide on our developers’ site).

Prior to the release of the 5.4.2 firmware, the PIV management key was a **TDES** key. This feature expanded 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** in *Smart Card (PIV Compatible)*.

3.8.3 Advanced Key Management Functions

With the 5.7.x firmware, the PIV application supports advanced key management functions such as moving and deleting keys:

- The ability to move keys enables retaining retired encryption keys on the device to decrypt older messages.
- The ability to delete keys enables destroying key material without overwriting with bogus data or resetting the PIV application.

Generate a New Key Pair

The four new algorithms (alg) that can be used for key generation are:

- RSA-3072 (0x05)
- RSA-4096 (0x16)
- Ed25519 (0xE0)
- X25519 (0xE1)

For more information, see Generate asymmetric key pair.

Import a Key

The four new algorithms (alg) that can be used for key import are:

- RSA-3072 (0x05)
- RSA-4096 (0x16)
- Ed25519 (0xE0)
- X25519 (0xE1)

For more information, see Import asymmetric key pair.

Below is the updated list of tags for the import data. Values followed by an asterisk (*) are new for firmware 5.7.

Table 2: **List of Tags for Import Data**

Algorithms	Key Element	Tag
RSA-1024 (0x06) RSA-2048 (0x07) RSA-3072 (0x05)* RSA-4096 (0x16)*	prime P	0x01
	prime Q	0x02
	prime p exponent dP	0x03
	prime q exponent dQ	0x04
	CRT coefficient QInv	0x05
ECC-P-256 (0x11) ECC-P-384 (0x14)	private value s	0x06
Ed25519 (0xE0)*	seed	0x07*
X25519 (0xE1)*	seed	0x08*

Move a Key

Keys can be moved from any slot except F9 (attestation) to any other slot except F9 using the instruction 0xF6.

Table 3: **Moving a Key**

CLA	00
INS	F6
P1	Destination slot
	9A, 9C, 9D, 9E,
	82, 93, 84, 85, 86, 87, 88, 89, 8A, 8B, 8C, 8D, 8E, 8F,
	90, 91, 92, 93, 94, 95
P2	Source slot
	9A, 9C, 9D, 9E,
	82, 93, 84, 85, 86, 87, 88, 89, 8A, 8B, 8C, 8D, 8E, 8F,
	90, 91, 92, 93, 94, 95
	P2

Delete a Key

Any key can be deleted from any slot, including F9 (Attestation) using the instruction 0xF6 with a value of 0xFF for P1.

Table 4: **Deleting a Key**

CLA	00
INS	F6
P1	FF
P2	Source slot
	9A, 9C, 9D, 9E,
	82, 93, 84, 85, 86, 87, 88, 89, 8A, 8B, 8C, 8D, 8E, 8F,
	90, 91, 92, 93, 94, 95
	F9

3.8.4 YubiKey PIV Metadata

YubiKey 5 PIV metadata enables services and client software to obtain information about PIV keys from a central location, which means it is no longer necessary to query PIV attestation. The YubiKey PIV application can therefore report on characteristics of cryptographic keys in the specified PIV slot. Integration with CMS vendors is thus facilitated by YubiKey PIV metadata.

PIV metadata is available with the 5.3.0 firmware. For details, see the Get Metadata section of the PIV extensions.

3.9 PIN Complexity

PIN Complexity is available on YubiKeys with firmware version 5.7.0 and later.

Note: The PIN complexity feature is available only with custom configuration, specified **before** you order keys.

This feature prevents users from adopting simple patterns or common PINs, and thereby significantly reduces the risk of users setting easily guessable PINS on their devices.

When PIN complexity is enabled:

- It applies to all the applications on the YubiKey that process PINs.
- The PINs for the different applications are all still separate and distinct, but they will all follow the same set of rules.
- It is applied to PINs for the following protocols on the YubiKey:
  - FIDO2
    - PIN
  - PIV
    - PIN
    - PUK
  - OpenPGP
    - user PIN
    - admin PIN
    - reset code
  - yubihsm-auth
    - credential PINs
    - YubiKey access codes

Unicode Characters

In firmware 5.7, the support for Unicode PINs has been extended to include the PIV and OpenPGP applications. Each unicode code point is counted as a single character.

PIN Blocking

The PINs that will be blocked are those that:

- Are less than 6 characters
- Contain only one unicode character, e.g. `111111`
- Are on the following blocklist:
  - 123456
  - 123123
  - 654321
  - 123321
  - 112233
  - 121212

        - 123456789
        - password
        - qwerty
        - 12345678
        - 1234567
        - 520520
        - 123654
        - 1234567890
        - 159753
        - qwerty123
        - abc123
        - password1
        - iloveyou
        - 1q2w3e4r

3.10 Expanded Storage (FIDO2 and OATH)

The FIDO2 and OATH applications both have increased storage capacity. FIDO2 has been increased to 100 discoverable credentials (aka Passkeys), and OATH storage has been increased to 64 seeds. As before, all storage limits are per-application, so users can store data up to the maximum for each application simultaneously for a potential total of 190 credentials:

- Up to 100 passkeys
- 24 PIV certificates (limited by overall memory used)
- 64 OATH seeds
- 2 OTP seeds

3.11 Restricted NFC

Restricted NFC mode prevents wireless device manipulation before a YubiKey NFC with the 5.7 firmware is taken out of its blister pack or other packaging such as a tray. To ensure that these keys cannot be tampered with during shipping, this mode is enabled by default on new NFC keys with the 5.7 firmware.

When these keys are taken out of their packaging, the only permitted action via the NFC connection is reading the URL configured by Yubico on the NDEF tag set by Yubico. Because both major mobile OSs read NDEF tags and open URLs by default, users immediately learn how to disable Restricted NFC mode. The NDEF tag is set to https://www.yubico.com/getting-started/.

When tapped against a mobile device, a YubiKey 5.7 NFC will cause the browser to open to the configured URL with the instructions for enabling full NFC operation. The end user is instructed to plug the key into USB power such as a USB charger or computer USB port for 3 seconds. This action is sufficient to disable Restricted NFC mode. The user can re-enable the restriction as often as they desire using the YubiKey Manager/ykman.

3.12 Yubico Crypto Library

Over the past few years Yubico has developed a library in-house that performs the underlying cryptographic operations (encryption, signing, etc.) for RSA and ECC.

Click for Yubico Support.

CHAPTER

FOUR

FIRMWARE SPECIFICS PRIOR TO 5.6.X

This section gives summary descriptions of features that came out with firmware versions prior to the current 5.7.x release.

4.1 Secure Channel (Firmware 5.3.0 and later)

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. This protocol for secure channel is implemented on YubiKeys as of Yubico 5.3.0 firmware.

For **detailed descriptions** of the secure channel feature refer to *Yubico Secure Channel Technical Description*, *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.

4.1.1 Security Domains & Key Diversification

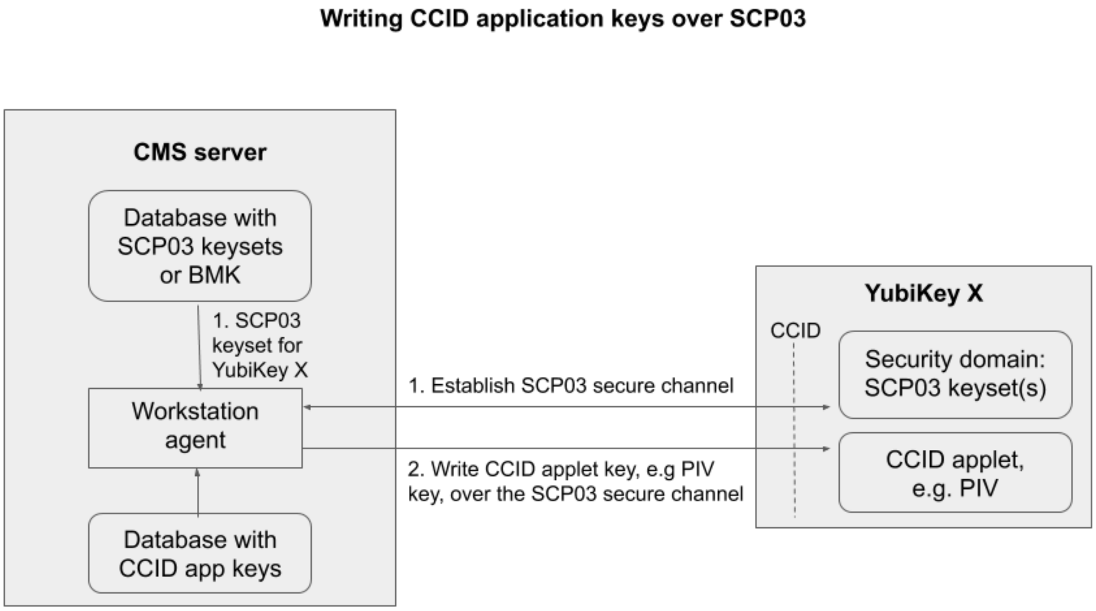
The authenticated and encrypted communication channel takes place over the CCID interface between a host and the secure element on the YubiKey. This includes configuration of, or communication from, CCID applications. The secure channel feature can therefore be used to load application keys onto the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

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 the device 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, establish a secure channel with the security domain. Do this with a previously loaded transport key set or the default transport key set. Each secure channel transport key set is protected by being


Writing CCID application keys over SCP03
CMS server
Database with SCP03 keysets or BMK
1. SCP03 keyset for YubiKey X
Workstation agent
Database with CCID app keys
1. Establish SCP03 secure channel
2. Write CCID applet key, e.g PIV key, over the SCP03 secure channel
YubiKey X
CCID
Security domain: SCP03 keyset(s)
CCID applet, e.g. PIV

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 on the host and the YubiKey, a secure session is established.

Establish SCP03 Secure Channel

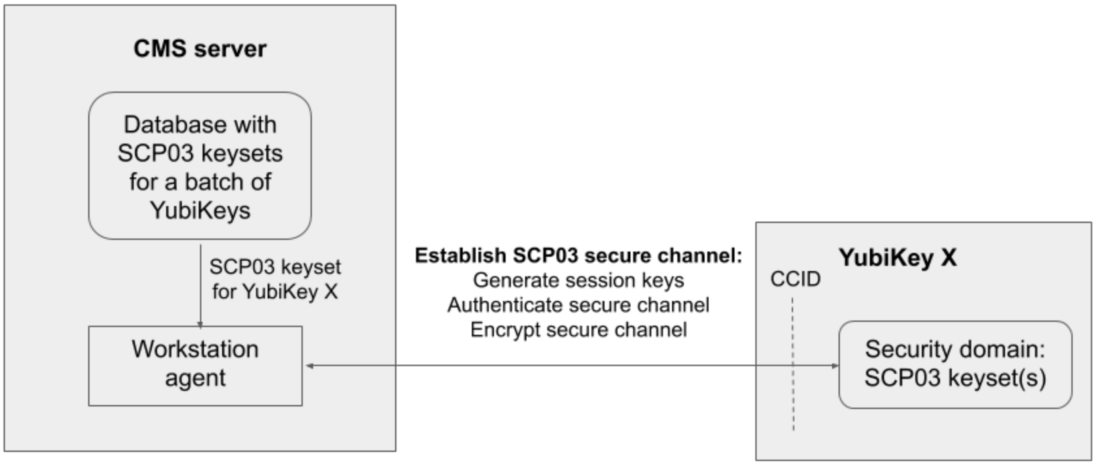
4.1.2 Secure Channel Benefits and Usage

- 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.

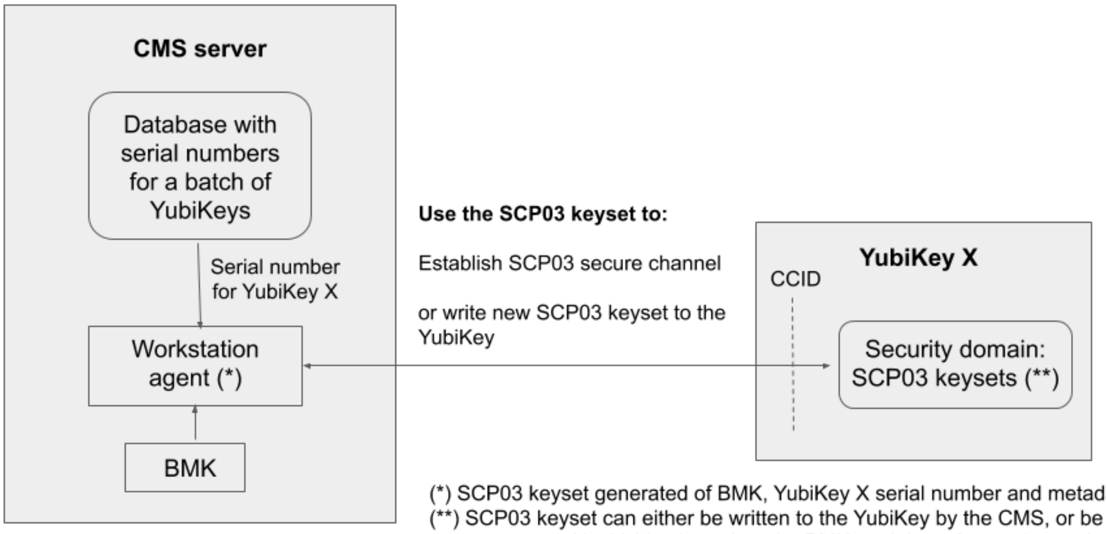
SCP03 Key Diversification

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

Establish SCP03 secure channel


CMS server
Database with SCP03 keysets for a batch of YubiKeys
SCP03 keyset for YubiKey X
Workstation agent
Establish SCP03 secure channel:
Generate session keys
Authenticate secure channel
Encrypt secure channel
YubiKey X
CCID
Security domain: SCP03 keyset(s)

SCP03 key diversification


CMS server
Database with serial numbers for a batch of YubiKeys
Serial number for YubiKey X
Workstation agent (*)
BMK
Use the SCP03 keyset to:
Establish SCP03 secure channel
or write new SCP03 keyset to the YubiKey
YubiKey X
CCID
Security domain: SCP03 keysets (**)

(*) SCP03 keyset generated of BMK, YubiKey X serial number and metadata
(**) SCP03 keyset can either be written to the YubiKey by the CMS, or be preprogrammed by Yubico based on the BMK, serial number and metadata

4.1.3 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*.

4.2 NFC ID: Calculation Changed (5.3.0)

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 5.3.0 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*.

4.3 YubiHSM Auth (5.4.3)

4.4 YubiHSM Auth

4.4.1 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), see *Yubico Secure Channel Technical Description*. YubiHSM Auth is supported by YubiKey firmware version 5.4.3 and above.

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.

4.4.2 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).

4.4.3 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.

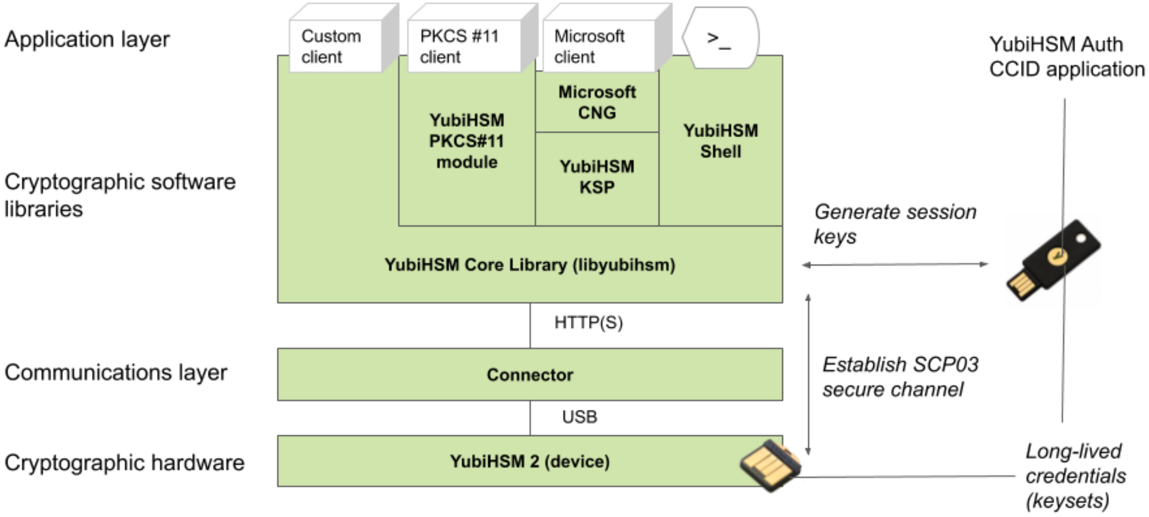
Those long-lived key sets are used by the YubiHSM Auth application to derive a set of three session-specific AES-128 keys using the challenge-response protocol as defined in SCP03:

- 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.

4.4.4 Architecture Overview

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



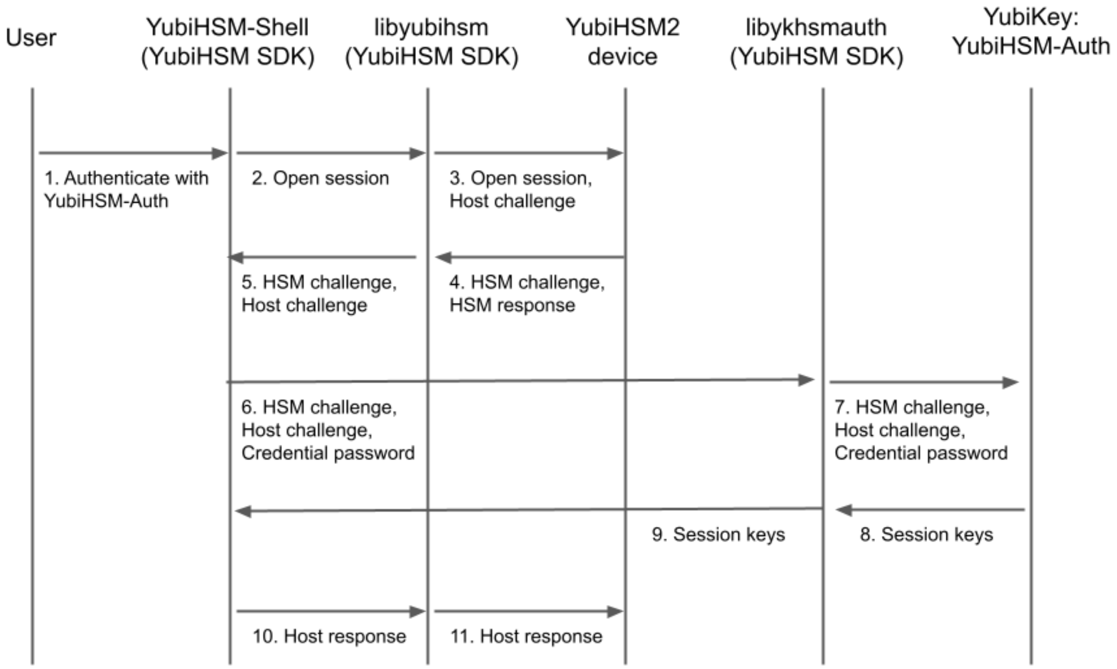
The identical long-lived credentials (key sets) are protected in both the YubiKey YubiHSM Auth application and in the YubiHSM 2. The YubiHSM-Shell software tool can be used for generating the key sets in the YubiHSM 2, and the YubiHSM-Auth software tool can be used for importing the same key sets to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

4.4.5 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.


User
YubiHSM-Shell (YubiHSM SDK)
libyubihsm (YubiHSM SDK)
YubiHSM2 device
libykhsmauth (YubiHSM SDK)
YubiKey: YubiHSM-Auth
1. Authenticate with YubiHSM-Auth
2. Open session
3. Open session, Host challenge
5. HSM challenge, Host challenge
4. HSM challenge, HSM response
6. HSM challenge, Host challenge, Credential password
7. HSM challenge, Host challenge, Credential password
9. Session keys
8. Session keys
10. Host response
11. Host response

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.

4.4.6 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.

To generate and store the equivalent YubiHSM Auth credentials on the YubiKey, use the `yubihsm-auth` command line tool. 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`. This requires the user 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 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 the YubiHSM User Guide, section YubiHSM Auth.

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

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.

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 (Firmware 5.3.0 and later)* 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.

4.4.7 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.

Click for Yubico Support.

CHAPTER

FIVE

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

In addition, all of the keys in the YubiKey 5 FIPS Series have the acronym “FIPS” underneath the Data Matrix code on the back, along with “v5” running up the left side of the Data Matrix code, except on the YubiKey 5C Nano, which has it on the right.

All current YubiKeys have been IP68-rated under the IEC standard 60529.

5.1 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: 3.7g
- Physical Interfaces: USB, NFC
- Operating Temperatures: 0 °C - 40 °C (32 °F - 104 °F)
- Storage Temperatures: -20 °C - 85 °C (-4 °F - 185 °F)

For more information, see our Support article on the YubiKey 5 NFC.

5.2 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: 0.5g
- Physical Interfaces: USB
- Operating Temperatures: 0 °C - 40 °C (32 °F - 104 °F)
- Storage Temperatures: -20 °C - 85 °C (-4 °F - 185 °F)

For more information, see our Support article on the YubiKey 5 Nano.

5.3 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: 1.8g
- Physical Interfaces: USB
- Operating Temperatures: 0 °C - 40 °C (32 °F - 104 °F)
- Storage Temperatures: -20 °C - 85 °C (-4 °F - 185 °F)

For more information, see our Support article on the YubiKey 5C.

5.4 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: 0.7g
- Physical Interfaces: USB
- Operating Temperatures: 0 °C - 40 °C (32 °F - 104 °F)
- Storage Temperatures: -20 °C - 85 °C (-4 °F - 185 °F)

For more information, see our Support article on the YubiKey 5C Nano.

5.5 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: 2.8g
- Physical Interfaces: USB, Lightning®
- Operating Temperatures: 0 °C - 40 °C (32 °F - 104 °F)
- Storage Temperatures: -20 °C - 85 °C (-4 °F - 185 °F)

For more information, see our Support article on the YubiKey 5Ci.

5.6 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: 4.3g
- Physical Interfaces: USB, NFC
- Operating Temperatures: 0 °C - 40 °C (32 °F - 104 °F)
- Storage Temperatures: -20 °C - 85 °C (-4 °F - 185 °F)

For more information, see our Support article on the YubiKey 5C NFC.

5.7 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 have a maximum transfer rate of 12 Mbps.

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

YubiKey Bio - FIDO Edition

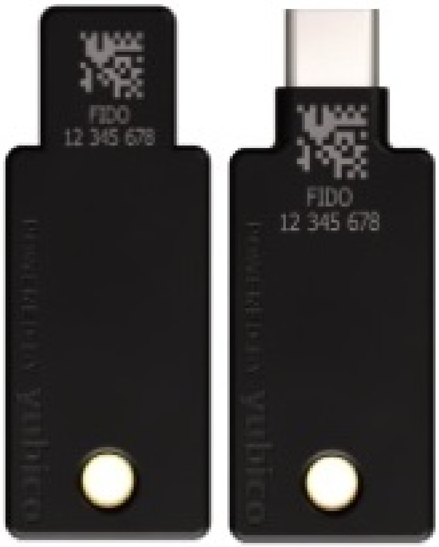
For more information, see our Support article on the YubiKey Bio - FIDO Edition.

YubiKey C Bio - FIDO Edition

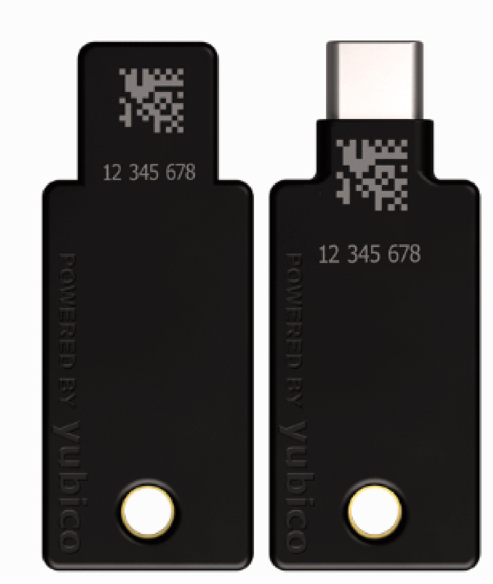
For more information, see our Support article on the YubiKey C Bio - FIDO Edition.

5.7.1 Laser Marking

The laser marking of the FIDO Edition of the YubiKey Bio Series was changed in March 2024. To distinguish the FIDO Edition from the Multi-protocol Edition, 'FIDO' is marked on the back near the serial number and data matrix. YubiKeys of product type YubiKey Bio Series - FIDO Edition produced before March 2024 do not have 'FIDO' marked on the back, instead they have only the serial number and data matrix on the back.



YubiKey Bio Series - FIDO Edition marked 'FIDO', March 2024 and later



YubiKey Bio Series - Multi-protocol Edition and YubiKey Bio Series - FIDO Edition prior to March 2024

5.7.2 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. Touch types:

Biometric

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 does not read the fingerprint.

Plain

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 is not read.

Fingerprint Tips provides detailed instructions on using the fingerprint sensor.

5.7.3 LEDs

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

5.7.4 Ratings

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

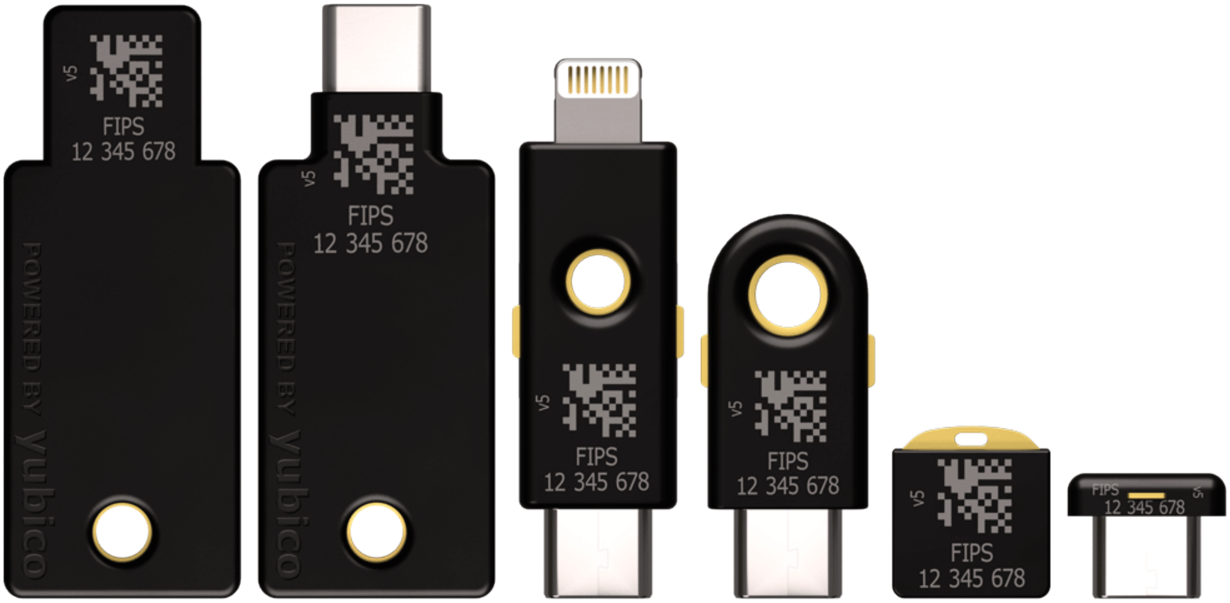
5.7.5 Care and Cleaning

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

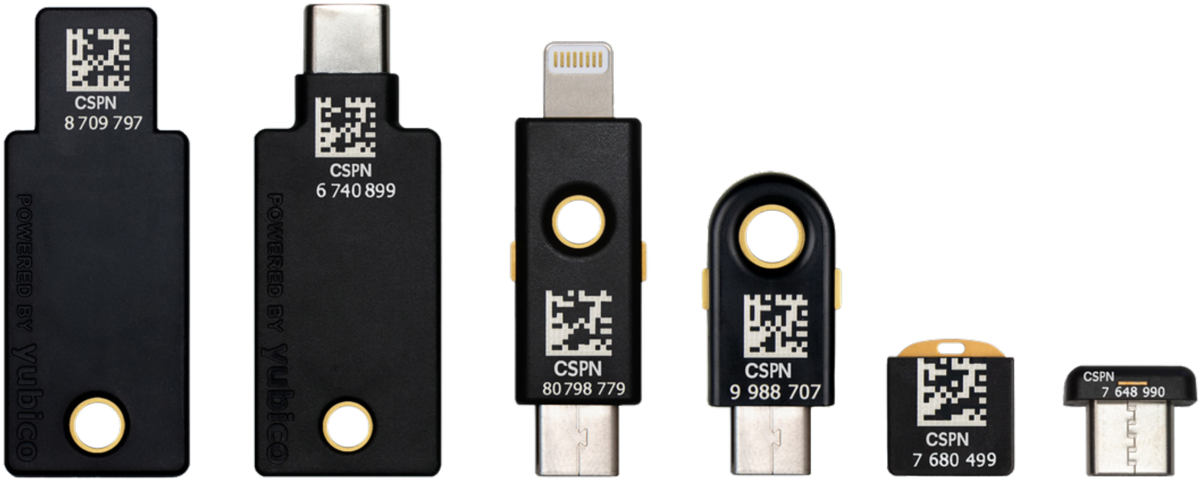
5.7.6 Operational Data

- Dimensions
  - YubiKey A Bio: 18mm x 45mm x 3.35mm
  - YubiKey C Bio: 18mm x 45mm x 3.75mm
- Weight: YubiKey Bio A: 4.3g; 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)

5.8 FIPS-specific Marking

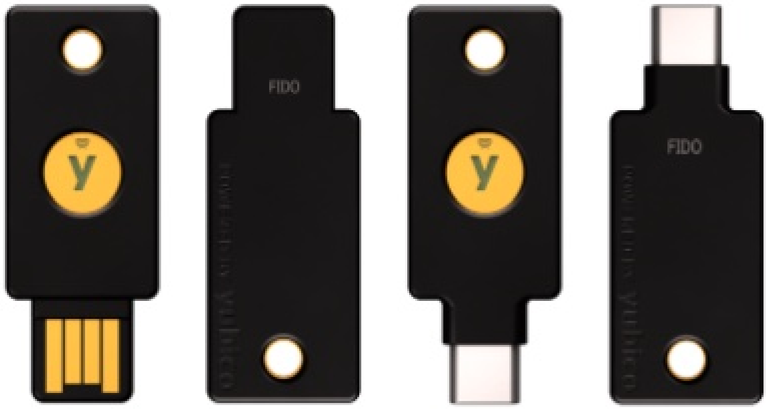


5.9 CSPN-specific Marking

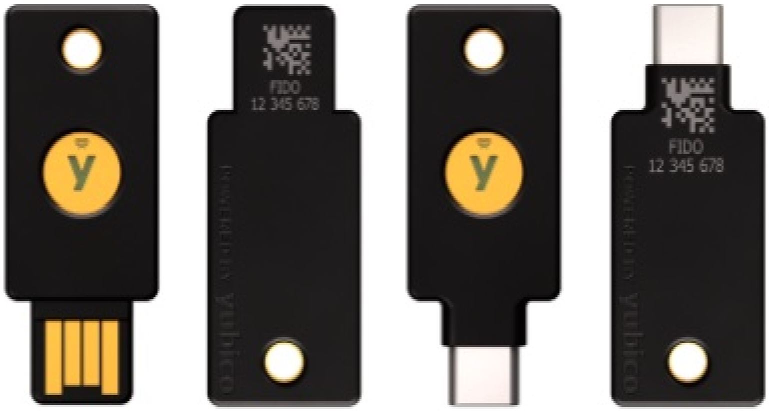


5.10 Security Key Series Marking

The color of the *non-Enterprise* edition of the Security Key Series was changed to black in January 2023. To distinguish the keys in the Security Key Series from those in the YubiKey 5 Series, **FIDO** is inscribed on the backs of both the Enterprise Edition and otherwise.



The serial number is inscribed on the back of each key in the *Enterprise* edition of the Security Key Series.



Click for Yubico Support.

CHAPTER

SIX

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.

6.1 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 presents 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*.

6.2 Apple Lightning®

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 through the touch-triggered OTP.
- Any WebAuthn-compliant application (starting in iOS 13). This includes the Safari browser.

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

Note: **Developers**: Enabling apps within iOS to use advanced protocols that send and receive information from the YubiKey 5Ci requires that you:

- Use the Yubico SDK. Access the Yubico iOS SDK at https://github.com/YubicoLabs/yubikit-ios
- Register the app with Yubico. Go to the Yubico iOS SDK App submission page.

The USB and iProduct strings that are displayed when connecting through Lightning® or USB are specific to the connection type. They are described in the YubiKey USB ID Values guide.

6.3 NFC

The NFC-capable hardware security keys from Yubico (YubiKey 5 NFC, YubiKey 5C NFC, YubiKey 5 NFC FIPS, YubiKey 5C NFC FIPS, Security Key NFC, Security Key C NFC) provide an NFC wireless interface in addition to USB. These models 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; follow the link below for an example of the new format. The `<OTP>` value is replaced with the OTP generated by the YubiKey.

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

On Apple iPhone devices, iPhone 7 and newer is required. Background tag reading is supported in the iPhone XS and newer.

For operations that require a touch, all touch requests within the first 20 seconds of the operation succeed. To help prevent unintended access, a YubiKey placed on a desktop NFC reader might power down unless the NFC reader has power-cycled, which prevents the YubiKey from powering down. To regain connectivity with an NFC reader, remove the YubiKey from the reader and reposition it on the reader.

Click for Yubico Support.

CHAPTER

SEVEN

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**.

7.1 OTP

The OTP interface presents itself to the operating system as a USB keyboard. The OTP application is accessible over this interface.

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.

7.2 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).

7.3 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: Access to the CCID interface on iOS, requires the Yubico iOS SDK.

Click for Yubico Support.

CHAPTER

EIGHT

PROTOCOLS AND APPLICATIONS

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

This topic covers:

- *FIDO2*
- *Smart Card (PIV Compatible)*
- *OATH*
- *OpenPGP*
- *OTP*
- *YubiHSM Auth*

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).
- We support a static password and Challenge-Response with Touch-triggered OTP. 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, see *OATH* and *OTP*.

8.1 FIDO2

For an overview of the FIDO2 features that became available with the 5.7.x firmware, see *5.7 Firmware Specifics*.

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.

8.1.1 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 6 and 63 alphanumeric characters.
- Once a FIDO2 PIN is set, it can be changed but it cannot be removed without resetting the FIDO2 application.
- After 3 incorrect PIN entries, power cycle the FIDO2 application. This allows the FIDO2 PIN to be attempted again.
- If the PIN is entered incorrectly 8 times in a row, the FIDO2 application locks and FIDO2 authentication is no longer possible. To use the FIDO2 application after the application has locked, a FIDO2 reset is required.

Note: Resetting the FIDO2 application also resets the U2F application. This means the YubiKey must be re-registered not only with all the FIDO2 sites, but also with all U2F sites.

Note: The YubiKey 5 supports FIDO2 credential management. This enables selectively deleting resident keys. See our article Enhancements to FIDO 2 Support for details.

8.1.2 FIDO2 PINs and Fingerprint Templates

PINs and Fingerprint templates, collectively referred to as “User Verification” or UV for short, are one of the enhancements from U2F included in FIDO2. FIDO2 UV enables single device Multi-Factor Authentication (MFA). It enables users to use a single device (the YubiKey) to provide two authentication factors: something you have - the YubiKey, and something you know (a PIN) or something you are (a biometric fingerprint template on the YubiKey Bio).

FIDO2 credentials on a YubiKey cannot be accessed without either the PIN or on the YubiKey Bio, the fingerprint. There are no backdoors to bypass the UV protections. This is the main reason that Yubico recommends configuring a minimum of two YubiKeys on each web site you use, to ensure you continue to have access to that site if you lose access to the first YubiKey. If the fingerprint sensor on the YubiKey Bio is damaged, the PIN is the only method available to use the credentials on the device.

Because the PIN or fingerprint is only used to authenticate with the YubiKey, and the protection against brute-force attacks (a maximum of 8 incorrect PIN attempts before the YubiKey FIDO2 application locks), there’s no security benefit in regularly changing the PIN unless there is reason to believe it has been compromised. Changing the FIDO2 PIN will not invalidate any credentials on the YubiKey. Note that previous values for the PIN are not stored within the YubiKey, and cannot be leveraged to regain access in the event the current PIN is forgotten.

8.1.3 Discoverable Credentials: Passkeys

Another new feature added in FIDO2 is discoverable credentials. Formerly referred to as resident keys, discoverable credentials are credentials which contain information about the site or service the credential belongs to, including the address and account username. These credentials can be more convenient for web sites that support them, because they allow secure login without requiring users to enter a username.

Not all web sites or service providers offer support for discoverable credentials, and there is no information stored on the YubiKey for any credential for a website or service provider that does not use a discoverable credential.

8.1.4 FIDO2 Connector Support

FIDO2 support is available to Apple devices 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

YubiKey 5Ci

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 through the touch-triggered OTP.
- Any WebAuthn-compliant application (starting in iOS 13). This includes the Safari browser.

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

Note: **Developers**: Enabling apps within iOS to use advanced protocols that send and receive information from the YubiKey 5Ci requires that you:

- Use the Yubico SDK. Access the Yubico iOS SDK at https://github.com/YubicoLabs/yubikit-ios
- Register the app with Yubico. Go to the Yubico iOS SDK App submission page.

The USB and iProduct strings that are displayed when connecting through Lightning® or USB are specific to the connection type. They are described in our Support article YubiKey USB ID Values.

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. All those aspects are covered by the second part of our Support article 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.

Note: 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.

8.1.5 FIDO2 PIN

The resident credentials can be protected by a PIN for an additional factor when authenticating, thereby making the YubiKey a multi-factor authenticator. This is achieved by performing UV at the time of authentication.

- The FIDO2 PIN must be between 4 and 63 characters, unless configured to another minimum PIN 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 locks. In order to restore this functionality, reset the FIDO2 application.

Note: Resetting the FIDO2 application also resets the U2F key. This means the YubiKey must be re-registered not only with all the FIDO2 sites, but also with all U2F sites.

Note: The YubiKey 5 Series and Security Key Series supports Credential Management to allow for selective deletion of resident keys. See Enhancements to FIDO 2 Support for details.

8.1.6 Default Values

PIN: None set.

8.1.7 AAGUID Values

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

New AAGUIDs are issued for new YubiKey products that support FIDO2, or when existing YubiKey products have FIDO2 features added or removed.

For the complete list of AAGUIDs, see our the article on our Support site, YubiKey Hardware FIDO2 AAGUIDs.

8.1.8 Supported Extensions

The YubiKey 5 Series supports 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 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 continue to work on that website.

Table 1: **FIDO2 Extensions Available per Firmware Version**

	Firmware Versions							
Extension and feature	5.0.x	5.1.x	5.2.x	5.3.x	5.4.x	5.5.x	5.6.x	5.7.x
ECC P256 Credentials	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
EdDSA/Ed25519 Credentials			Yes	Yes	Yes	Yes	Yes	Yes
ECC P384 Credentials							Yes	Yes
Credential Protection extension			Yes	Yes	Yes	Yes	Yes	Yes
HMAC-Secret extension			Yes	Yes	Yes	Yes	Yes	Yes
Credential Management			Yes	Yes	Yes	Yes	Yes	Yes
PIN Protocol v2					Yes	Yes	Yes	Yes
Credential Blob						Yes	Yes	Yes
Authenticator Large Blob						1kB	1kB	4kB
alwaysUV						Yes	Yes	Yes
Force PIN Change						Yes	Yes	Yes
Minimum PIN Length						Yes	Yes	Yes
Biometric Enrollment						Yes	Yes	Yes
Built-in UV (fingerprint)						Yes	Yes	Yes
Make Credential UV Not Required						Yes	Yes	Yes
Enterprise Attestation							Yes	Yes

Enterprise Attestation

Enterprise Attestation (EA) enables pre-defined Relying Parties (RPs) to read the YubiKey serial number on custom-programmed keys during FIDO2 registration. This satisfies a variety of asset tracking requirements, and can aid in account recovery by allowing an end user to prove they have a specific FIDO2 device. In addition to support from the RP and/or Identity Provider (IdP), EA requires platform support in the form of CTAP 2.1 capabilities.

EA's ability to identify individual authenticators as opposed to just the type of authenticator changes the privacy model of the FIDO protocol; however, the model is changed only by RPs defined by the customer at the time of order (utilizing EA vendor mode) or by supported platforms (utilizing EA platform mode).

Typical Use-Cases

- Tracking of individual authenticators on registration ensures that only authenticators issued by the organization are used. This resolves a common compliance requirement that previously could only be met by using policies or custom AAGUIDs.
- If the organization knows what serial number a user was issued but does not see it registered or did not register it on the user's behalf, the organization can take appropriate steps to help the end user register their authenticator. This will help organizations roll out phishing-resistant MFA.
- Tie the FIDO credential to a PIV certificate by matching serial numbers (or other device-specific information) between the FIDO EA and the PIV Attestation certificate.
- Identify individual authenticators in troubleshooting scenarios. When a key is lost or broken, a user can be guided by an IT admin who knows what authenticator holds what credential. The admin can advise which key is being used and which should be de-activated. The serial number of a back-up authenticator can be identified, too.

Developers seeking more information can refer to Enterprise Attestation on developers.yubico.com.

Current Platform/RP Support

At present, few RPs support this feature; however, there is platform support for it in Chrome and some Chromium-based browsers. Windows 11 is required on Windows platforms.

CMS vendor support for EA is currently a little sparse; however, that is rapidly changing.

Minimum PIN Length

The Minimum PIN Length Extension (`minPINLengthExtension`) enables RPs to enforce PIN length requirements, for example in regulated environments. The RPs pre-defined by the organization or end user are able to query the `minPINLength` of the authenticator. Once set, the PIN length cannot be shortened until the authenticator is reset. The minimum PIN length is configurable only by platforms, or by communicating with the YubiKey directly, and can only be read by IdPs or RPs via an allowed list configured on the YubiKey.

This extension resolves compliance requirements for organizations that require certain PIN lengths. Before 5.7, this was only possible through FIPS (YubiKey 5 FIPS Series with firmware version 5.4.3 has a minimum PIN length of 6), or custom configuration in which there were no checks the RP could perform unless the authenticator had a custom AAGUID.

Current Platform/RP Support

No current RP supports this feature, however there is platform support for it in Chrome and some Chromium-based browsers. Windows 11 is required on Windows platforms.

Force PIN Change

`Force PIN Change` enables vendors or IT admins to prompt end-users to change their FIDO2 PIN upon next use. This is valuable in a pre-registration/enroll-on-behalf of scenario where the organization does not want their end users' PINs to be known. End-users are prompted to set their own PINs (can be combined with `minPINLength`).

This feature also minimizes the number of help-desk calls due to forgotten PINs because end-users can set PINs that are meaningful to them.

Note: A PIN is not a password; it is local to the authenticator.

Current Platform Support

`Force PIN change` only requires support on the platform, and can only be set by communicating with the YubiKey directly. Chrome and some Chromium-based browsers support it on macOS and Linux while other browsers support it on Windows. Windows 11 is required on Windows platforms.

There is no need for explicit RP support, the `force PIN change` flag is set on the client/platform side, which is where it will trigger the flow.

Always UV (`alwaysUV`)

Always UV was introduced to prompt users for user verification (UV) with each use (authentication and registration), which provides consistency in behavior between different platforms and RPs. End-users are often confused because the setting `uv=preferred/discouraged` behaves differently depending on whether the user is on a macOS or a Windows machine.

This feature is enabled by default in the YubiKey Bio, which always asks for biometrics and never "plain touch" in a second factor flow when UV is not necessarily required. If this feature is not enabled, an end user might touch the fingerprint sensor with an unenrolled finger and authenticate, therefore mistakenly believing they bypassed the biometrics or that the biometric sensor was faulty, in that it apparently allowed an unenrolled finger to authenticate.

Another reason for enabling `alwaysUV` is to oblige users to enter their PINs, thus ensuring they are less likely to forget them.

Current Platform/RP Support

This setting is internal to the authenticator and requires no specific platform or RP support.

Blob Storage

There are two blob storage options available on the YubiKey 5.7 and later. Both Credential Blobs and Large Blobs require support on the platform as well as from the RP.

Cred blob (credBlob)

Credential Blobs are 32 bytes of unencrypted storage per credential that can be set during registration and retrieved during authentication for discoverable credentials. This feature allows for a small amount of data to be associated with a discoverable credential during `makeCredential`. The blob is opaque to the authenticator. This enables IdPs to include a small amount of information such as a certificate thumbprint to aid in authentication scenarios. PII can be stored in this field if it is used with `credProtect`.

There are many use cases, as the `credBlob` enables storage of arbitrary data; however, it can:

- Be used as HPKP-like public key hash to identify for example kerberos certificates to trust when using a given credential ("on prem AD").
- Provide information about the issuance of the specific credential.

Large blob (authenticatorLargeBlob)

Large blob storage is 4096 bytes on firmware 5.7.0 and later, (1024 bytes on firmware 5.5.x and 5.6.x) of compressed, shared storage on the authenticator. It is managed by the platform, and is always encrypted with the Large Blob Key - a per-credential symmetric encryption key that is used by the platform to read the contents of the large blob. Large blobs can be used for storing authentication certificates or other artifacts linked to the private FIDO2 key stored on an authenticator.

The large blob feature allows for a "large" amount of data to be added to a discoverable credential upon creation. The typical use case is a public SSH key.

Creating an SSH key using a discoverable FIDO2 credential would allow the authenticator (a) to perform SSH authentications using a key stored in the FIDO2 applet and (b) be hardware-bound. With the addition of large blob the user can take the authenticator to a new machine and would not need to copy the public part to the new client machine.

Any other data can be associated with the key, such as linkage to a PIV certificate or details on the creation of the credential.

The `largeBlobKey` is required to decrypt the data in the Large Blob.

8.1.9 Calculating the RPID

The FIDO2 protocol uses the RPID (Relying Party ID) as an identifier for the RP (Relying Party) that an authenticator authenticates against. To calculate the PRID of an URL, apply the SHA-256 algorithms over the RP's URL, e.g. `webauthntest.identitystandards.io` has the RPID of `0b997ccceb3aeb29c55c94a894b11cf01a24b4c8ae706f328cc2ea8cebc4ad5c`.

Developers seeking more information can refer to The W3C's WebAuthn specification.

8.1.10 Tools for Managing the FIDO2 Application

Each operating system has different software available to manage the YubiKey, with different capabilities. Note that the same YubiKey can be configured using any or all of the tools listed.

Yubico Authenticator for Windows, MacOS and Linux

Platforms
Windows, MacOS, Linux

Capabilities
Set or change the PIN, manage fingerprint templates, manage discoverable credentials, reset the YubiKey

Works with
All YubiKeys that support FIDO2

Yubico Authenticator enables users to manage all aspects of the FIDO2 application, and can do all the things that are outlined in this document for managing the YubiKey, including adding fingerprint templates to the YubiKey Bio. Yubico Authenticator requires administrative privileges for several operations on Windows, so for non-administrative users on Windows, the built-in Windows Security Key tools may be a better option.

See *Managing Discoverable Credentials with Yubico Authenticator*. Download Yubico Authenticator for Windows, MacOS or Linux.

YubiKey Manager (ykman) for Windows, MacOS and Linux

Platforms
Windows, MacOS, Linux

Capabilities
Set or change the PIN, reset the YubiKey, manage discoverable credentials (ykman CLI only)

Works with
All YubiKeys that support FIDO2.

The YubiKey Manager GUI is a general purpose utility that is able to configure many of the applications on the YubiKey in addition to FIDO2. YubiKey Manager requires administrative privileges to configure FIDO2, or detect FIDO2-only devices like the Security Key series or the YubiKey Bio on Windows, so for non-administrative users, the built in Windows Security Key tools may be a better option.

The command line version of the YubiKey Manager available for advanced users is referred to as ykman.

Download YubiKey Manager for Windows, MacOS, Linux.

Chrome on MacOS, Linux and ChromeOS

Platforms
MacOS, Linux, ChromeOS

Capabilities
Set or change the FIDO2 PIN, manage fingerprint templates, manage discoverable credentials, reset the YubiKey

Works with
All YubiKeys that support FIDO2.

Chrome has built-in support for managing FIDO2 devices, and will allow for managing security keys on the non-mobile based platforms where it is available. These capabilities are not available on Windows installations of Chrome.

Built-in Security Key Management on Windows

Platforms
Supported versions of Windows 10 or 11, and Windows Server

Capabilities
Set or change the PIN, Manage fingerprint templates, Reset the YubiKey

Works with
All YubiKeys that support FIDO2.

Windows has supported security keys for many versions, and all the most recent releases of all supported Windows Desktop and Server support FIDO2, and make that support available to web browsers running on the platform. Windows includes built-in tools for setting and changing the PIN on FIDO2 devices like the YubiKey, as well as resetting the YubiKey. Search for "Set up Security Key" in the Start menu to find Windows' built-in FIDO2 management tools. This method of interacting with the security key does not require administrative rights.

8.1.11 Managing Discoverable Credentials

Any YubiKey with firmware 5.2.1 and higher supports viewing and deleting individual discoverable credentials (also known as *Passkeys*) that are stored on the YubiKey. A PIN must be configured, and entered each time you want to view discoverable credentials. Deleting a discoverable credential is a permanent action, and can not be undone. It is recommended to ensure that you have access to an account by other means (such as a different YubiKey) before deleting a discoverable credential for a specific account.

Managing Discoverable Credentials with Yubico Authenticator

1. Download and install Yubico Authenticator for Windows, MacOS or Linux. The iOS and Android versions of Yubico Authenticator do not support resetting the FIDO2 application of the YubiKey.
2. Insert your YubiKey or Security Key in a USB port on your computer.
3. Open Yubico Authenticator.
4. Click the menu icon (three vertical bars) in the upper left hand corner and select **WebAuthn**.
   - Windows 10 or 11 users, if prompted, enter administrator consent.

     Due to underlying OS mechanics, when using Windows 10 or 11, applications that manage FIDO2 devices need to be run as administrator in order to access FIDO2 options and/or to detect the Security Key Series keys.
   - Enter a PIN at the prompt, if a PIN is set on the device.

   Any discoverable credentials on the device are listed. Most discoverable credentials provide a way to identify the account. The URL that the credential is used for is always visible.
5. Locate the credential you want to delete, and click on the elipses (...) icon to the right of the credential.

   The Username and URL of the credential is listed again.
6. Click the **Delete Passkey** button under the credential. To cancel the deletion, click the **X Close** button.
7. Confirm deletion by clicking the **delete** button. This permanently deletes the credential.

Managing Discoverable Credentials with Google Chrome on MacOS or Linux

Note: Chrome for Windows does not support managing individual FIDO2 credentials due to Windows operating system restrictions.

To list and delete credentials:

1. Open the Chrome Settings menu. Click the 3 vertical dots.

   Alternatively, navigate to chrome://settings/securityKeys and skip to step 5.

2. Select **Privacy & Security** from the settings navigation on the left hand side.
3. Scroll down and select **Security**.
4. Scroll down and select **Manage security keys**.
5. Select **Sign-in data**.
6. Enter your YubiKey's PIN and click **Continue**.
7. Located the credential you want to delete and click the trash can icon next to it.
8. Confirm deletion by clicking the **delete** button. This permanently deletes the credential.

8.1.12 Resetting the FIDO2 Application

If the FIDO2 PIN has been forgotten, and the fingerprint sensor on the YubiKey Bio is not working, the key will need to be reset. Resetting the key will remove the PIN, but it will also destroy all the U2F and FIDO2 credentials on the YubiKey, whether they are discoverable or not. Entering the PIN incorrectly 8 times will also cause the FIDO2 application to lock.

Note: Device Support: This article applies to Yubico devices that support the FIDO2 protocol, like the YubiKey 5 Series, YubiKey 5 FIPS Series, and Security Key Series, but not to the FIDO U2F Security Key, which cannot be reset.

Before Resetting the FIDO2 Application

Once the FIDO2 application on the YubiKey has been reset, there is no way to recover the previously stored credentials or PIN. Resetting the FIDO2 application will effectively unregister your key from any accounts it was registered with using FIDO U2F or FIDO2. We therefore recommend following the steps below, prior to resetting.

Determine which accounts will be affected by a reset (see below). Log in to each of those accounts, unregister the key to be reset, and then double-check that you are still able to log in and modify the account's 2FA settings (without the key that is to be reset). This process is easier if you have more than one key registered with your accounts, which we recommend.

Determining which accounts may be affected

To determine which of your accounts may be affected by a FIDO reset:

1. Search for each service your YubiKey is registered with in the Works With YubiKey Catalog.
2. Under each service's listing, check the security protocol support section for FIDO2/WebAuthn, Universal 2nd Factor (U2F), or similar. Services that indicate support for these may be affected by a FIDO2 reset.

   For instance, Google's listing in the WWYKC has both of these listed, indicating it would be affected by a reset.

Services that only list Yubico OTP, OATH-TOTP, etc., and do not include any of the aforementioned protocols should not be affected.

The YubiKey will return to its initial state without a FIDO2 PIN. We recommend using the Yubico Authenticator app or built-in OS support on a desktop OS to set the PIN prior to using the YubiKey again.

Resetting the FIDO2 Application Using YubiKey Manager (GUI)

1. Download and install YubiKey Manager.
2. Insert your YubiKey or Security Key into an available USB port on your computer.
3. Open YubiKey Manager.
4. Navigate to Applications > FIDO2.
5. Click “Reset FIDO” > “YES”.
6. Follow the YubiKey Manager prompts to remove, re-insert, and touch your key.

Resetting the FIDO2 Application Using Yubico Authenticator

If a PIN has been set, Yubico Authenticator requires that PIN in order to reset the FIDO2 application.

Note: When using Windows 10 or 11, applications that manage FIDO2 devices need to be run as administrator in order to access FIDO2 options and/or to detect the Security Key Series keys. If you are using Windows and do not have administrative access, consider using the built-in security key management features of Windows.

1. Download and install Yubico Authenticator for Windows, MacOS or Linux. The iOS and Android versions of Yubico Authenticator do not support resetting the FIDO2 application of the YubiKey.
2. Insert your YubiKey or Security Key into an available USB port on your computer.
3. Open Yubico Authenticator.
4. In the upper left hand corner, click on the menu icon (three vertical bars) and then select “WebAuthn”.
5. You will be prompted for a PIN if a PIN is set on the device; however, if a PIN has not been set, entering the PIN is not required for a reset.
6. In the upper right hand corner of the Authenticator, click on the icon for the device you are using, and select “Reset FIDO”.
7. Follow the on-screen instructions.

Resetting the FIDO2 Application Using Windows Built-in Security Key Management

Windows 10 and 11 provide built-in tools to manage FIDO2 devices.

1. Open the Start menu and select “Set up security key”.

   Alternatively, open Windows Settings and navigate to “Accounts” > “Sign-in options” > “Security Key”.
2. Click the “Manage” button.
3. When prompted, touch your security key.
4. Click “Reset security key”, and follow the on-screen prompts.

Resetting the FIDO2 Application Using Google Chrome on MacOS or Linux

Chrome for Windows does not support resetting the FIDO2 application because of Windows OS restrictions.

1. Open the Chrome Settings menu by clicking on the 3 vertical dots on the upper right of the browser, next to the URL field.

   Alternatively, navigate to chrome://settings/securityKeys and skip to step 5.

2. Select "Privacy & Security" from the settings navigation on the left hand side.
3. Scroll down and select "Security".
4. Scroll down and select "Manage security keys".
5. Click "Reset your security key", and follow the on-screen prompts.

8.1.13 Setting or Changing the FIDO2 PIN

For a general description of the FIDO2 PIN, see *FIDO2 PINs and Fingerprint Templates*.

Note that the FIDO2 PIN is independent from the PIV PIN, and may be set to a different value, or not set at all. Changing the FIDO2 PIN will not change the PIV PIN, and vice-versa.

Setting or Changing the FIDO2 PIN Using YubiKey Manager

When using Windows 10 or 11, applications that manage FIDO2 devices need to be run as administrator in order to access FIDO2 options and/or to detect the Security Key Series keys If you are using Windows and do not have administrative access, consider using the built-in security key management features of Windows.

1. Download and install YubiKey Manager.
2. Insert your YubiKey or Security Key into an available USB port on your computer.
3. Open YubiKey Manager.
4. Navigate to Applications > FIDO2.
5. Click on "Set PIN" or "Change PIN"
6. Follow the prompts from YubiKey Manager to set or change the PIN

Setting or Changing the FIDO2 PIN Using Yubico Authenticator

When using Windows 10 or 11, applications that manage FIDO2 devices need to be run as administrator in order to access FIDO2 options and/or to detect the Security Key Series keys. If you are using Windows and do not have administrative access, consider using the built-in security key management features of Windows.

The iOS and Android versions of Yubico Authenticator do not support setting the FIDO2 PIN.

1. Download and install Yubico Authenticator for Windows, MacOS or Linux.
2. Insert your YubiKey or Security Key into an available USB port on your computer.
3. Open Yubico Authenticator.
4. In the upper left hand corner, click on the menu icon (three vertical bars) and select "WebAuthn".
5. You will be prompted for a PIN if a PIN is set on the device. In the upper right hand corner, click on the icon representing your device, and select "Set PIN" or "Change PIN", depending on whether your device already has a PIN configured.

Setting or Changing the FIDO2 PIN Using Windows Built-in Security Key Management

Windows 10 and 11 provide built in tools to manage FIDO2 devices without needing administrative access.

1. Open the start menu and search for "Set up security key".

   Alternatively, open Windows Settings and navigate to "Accounts" -> "Sign-in options" -> "Security Key"

2. Click on the "Manage" button.
3. Touch your security key as prompted
4. Under "Security key PIN", Click on "Add" or "Change", and follow the on-screen prompts.

Setting or Changing the FIDO2 PIN Using Google Chrome on MacOS or Linux

Chrome for Windows does not support setting or changing the FIDO2 PIN because of Windows OS restrictions.

1. Open the Chrome Settings menu by clicking on the 3 vertical dots.

   Alternatively, navigate to chrome://settings/securityKeys and skip to step 5.

2. Select "Privacy & Security" from the settings navigation on the left hand side.
3. Scroll down and select "Security".
4. Scroll down and select "Manage security keys".
5. Click on "Create a PIN", and follow the on-screen prompts to set or change the FIDO2 PIN.

8.1.14 Enrolling Fingerprints on the YubiKey Bio

Videos demonstrating how to enroll fingerprints in the YubiKey Bio can be found here: https://www.yubico.com/setup/yubikey-bio-series/#enrollment

8.2 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.

8.3 Smart Card (PIV Compatible)

For an overview of the PIV features that became available with the 5.7.x firmware, see *PIV Enhancements*.

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.

8.3.1 Default Values

- PIN: `123456`
- PUK: `12345678`
- Management Key:
  - Firmware 5.0.x-5.6.x: `010203040506070801020304050607080102030405060708` (3DES)
  - Firmware 5.7.x: `010203040506070801020304050607080102030405060708` (AES-192)

Note: The Management Key changed to AES-192 in firmware 5.7, however the default value itself did not change.

8.3.2 Supported Algorithms

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

Table 2: **Available PIV Algorithms per Firmware Version**

	Firmware Versions	
Algorithm (Identifier)	5.0.x - 5.6.x	5.7.x
RSA-1024 (0x06)	Yes	Yes
RSA-2048 (0x07)	Yes	Yes
RSA-3072 (0x05)		Yes
RSA-4096 (0x16)		Yes
ECC P-256 (0x11)	Yes	Yes
ECC P-384 (0x14)	Yes	Yes
Ed25519/x25519 (0xe0)		Yes

Note: The algorithms RSA-1024 and X25519 are not allowed on FIPS 140-3 capable devices.

8.3.3 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.

8.3.4 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 through 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 the device with instruction f9. This slot is not cleared on reset, but can be overwritten.

Advanced Key Management Functions

With the 5.7.x firmware, the PIV application supports advanced key management functions such as moving and deleting keys:

- The ability to move keys enables retaining retired encryption keys on the device to decrypt older messages.
- The ability to delete keys enables destroying key material without overwriting with bogus data or resetting the PIV application.

8.3.5 Import a Key

For more information, see Import asymmetric key pair.

8.3.6 Move a Key

Keys can be moved from any slot except F9 (attestation) to any other slot except F9 using the instruction 0xF6.

Table 3: **Moving a Key**

CLA	00
INS	F6
P1	Destination slot
	9A, 9C, 9D, 9E,
	82, 93, 84, 85, 86, 87, 88, 89, 8A, 8B, 8C, 8D, 8E, 8F,
	90, 91, 92, 93, 94, 95
P2	Source slot
	9A, 9C, 9D, 9E,
	82, 93, 84, 85, 86, 87, 88, 89, 8A, 8B, 8C, 8D, 8E, 8F,
	90, 91, 92, 93, 94, 95
	P2

8.3.7 Delete a Key

Any key can be deleted from any slot, including F9 (Attestation) using the instruction 0xF6 with a value of 0xFF for P1.

Table 4: **Deleting a Key**

CLA	00
INS	F6
P1	FF
P2	Source slot
	9A, 9C, 9D, 9E,
	82, 93, 84, 85, 86, 87, 88, 89, 8A, 8B, 8C, 8D, 8E, 8F,
	90, 91, 92, 93, 94, 95
	F9

8.3.8 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

8.3.9 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 ADCS 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
```

Notes on 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 as the attestation attests that a private key only exists on that particular 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 fails:

```
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 firmware 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 ykpiv.h, which includes the functions `ykpiv_get_metadata()` and `ykpiv_util_parse_metadata()`. The source code of these functions is available in the file ykpiv.c.

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

The function `_ykpiv_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 PIV extensions section, and the TLV-encoded output is returned in the `ykpiv_get_metadata()` parameter data.

Table 5: 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 `ykpiv_transfer_data()` directly for low-level APDU communication with the YubiKey's PIV applet.

The function `ykpiv_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 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 parses out the public keys for all PIV key slots. If PIV attestation is supported, the PIV attestation certificate is used for parsing out the public key.

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

If both attestation and PIV metadata fail, in that order, YKCS11 falls 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.

8.3.10 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 5 Series have a PIV attestation root certificate authority different than previous YubiKeys. 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 through 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.

With the release of YubiKey firmware version 5.7 the YubiKey PIV Management Key is AES-192 by default.

8.4 OATH

For an overview of the OATH features that became available with the 5.7.x firmware, see *5.7 Firmware Specifics*.

The OATH application can store up to 64 OATH credentials on firmware 5.7.0 and later (32 on older firmware), 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 through the CCID channel. In order to manage these credentials and read the OTPs generated by the YubiKey, requires the Yubico Authenticator.

To restrict access to the OTPs, set an access code for the OATH application.

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

8.4.1 HOTP and TOTP

Both **OATH-TOTP** and **OATH-HOTP** credentials are described in detail in the OATH Overview.

8.5 OpenPGP

For an overview of the OpenPGP features that became available with the 5.7.x firmware, see *5.7 Firmware Specifics*.

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 and later 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.

8.5.1 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 the algorithms listed below in **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 cardholder certificates, refer to Enhancements to OpenPGP Support.

8.5.2 RSA Algorithms

- RSA-1024 (removed in firmware 5.3.2 and later)
- RSA-2048
- RSA-3072 (requires GnuPG version 2.0 or higher)
- RSA-4096 (requires GnuPG version 2.0 or higher)

8.5.3 Default Values

- PIN: `123456`
- Admin PIN: `12345678`

8.6 OTP

For an overview of the OTP features that became available with the 5.7.x firmware, see *5.7 Firmware Specifics*.

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.

8.6.1 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).

8.6.2 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).

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

8.7 YubiHSM Auth

8.7.1 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), see *Yubico Secure Channel Technical Description*. YubiHSM Auth is supported by YubiKey firmware version 5.4.3 and above.

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.

8.7.2 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).

8.7.3 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.

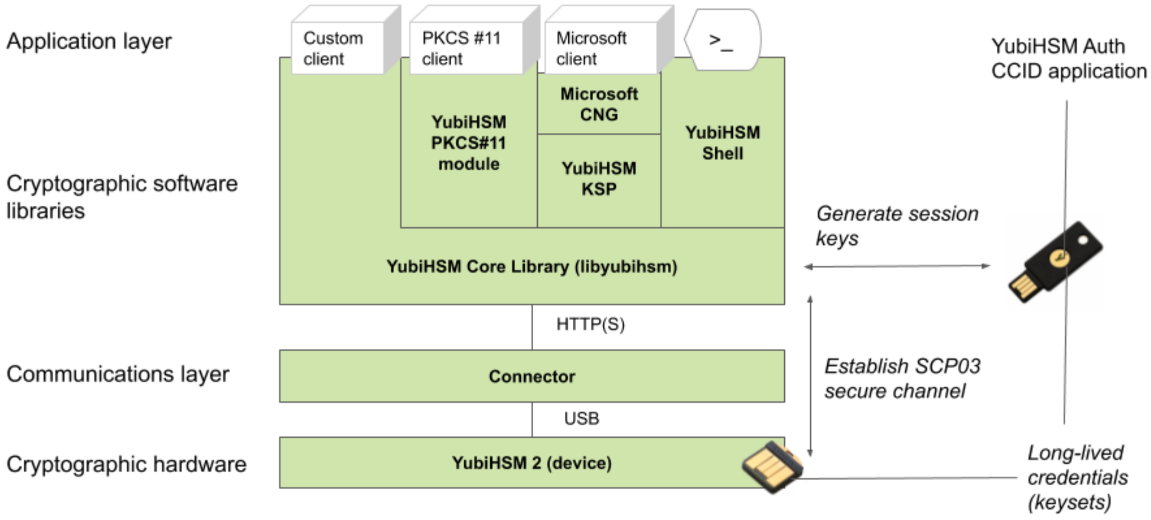
Those long-lived key sets are used by the YubiHSM Auth application to derive a set of three session-specific AES-128 keys using the challenge-response protocol as defined in SCP03:

- 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.

8.7.4 Architecture Overview

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



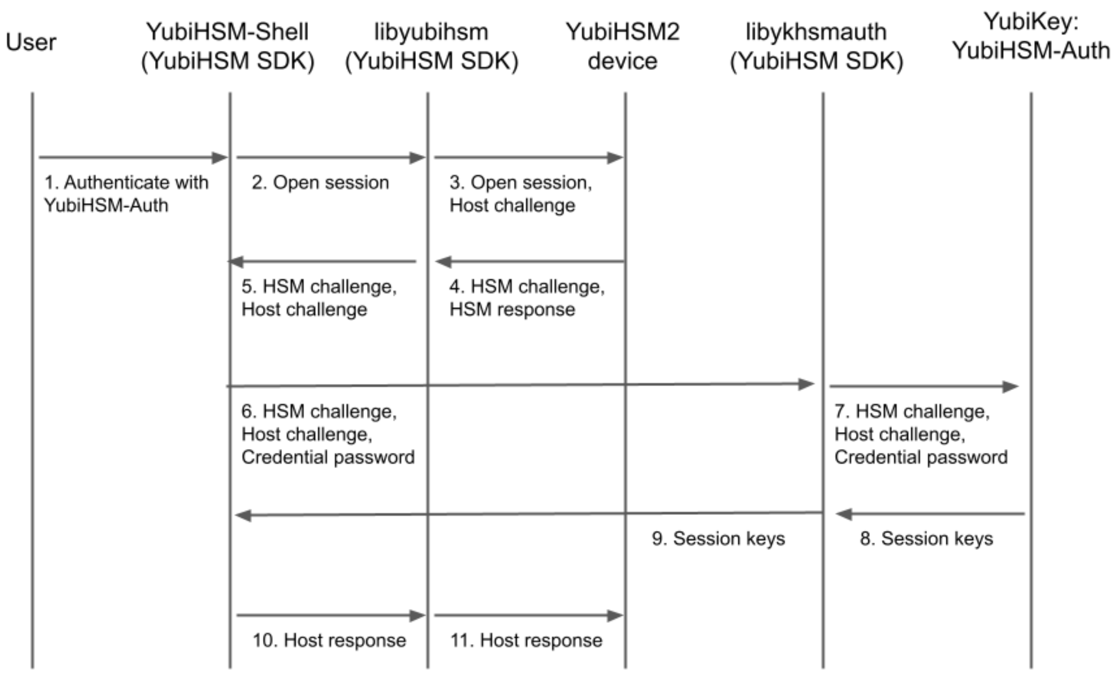
The identical long-lived credentials (key sets) are protected in both the YubiKey YubiHSM Auth application and in the YubiHSM 2. The YubiHSM-Shell software tool can be used for generating the key sets in the YubiHSM 2, and the YubiHSM-Auth software tool can be used for importing the same key sets to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

8.7.5 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.


User
YubiHSM-Shell (YubiHSM SDK)
libyubihsm (YubiHSM SDK)
YubiHSM2 device
libykhsmauth (YubiHSM SDK)
YubiKey: YubiHSM-Auth
1. Authenticate with YubiHSM-Auth
2. Open session
3. Open session, Host challenge
5. HSM challenge, Host challenge
4. HSM challenge, HSM response
6. HSM challenge, Host challenge, Credential password
7. HSM challenge, Host challenge, Credential password
9. Session keys
8. Session keys
10. Host response
11. Host response

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.

8.7.6 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.

To generate and store the equivalent YubiHSM Auth credentials on the YubiKey, use the `yubihsm-auth` command line tool. 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`. This requires the user 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 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 the YubiHSM User Guide, section YubiHSM Auth.

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

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.

Click for Yubico Support.

CHAPTER

NINE

TOOLS AND TROUBLESHOOTING

9.1 Managing Applications

9.1.1 Enabling/Disabling

The YubiKey Manager (ykman) 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. Select the ones you want enabled, unselect the ones you want disabled, and click **Save Interfaces**.

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

9.1.2 Locking

Once the desired applications have been selected, you can set a lock code to prevent changing the set of enabled applications. To do this, use 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/>* `_.

9.2 YubiKey Manager (ykman)

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

9.2.1 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 allowed through physical interface.
- Setting or changing the FIDO2 PIN, as well as resetting the FIDO application.
- Managing the credentials in the OTP application.

9.2.2 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 and GUI Guide.

9.3 Yubico Authenticator

Yubico Authenticator is used to manage credentials on OATH applications and it lists OTPs generated by the YubiKey. Yubico Authenticator provides the time element for generating OTPs for OATH-TOTP credentials because the YubiKey does not have a battery and cannot track time. The Yubico Authenticator 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.

9.4 YubiKey Smart Card Minidriver

The YubiKey Smart Card Minidriver extends the PIV / Smart Card application for YubiKey on Windows. It facilitates deployment and management. Key benefits include:

- Enroll the YubiKey using standard Windows utilities.
- Auto-enrollment for self-provisioning and automatically renewing a YubiKey.
- Multiple authentication certificates on one YubiKey.
- Change the PIN from the Ctrl+Alt+Del menu.
- Unblock the PIN using the PUK at the Windows logon screen.

To get started with the YubiKey Smart Card Minidriver, see the deployment guide

Note: To use PIV / Smart Cards with YubiKey 5 Series, requires YubiKey Smart Card Minidriver version 4.0 or later.

9.5 Troubleshooting

For any issues with a key from the YubiKey 5 Series, refer to the Knowledge Base and search for the issue. If your issue is not listed in the Knowledge Base, or if you have any technical questions, submit a request with Yubico Support.

Click for Yubico Support.

CHAPTER

TEN

NFC ID CALCULATION TECHNICAL DESCRIPTION

10.1 YubiKey for 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 unlocked.

10.2 NFC ID Calculation for YubiKey v5.2.x and Earlier

For YubiKey with firmware version 5.2.x and earlier, 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, which results in non-unique values.

10.3 NFC ID Calculation for YubiKey v5.3.0 and Later

For YubiKeys with firmware version 5.3.0 and later, the NFC ID calculation so 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 Yubico Security Keys, which do not have serial numbers, 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.

Click for Yubico Support.

CHAPTER

ELEVEN

SECURE CHANNEL PROTOCOL (SCP03 AND SCP11)

11.1 Yubico Secure Channel Technical Description

11.1.1 Introduction

Yubico has implemented a subset of the (GlobalPlatform Secure Channel Protocol 03) specification, specifically, the most secure implementation including command and response message authentication code (MAC) and encryption. This is available in YubKey firmware 5.3.0 and later.

Yubico has implemented a subset of the (GlobalPlatform Secure Channel Protocol 11) specification, specifically, the most secure implementation including command and response message authentication code (MAC) and encryption. This is available in YubKey firmware 5.7.0 and later.

At the highest level, implementing a secure channel consists of providing overhearing and tampering resistance to information sent between an external service, like a Credential Management Solution (CMS) and a smart card. Overhearing resistance is provided by using a unique, private symmetric AES key to apply AES encryption on all commands sent and received. Tamper resistance is provided by using an AES key, unique to each session, to send a securely encrypted MAC of both the commands and associated responses. 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.

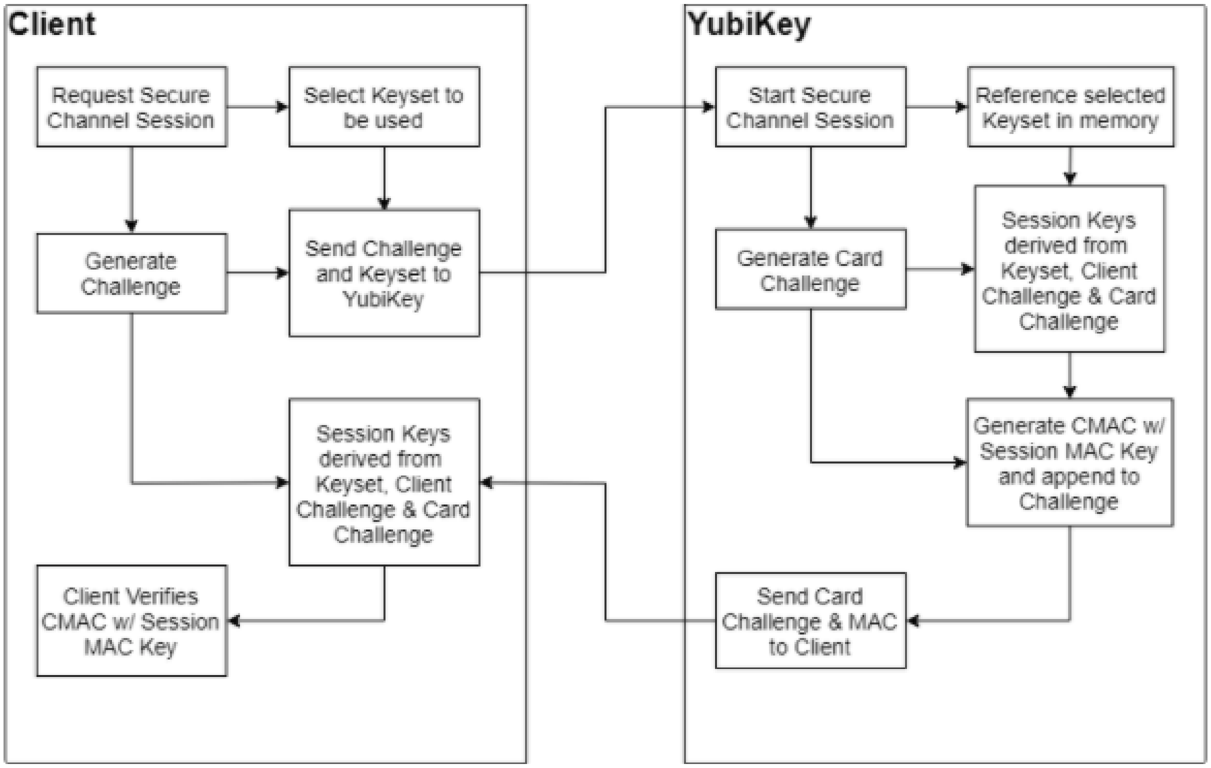
On the YubiKey, all of the secure channel operations occur within the secure cryptographic processor. The plain text of the communication is never exposed to outside observers.

Flow when Initializing a Secure Channel on a YubiKey

11.1.2 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. It sits between the input of APDU commands sent into the secure element and the applications on the YubiKey. As such, any command that 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), because they do not support communication over the CCID channel.

The SCP11 addition is compatible with existing SCP03 setups, however the long lived encryption keys are symmetric when using SCP03 (AES-128) and asymmetric when using SCP11 (ECC P-256).


Client
Request Secure Channel Session
Select Keyset to be used
Generate Challenge
Send Challenge and Keyset to YubiKey
Session Keys derived from Keyset, Client Challenge & Card Challenge
Client Verifies CMAC w/ Session MAC Key
YubiKey
Start Secure Channel Session
Reference selected Keyset in memory
Generate Card Challenge
Session Keys derived from Keyset, Client Challenge & Card Challenge
Generate CMAC w/ Session MAC Key and append to Challenge
Send Card Challenge & MAC to Client

11.1.3 Transport Keys and Session Keys

Key	Usage	Creation
Static Secure Channel Encryption Key (Key-ENC)	Generate session key for Decryption/ Encryption (AES-128)	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-128)	Imported from Trusted source
Data Encryption Key (Key-DEK)	Sensitive Data Decryption (AES-128) used to encrypt 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-128 or ECC P-256 keys as defined in the SCP03 or SCP11 specifications, respectively; 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.

11.1.4 Transport Keys

A Transport Key set is made of 3 AES-128 or ECC P-256 keys:

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

Transport key sets are used for establishing the secure channels and 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. To import new transport key sets, establish a secure channel with the security domain. Do this with either a previously loaded transport key set or the default transport key set.

When initializing a session, the Secure Channel Encryption Key is used to generate the Session Secure Channel Encryption Key for use during that session. Similarly, the Secure Channel MAC Key is used to generate the Session Secure Channel MAC key for the Command and Session Secure Channel MAC Key for Response. The Data Encryption Key is only used when importing new transport key sets; the imported keys 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.

11.1.5 Session Keys

The Session Key set is made of 3 AES-128 or ECC P-256 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 GlobalPlaftorm Secure Channel Protocol (SCP) 03 specification, 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 that can be verified to ensure the data was not tampered with in transit, nor is a command being replayed.

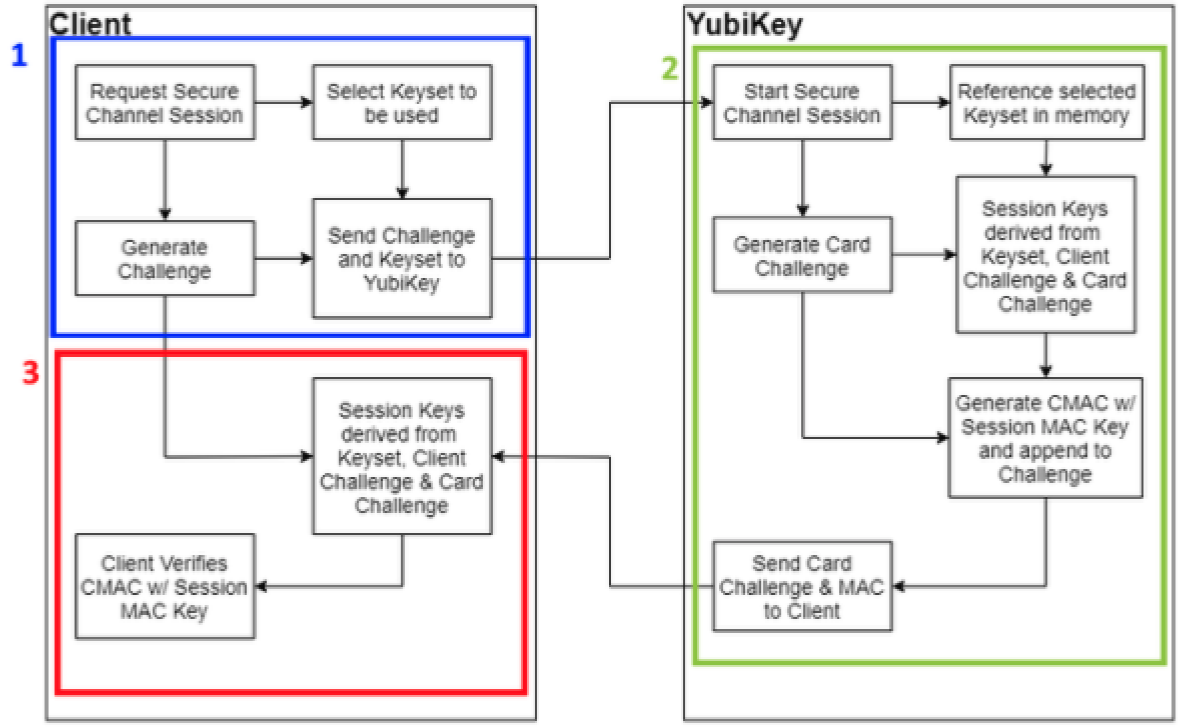
The Response MAC is generated using the encrypted response APDU from the YubiKey and the Session MAC key for Response. Each Response MAC also includes a value derived from the original command MAC sent from the client, providing a verification that the data included in the response corresponds to the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

11.1.6 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 is 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 ends the session and removes any previously granted authorizations.

The AID to access the Secure Channel functionality on the YubiKey is A000000308. The EXTERNAL_AUTHENTICATE command with security level C-DECRYPTION, R-ENCRYPTION, CMAC and R-MAC is the only supported option.

To begin, the client identifies the function they wish to communicate with and sends the **Initialize Update** command.


Client
1
Request Secure Channel Session
Select Keyset to be used
Generate Challenge
Send Challenge and Keyset to YubiKey
3
Session Keys derived from Keyset, Client Challenge & Card Challenge
Client Verifies CMAC w/ Session MAC Key
YubiKey
2
Start Secure Channel Session
Reference selected Keyset in memory
Generate Card Challenge
Session Keys derived from Keyset, Client Challenge & Card Challenge
Generate CMAC w/ Session MAC Key and append to Challenge
Send Card Challenge & MAC to Client

YubiKey Secure Channel Initialize Update Flow

Step 1

When a client starts the process of establishing a secure channel with the YubiKey, it selects the Transport key set on the YubiKey to be used, and generates a unique challenge. This challenge is used by the client to derive the session keys that are used going forward.

The `Initialize Update` command, including the challenge and Transport Key set identifier, is sent from the client to the YubiKey CCID function; specifically, the YubiKey CCID function for which the secure channel is being established.

Step 2

When the YubiKey receives an `Initialize Update` command as a client is starting a session with any CCID facing function, the YubiKey directs communication between the client and the YubiKey to the Secure Channel function for the remainder of the session.

The Secure Channel Function uses the Transport Key Identifier to select the Transport Key set in memory to use.

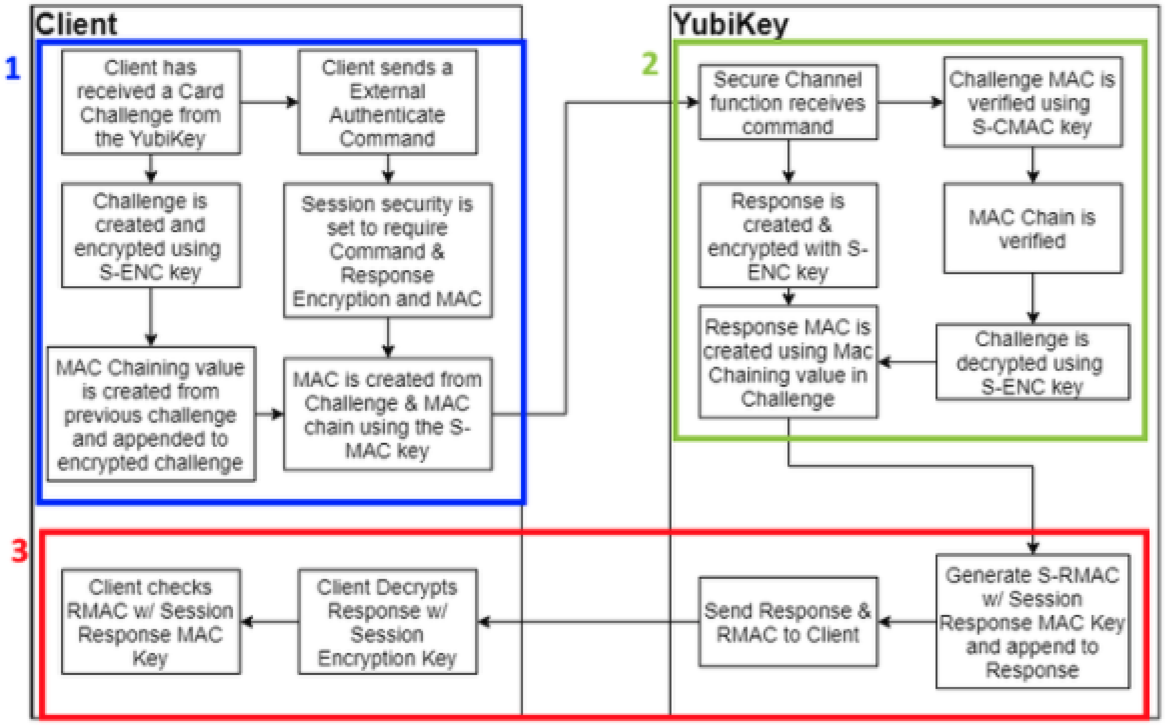
The YubiKey derives the Session keys for Encryption and MAC by generating a card challenge, then using that plus the challenge provided from the client and the selected Transport Key set.

The YubiKey then generates 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

When the client receives the response from the YubiKey, it 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, run the **External Authenticate** command. Security level C-DECRYPTION, R-ENCRYPTION, CMAC and R-MAC is the only supported option.


Client
1
Client has received a Card Challenge from the YubiKey
Client sends a External Authenticate Command
Challenge is created and encrypted using S-ENC key
Session security is set to require Command & Response Encryption and MAC
MAC Chaining value is created from previous challenge and appended to encrypted challenge
MAC is created from Challenge & MAC chain using the S-MAC key
YubiKey
2
Secure Channel function receives command
Challenge MAC is verified using S-CMAC key
Response is created & encrypted with S-ENC key
MAC Chain is verified
Response MAC is created using Mac Chaining value in Challenge
Challenge is decrypted using S-ENC key
3
Client checks RMAC w/ Session Response MAC Key
Client Decrypts Response w/ Session Encryption Key
Send Response & RMAC to Client
Generate S-RMAC w/ Session Response MAC Key and append to Response

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 defines the session security settings - the YubiKey only supports the strictest option, with both commands and responses encrypted and 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

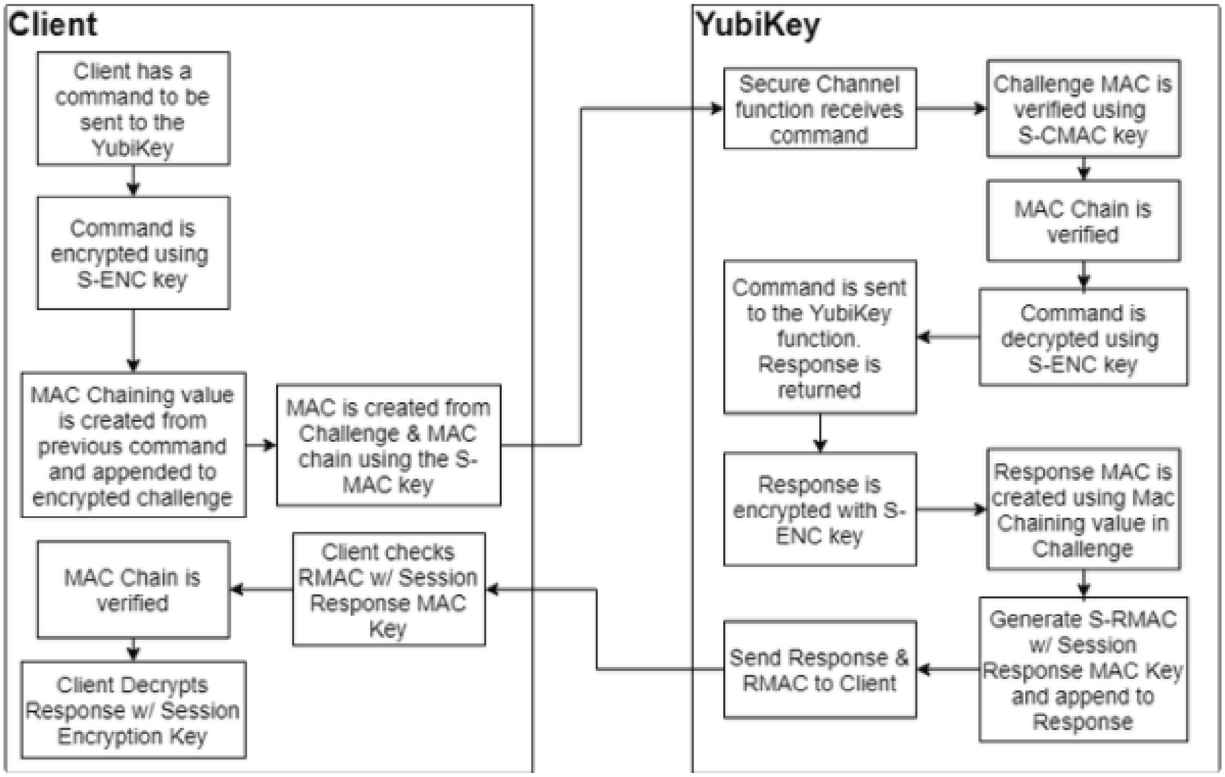
The YubiKey receives the `External Authenticate` command, and verifies the Challenge using the MAC value and S-MAC key. The MAC Chain is then verified, confirms that the client has the same session keys and that a secure channel has been created. Then the challenge is decrypted using the S-ENC key and 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. Then 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.

11.1.7 Communicating Over Secure Channel


Client
Client has a command to be sent to the YubiKey
Command is encrypted using S-ENC key
MAC Chaining value is created from previous command and appended to encrypted challenge
MAC is created from Challenge & MAC chain using the S-MAC key
MAC Chain is verified
Client checks RMAC w/ Session Response MAC Key
Client Decrypts Response w/ Session Encryption Key
YubiKey
Secure Channel function receives command
Challenge MAC is verified using S-CMAC key
MAC Chain is verified
Command is sent to the YubiKey function. Response is returned
Command is decrypted using S-ENC key
Response is encrypted with S-ENC key
Response MAC is created using Mac Chaining value in Challenge
Send Response & RMAC to Client
Generate S-RMAC w/ Session Response MAC Key and append to Response

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). Sending an unencrypted command ends the current session and removes any authorizations.

Important: Any APDU delivering instructions or data, such as a key or certificate, to a YubiKey is considered a Command APDU.

Step 2
A command MAC is created using the Session MAC key (S-MAC), the encrypted APDU, 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 called in the communication is returned.

Step 5
The response is encrypted with the S-ENC key, then a MAC chain value is 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.

11.2 Yubico Secure Channel Key Diversification and Programming

11.2.1 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 because only the Batch Master Key must be shared with the CMS system to derive the YubiKey transport key sets.

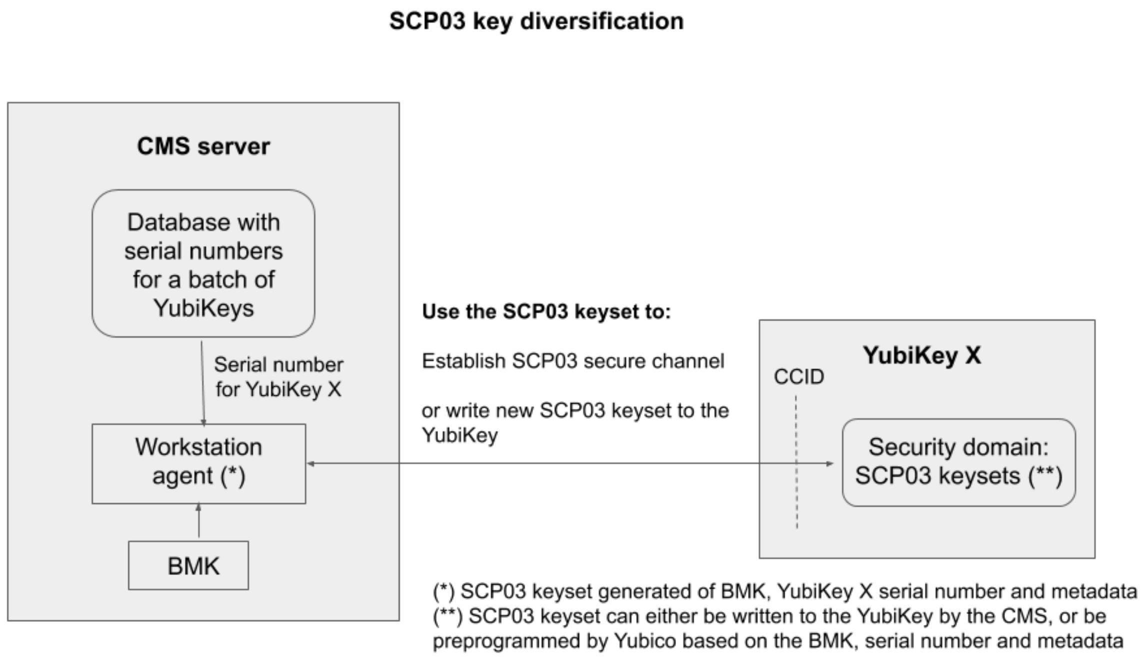
Hence, the secure channel transport key sets can be pre-programmed by Yubico, assuming that Yubico has access to the BMK of the CMS vendor.

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

11.2.2 Secure Channel and Security Domains

The YubiKey supports up to three 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 when connected to the YubiKey. This allows for YubiKeys to be shipped with a default key set, then have the key set be changed to a random set of keys at initialization. .This in turn, ensures that only the CMS server has the actual keys.


SCP03 key diversification
CMS server
Database with serial numbers for a batch of YubiKeys
Serial number for YubiKey X
Workstation agent (*)
BMK
Use the SCP03 keyset to:
Establish SCP03 secure channel
or write new SCP03 keyset to the YubiKey
YubiKey X
CCID
Security domain: SCP03 keysets (**)

(*) SCP03 keyset generated of BMK, YubiKey X serial number and metadata
(**) SCP03 keyset can either be written to the YubiKey by the CMS, or be preprogrammed by Yubico based on the BMK, serial number and metadata

11.2.3 Key Diversification Option

When purchasing YubiKeys from Yubico, there is an option to custom-configure the transport keys from the default values to values derived from details specific to the hardware and a BMK. This means these keys can be distributed with locked down key sets, and that ensures 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. This limits access to just that CMS. As with other custom configuration options, these keys can be overwritten or deleted by the customer; the keys are not "baked into" the YubiKey firmware.

Batch Master Key (BMK) Generation

Each custom order with diversified keys has a unique BMK. This 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 YubiHSM 2 can be used to generate a random AES key with the `GET PSEUDO RANDOM` command, for example: `get random 0 32`.

For more information, see the *YubiHSM 2 User Guide, Command Reference* for the GET PSEUDO RANDOM command.

Before every order with key diversification, the customer must generate and provide the BMK to Yubico by a secure means. After the YubiKeys are programmed using the BMK provided, the BMK on the Yubico programming station is destroyed. This ensures the customer has the only extant BMK. The customer must maintain and securely archive their BMK for use with 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 in this code sample:

```
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. They are used to generate a 32-byte key value and 24-byte key value respectively. The first KDF has a counter value set to `01` and the second SDK has a counter value set to `02`.

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

- `0100` for a 256-bit key
- `00C0` for a 192-bit Key (PIV Admin Key)
- `0080` for a 128-bit key (ISD Keys & Interfaces Management Key)
- `0040` for a 64-bit code (the PUK)

Labels for Key Diversification

The KDF function of separating keys uses the following labels as input. Note that these are example values, using: 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 listed values:

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. Scroll horizontally to see the full line, if needed.

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

11.2.4 Global Platform: CPLC Data

Description

Although this format is officially deprecated and not part of the Global Platform 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 does not attribute any specific meaning to 40 of those bytes. Only the first two bytes are meaningful.

Example Command

To retrieve the value use the command: Scroll horizontally, if needed.

```
opensc-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.

11.3 Yubico SCP03 Developer Guidance

This section describes how Secure Channel Protocol 3 (SCP03) works in the YubiKey. It is intended for developers integrating support for it.

11.3.1 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
- 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. It does set up a mutually authenticated and encrypted channel between the YubiKey and the host. Unencrypted commands sent over the secure channel end the session and revokes any previously issued authorizations.

For more details on SCP03, refer to the Global Platform SCP03 specifications.

11.3.2 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. Manipulating the individual keys within the set, is not allowed.

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 and the value `404142434445464748494a4b4c4d4e4f` for all three keys. These keys 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 (for example, 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, send the delete instruction with `p2=1`.

11.3.3 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 a000000151000000 - the Issuer Security Domain.

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

11.3.4 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 ascribed to any of the fields.

11.3.5 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.

11.3.6 GlobalPlatformPro

GlobalPlatformPro is a Java library and tool for interacting with smart cards supporting the Global Platform secure channel protocols.

Examples

Some of the following examples are long lines of code. Scroll horizontally, as needed.

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
```

11.3.7 gpshell

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

Examples

Gpshell works with scripts. The following code is an example of opening a channel with the YubiKey. Scroll horizontally to see the last line of code in its entirety, as needed.

```
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
   -scpimpl 96
```

11.3.8 References

- Global Platform Card Specification v2.3.1 | GPC_SPE_034
- Global Platform Secure Channel Protocol ‘03’ - Amendment D v1.2 | GPC_SPE_014
- GitHub: GlobalPlatformPro (Martin Paljak)
- SourceForge: GlobalPlatform
- SourceForge: GlobalPlatform Wiki

Click for Yubico Support.

CHAPTER

TWELVE

YUBIKEY 5 FIPS SERIES SPECIFICS

12.1 YubiKey 5 FIPS Series under FIPS 140-3

The YubiKey 5 FIPS Series based on the 5.7.x firmware is undergoing a number of changes for FIPS 140-3 submission. The most notable of these changes is that the FIPS-specific requirements are now enforced by the YubiKey.

12.1.1 PIV Changes for FIPS 140-3

In order for the PIV application to be in FIPS Approved Mode the following requirements must be met:

- The default PIN needs to be changed to an 8 character value
- The default PUK needs to be changed and remain an 8 character value
- The default Management Key needs to be changed and be set to an AES key.

Additionally,

- Creating credentials prior to the application being in FIPS Approved Mode is not acceptable and the device will refuse to create credentials until the device is in FIPS Approved Mode
- The use of RSA1024, TDEA and X25519 is not allowed
- Operations over NFC must go through a secure channel (SCP03 or SCP11).

12.1.2 FIDO2 Changes for FIPS 140-3

In order for the FIDO2 application to be in FIPS Approved Mode

- The FIDO2 PIN must be set and must be at least 8 characters.

Additionally,

- Creating credentials prior to the application being in FIPS Approved Mode is not acceptable and the device will refuse to create credentials until it is in FIPS Approved Mode
- PIN Protocol v2 must be used over NFC unless a secure channel is set up (SCP03 or SCP11)
- `alwaysUV` is permanently enabled
- U2F is disabled on FIPS-capable devices.

12.1.3 OpenPGP Changes for FIPS 140-3

In order for the OpenPGP application to be in FIPS Approved Mode the following requirements must be met:

- The default user PIN must be changed and it must be at least 8 characters
- The default admin PIN must be changed and it must be at least 8 characters
- If the Reset Code is set, it must be at least 8 characters.

Additionally,

- The use of RSA decryption, X25519 and SECP256k1 is blocked
- Changing the user PIN, admin PIN or Reset Code to a value shorter than 8 characters is blocked
- Operations over NFC must go through a secure channel (SCP03 or SCP11)
- Creating credentials prior to the application being in FIPS Approved Mode is not acceptable and the device will refuse to create credentials until it is in FIPS Approved Mode.

12.1.4 OATH Changes for FIPS 140-3

In order for the OATH application to be in FIPS Approved Mode the following requirements must be met:

- The access code must be set (minimum length of 14 bytes).

Additionally,

- Creating credentials prior to the application being in FIPS Approved Mode is not acceptable and the device will refuse to create credentials until it is in FIPS Approved Mode
- When performed over NFC, `SET CODE` and `PUT` must go through a secure channel (SCP03 or SCP11).

12.1.5 YubiHSM Auth Changes for FIPS 140-3

In order for the YubiHSM Auth application to be in FIPS Approved Mode the following requirements must be met:

- The default admin code must be changed.

Additionally,

- Creating credentials prior to the application being in FIPS Approved Mode is not acceptable and the device will refuse to create credentials until the device is in FIPS Approved Mode
- Operations performed over NFC must go through a secure channel (SCP03 or SCP11).

12.1.6 Security Domain (SCP03 and SCP11) Changes for FIPS 140-3

In order for the Security Domain application to be in FIPS Approved Mode the following requirements must be met:

- The default key set must be changed.

Additionally,

- Until the application is in Approved mode, the default key set can only be used to establish a secure channel with the Security Domain itself and only for the purpose of loading a new key set. This operation must be performed over USB.

12.2 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. This provides both tamper-evidence and tamper-resistance. In turn, 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 for securing the various functions. 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 *FIPS Level 1 vs FIPS Level 2*.

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 a key from the YubiKey 5 Series. The user experience with YubiKey 5 FIPS Series deployed under FIPS 140-2 Level 2 is more complicated.

NIST classified the YubiKey 5 Series FIPS as “composite authenticators”. As such, no device in this series can be taken out of the FIPS-approved mode after initialization without zeroing the function. This means that once the YubiKey is correctly configured, it remains in the correct configuration. This 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.

12.2.1 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

12.2.2 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 that 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 requires a secure channel. For more information on Secure Channel (SCP03) in connection with YubiKeys, see the *YubiKey 5 Series Technical Manual*, Secure Channel Technical Description. For more information on SCP03 requirements from NIST, see NIST Special Publication 800-63C and NIST Special Publication 800-63B.

For a YubiKey 5 FIPS Series to be operating as a security key in FIPS-approved mode, in a FIPS 140-2 Level 2 authenticator in a FIPS environment, all of the applications must be in a FIPS-approved operation mode.

By default, not all of the applications on the YubiKey 5 FIPS Series are in FIPS operation mode. Before deploying the YubiKey 5 FIPS Series in a secured environment to end-users, the person with the crypto officer role 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 operation mode.

Every function of the YubiKey must require permissions defined by a 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 and GUI Guide: https://docs.yubico.com/ykman/

  PIV

  The PIV application has its credentials set to default values, and is therefore already in a FIPS-approved mode.

  OTP

  OATH and WebAuthn: To be in a FIPS-approved mode, the OTP, OATH and WebAuthn applications must have their respective credentials set.

Note: Using U2F is not allowed when the YubiKey 5 FIPS Series is deployed as a 140-2 Level 2 authenticator.

Note: It is highly recommended 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, even if FIPS 140-2 Level 2 does not explicitly require it.

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: FIPS 140-2 with YubiKey 5 FIPS Series*
- *OATH: FIPS 140-2 with YubiKey 5 FIPS Series*
- *PIV: FIPS 140-2 with YubiKey 5 FIPS Series*
- *OpenPGP*: *Protocols and Applications*
- *FIDO: FIPS 140-2 with YubiKey 5 FIPS Series* (WebAuthn)

12.3 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.

1. Trigger the YubiKey to produce the credential in the first slot by briefly touching the metal contact of the YubiKey.
2. 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.

12.3.1 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: the first 12 characters are the public identity that a validation server uses to link to a user, the remaining 32 characters are the unique passcode that is changed every 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 Modified Hexadecimal (Modhex) 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 through “scan codes” rather than actual characters. The device, where the YubiKey is connected, translates the scan codes into keystrokes.

12.3.2 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. A short touch (1~3 seconds) on the gold contact triggers the output of OTP slot 1. A long touch (+3 seconds) triggers the output of OTP slot 2.

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 that 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. Access codes are required 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 and 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. Use the command:

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

where -

`<access code>` is the access code to be set.

For the characteristics of the access code, see *Credentials and Permitted Values*.

`[OTP Slot]` is either 1 or 2 corresponding to the OTP configuration being applied to OTP slot 1 or OTP slot 2.

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 corresponding to the OTP configuration being applied to OTP slot 1 or OTP slot 2.

12.4 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. They can only be accessed through the CCID channel.

When an OATH-HOTP credential is programmed, the OTP is generated using the standard RFC 4226 HOTP algorithm and the YubiKey automatically types 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, requires the Yubico Authenticator. The Yubico Authenticator is supported on Windows, Linux, macOS, Android and iOS.

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

For an application to be in a FIPS-approved mode requires an Authentication Key that protects access to the YubiKey 5 FIPS Series OATH application. To get the permitted values for the following operation, see *Credentials and Permitted Values*.

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 and 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.

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

12.5.1 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.

12.5.2 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 storing resident credentials. 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.

See *Locking FIDO2 Credentials*.

12.5.3 Placing the WebAuthn Application in FIPS-approved Mode

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

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

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

where `<PIN>` is the WebAuthn PIN to be set. See *Credentials and Permitted Values* for PIN requirements.

U2F

The YubiKey 5 U2F FIPS application cannot be used in FIPS 140-2 Level 2 mode. Instead of the U2F functionality, use the FIDO WebAuthn application. FIPS-certified services should not call the U2F functionality; nonetheless, disable the U2F function 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**, secure access to it 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 the functions that are accessible over NFC or USB. The lock code cannot be recovered if lost. Losing the lock code makes the YubiKey permanently inaccessible.

12.6 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 and 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`.

12.6.1 Default Values

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

12.6.2 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

12.6.3 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.

12.6.4 Slot Information

The keys and certificates for the smart card application are stored in slots. The PIN policies 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 on documents, email, files, and executable signings. The end user PIN is required perform any private key operations. 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 through 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 for decrypting 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.

12.6.5 Attestation

Attestation enables you to verify that a key on the smart card application was generated on the YubiKey rather than being imported. If the key was generated on the YubiKey, an X.509 certificate was created for the key. 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

12.6.6 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.

12.6.7 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 ADCS with YubiKey minidriver or a third party tool. The credential management tool replaces the default values by automatically setting a random value for the management key and PUK and allows 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, use the YubiKey Manager (ykman).

- Download the YubiKey Manager tool: https://www.yubico.com/products/services-software/download/yubikey-manager/
- YubiKey Manager (ykman) CLI and 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

`<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.

12.7 FIPS Level 1 vs FIPS Level 2

The YubiKey 5 FIPS Series is certified in two modes of operations:

- Configuration which meets the requirements for FIPS Level 1
- 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.

12.7.1 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.

Click for Yubico Support.

CHAPTER

THIRTEEN

YUBIKEY 5 CSPN SERIES SPECIFICS

13.1 CSPN Mode Configuration

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

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

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

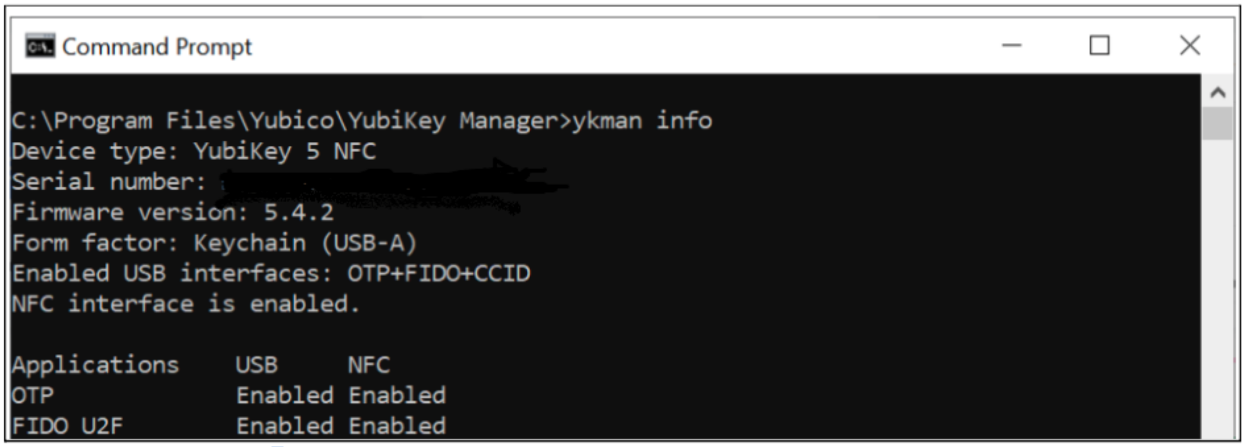
For each section there is a summary of the YubiKey application, how to operate it in a CSPN-approved mode, and how the the application can be configured.

13.1.1 Listing the Applications on the YubiKey 5

To obtain a list of all applications on the YubiKey 5, use ykman's command line YubiKey Manager. To do so, in a command prompt, execute the command `ykman info`.

The output contains general information about the YubiKey 5, such as the current firmware version, but also all of the available applications, both enabled and disabled. An example of this command is shown in the screenshot below.

Note: The Security Domain application is hidden from the user and therefore not listed by ykman.



```
C:\Program Files\Yubico\YubiKey Manager>ykman info
Device type: YubiKey 5 NFC
Serial number: [redacted]
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

13.1.2 Password Strength

Adherence to ANSSI's guidelines on password strength is highly recommended whenever applicable, particularly with any of the YubiKey 5 applications.

13.1.3 Configuration Environment

Configuration of the YubiKey can be performed in two different areas:

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

13.2 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 following sections.

13.2.1 Yubico OTP

Feature Summary

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

- 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 that must be shared by the user between the YubiKey 5 and the verification server during 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, OPTs Explained.

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. For example, username/password. The details for such an authentication scheme are beyond the scope of this document.

Once a Yubico OTP application is configured, set an access code to protect the key material and configuration. More details for such a configuration are described in the following section.

Technical Configuration

In order to protect the Yubico OTP credentials, use the YubiKey Manager (ykman).

To protecting the credentials, use the command:

```
ykman otp  --access-code <value> yubiotp <slot> --public-id <value> --generate-private-
↪id --generate-key
```

Where `--access-code` parameter is set to a six byte long hex value.

See the figure below for an example command line interaction for creating protected Yubico OTP credentials with ykman.

A code is now required for any operation that requires access to the Yubico OTP credentials.

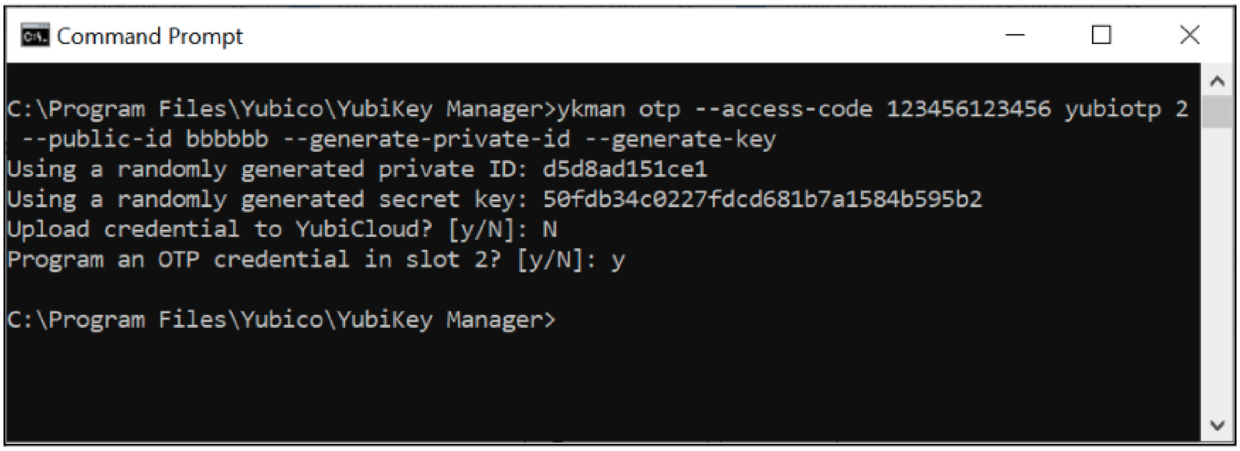
To **delete credentials**, use the command:

`ykman otp --access-code <value> delete [1|2]`

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

`ykman otp --access-code <value> settings [OPTIONS] [1|2]`

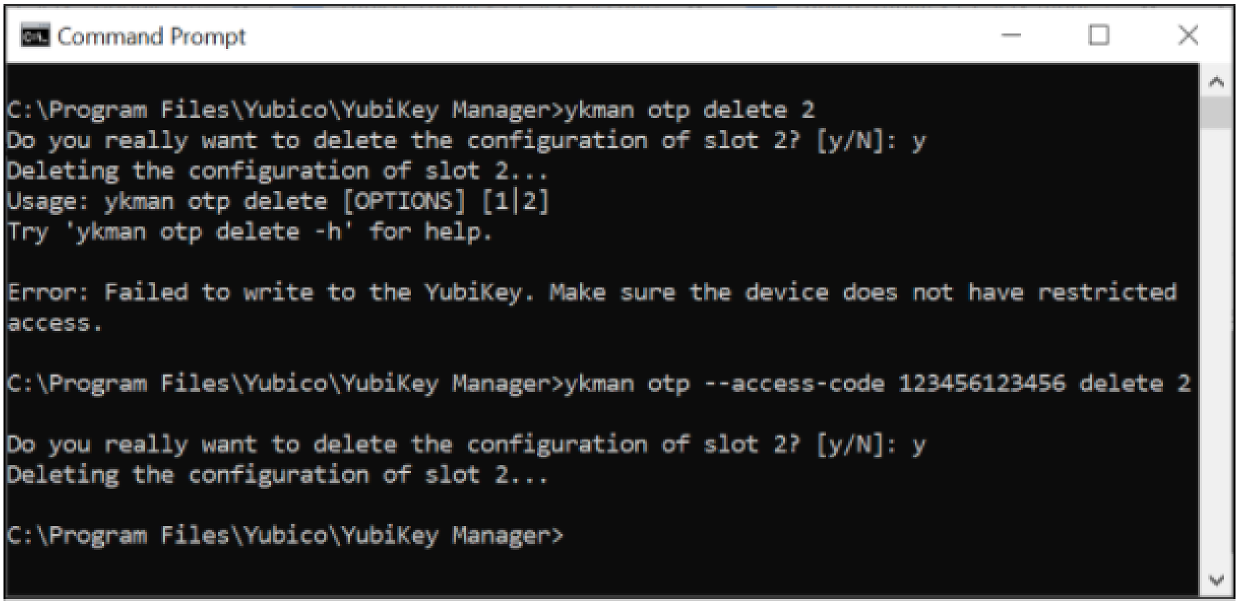
For instance, it is not possible to now delete the Yubico OTP credentials without providing the correct access code. The screenshot below is another example of how to use the ykman command line for deleting protected Yubico OTP credentials. The first attempt fails because `--access-code` is not provided, but the second attempt succeeds when the flag `--access-code` is set.



```
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



```
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

13.2.2 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 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, Challenge and Response.

CSPN Approved Mode

To operate the YubiKey 5 in a CSPN approved mode:

Identify the user by a first factor authentication scheme. For example, username/password. The details for such an authentication scheme are beyond the scope of this document.

Set usage of the YubiKey 5 touch sensor to required when configuring the Challenge-Response application.

When the Challenge-Response application is enabled on the YubiKey 5, set an access code to protect both the secret key and configuration. More details for such a configuration is described in the following section.

Technical Configuration

In order to protect the Challenge-Response credentials and enforce the touch sensor, use the command line YubiKey Manager (ykman).

Use the command:

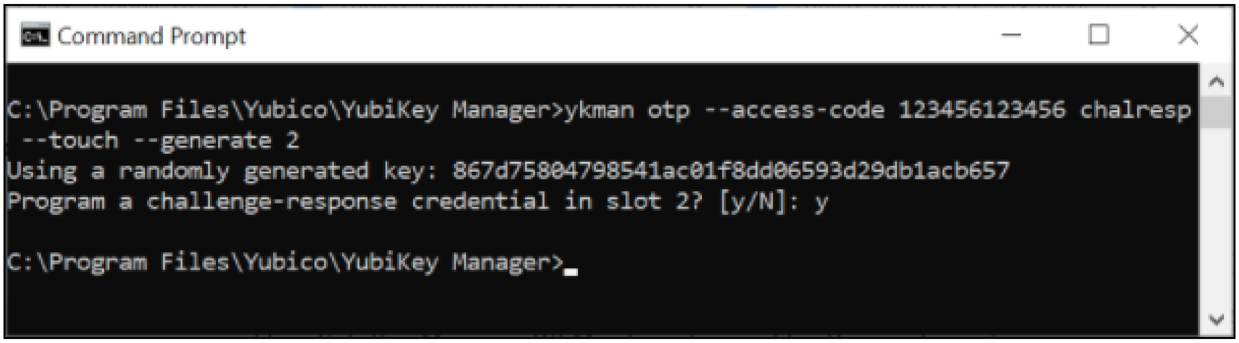
```
ykman otp --access-code <value> chalresp --touch --generate <slot>
```

where -

`--access-code` for protecting the credentials. Set the `--access-code` parameter to a six byte long hex value.

`--touch` for requesting proof of user presence.

See the figure below for an example command line interaction for creating protected Challenge-Response credentials requiring touch with ykman.



```
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

To **delete credentials**, use the command

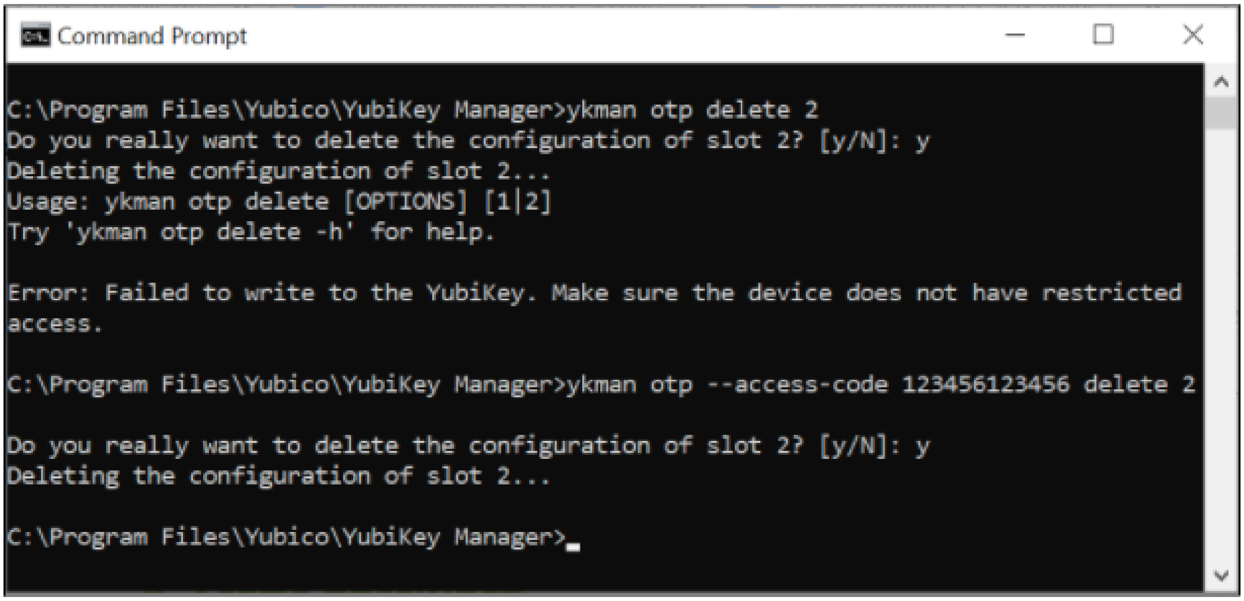
```
ykman otp --access-code <value> delete [1|2]
```

To **Change the settings**, use the command

```
ykman otp --access-code <value> settings [OPTIONS] [1|2]
```

For instance, now it is not possible to 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 `--access-code` is not 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

13.2.3 Static Password

Feature Summary

The static password application allows for storing a complete or partial static password. The password is 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, Understanding Core Static Password Features.

CSPN Approved Mode

To operate the YubiKey 5 in a CSPN approved mode, store only one portion of the password within the YubiKey 5 and keep the remaining portion of the password in a different, but also secure location. Reconstruct the complete password by combining the portion from the YubiKey with the other portion stored elsewhere. Then authenticate in conjunction with the username. The overall details for such a password splitting scheme are beyond the scope of this document. 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, set an access code to protect both the static password and configuration. More details for such a configuration are described in the following section.

Technical Configuration

To protect the static password, use the command line YubiKey Manager (ykman).

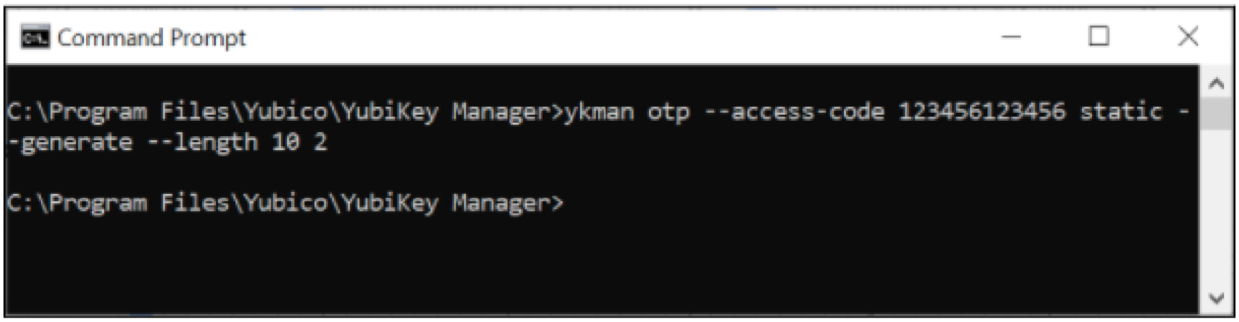
Use the command

```
ykman otp  --access-code <value> static --generate --length <value> <slot>
```

where -

`--access-code` for protecting the static password. Set the `--access-code` parameter to a six byte long hex value.

See the figure below for an example command line interaction for creating a protected static password with ykman.



```
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.

To **delete static password**, use the command:

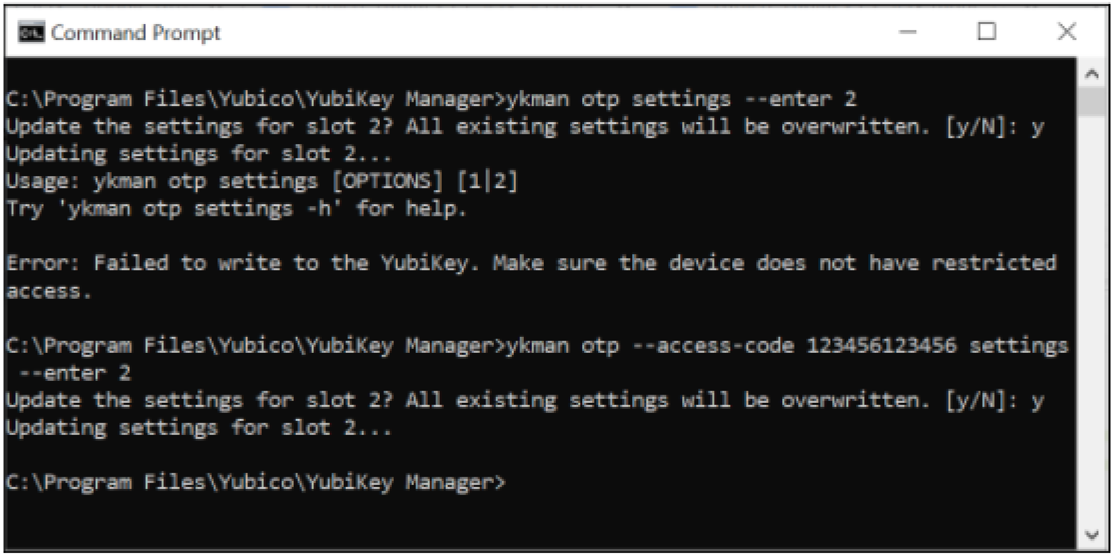
`ykman otp --access-code <value> delete [1|2]`

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

`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.



```
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

13.2.4 OATH-HOTP

Feature Summary

The OATH-HOTP protocol is implemented according to RFC 4226, “HOTP: An HMAC-Based One-Time Password Algorithm”, [RD5]. The algorithm underpinning this application on the YubiKey 5 is HMAC-SHA-1. 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, OATH.

CSPN Approved Mode

To operate the YubiKey 5 in a CSPN approved mode, identify the user by a first factor authentication scheme. For example, username/password. The details for such an authentication scheme are beyond the scope of this document.

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

Technical Configuration

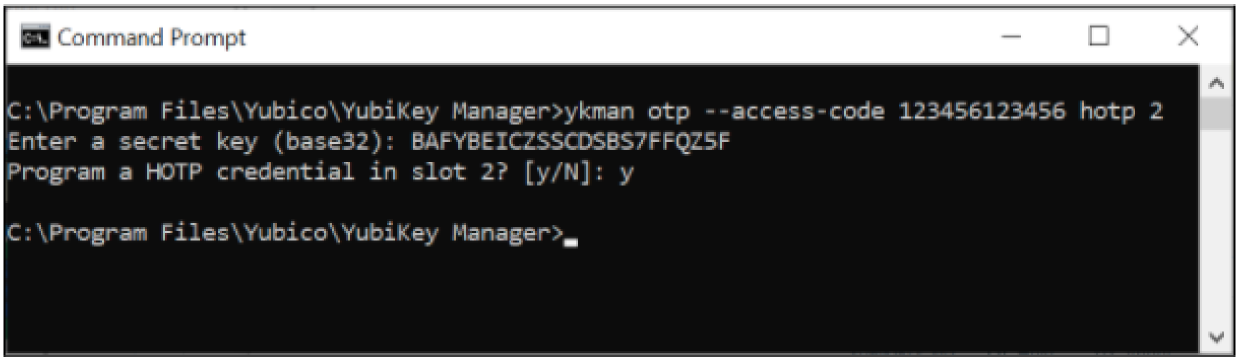
In order to protect the OATH-HOTP credentials, use the command line YubiKey Manager (ykman).

For protecting the OATH-HOTP credentials, use the command:

`ykman otp --access-code <value> hotp <slot>`

where `--access-code` is 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 following screenshot.



```
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.

To **Delete credentials**, use the command

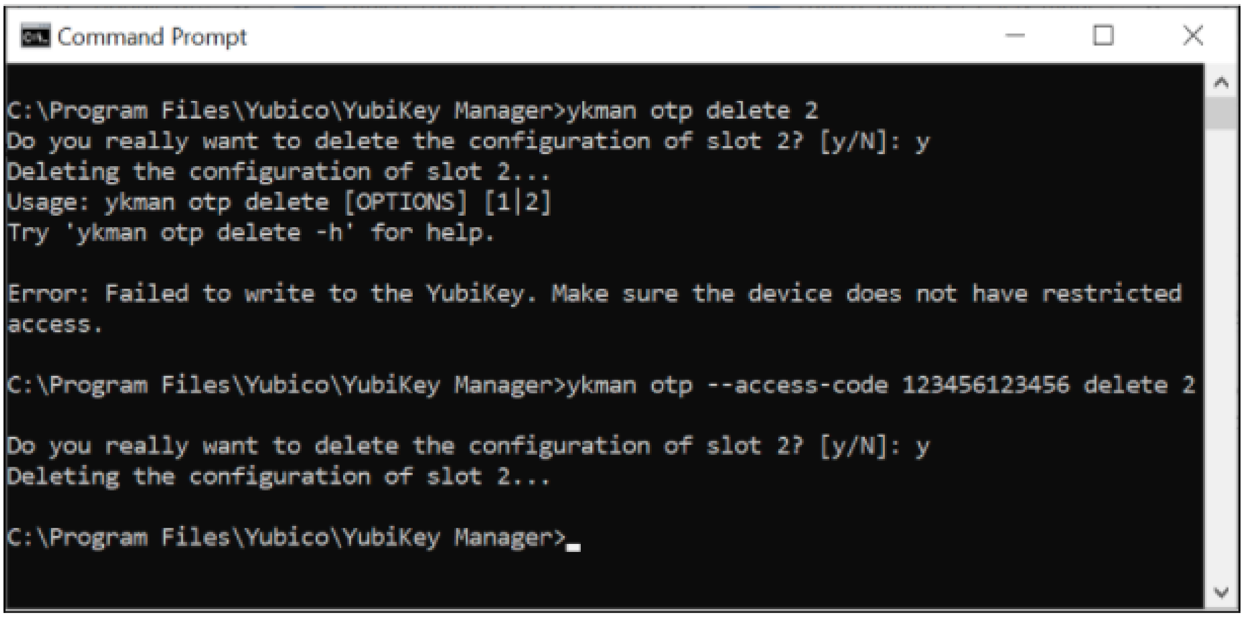
`ykman otp --access-code <value> delete [1|2]`

To **Change the settings**, use the command

`ykman otp --access-code <value> settings [OPTIONS] [1|2]`

For instance, you can no longer 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.



```
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

13.3 OATH

13.3.1 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 credentials[1] can be stored within the YubiKey's OATH application. Use the software tool Yubico Authenticator to configure and use this application.

A password may be set to protect the OATH credentials. If a password is configured, the password is required to unlock the application, which can then be used to generate any number of OTPs for the remainder of the session (until application is deselected).

When enrolling credentials, you can configure whether touching the sensor of the YubiKey 5 is required for each OTP generation.

13.3.2 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, identify the user by a first factor authentication scheme. For example, username/password. The details for such a first factor authentication scheme are beyond the scope of this document.

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

13.3.3 Technical Configuration

To protect the OATH-HOTP/TOTP credentials with a password, install and use the Yubico Authenticator for the configuration.

To set the password, launch the Yubico Authenticator application, select File from the menu, then select the option Set Password. In the dialog box that appears, enter a new password and confirm it. This configuration protects all OATH-HOTP/TOTP credentials with the same nominated password.

When Yubico Authenticator is used for generating an OATH one-time password, the password must be entered each time to unlock the credentials.

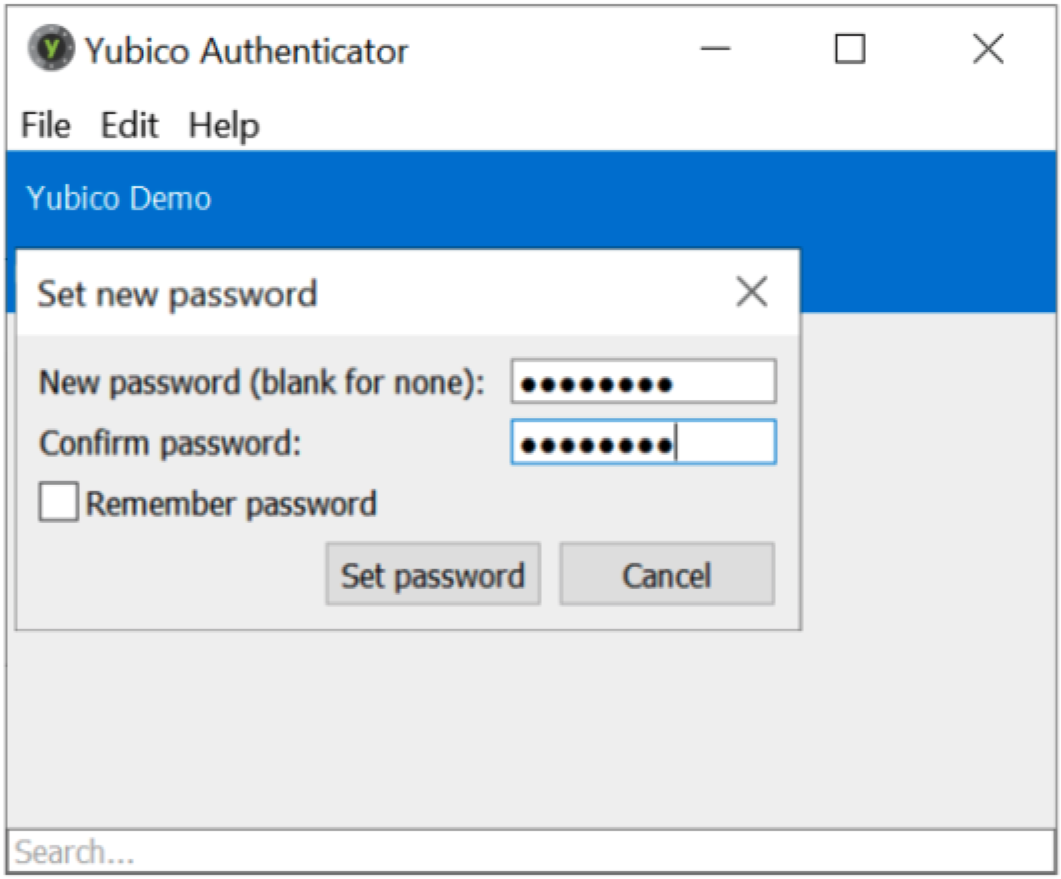
13.4 FIDO U2F

13.4.1 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, U2F.

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

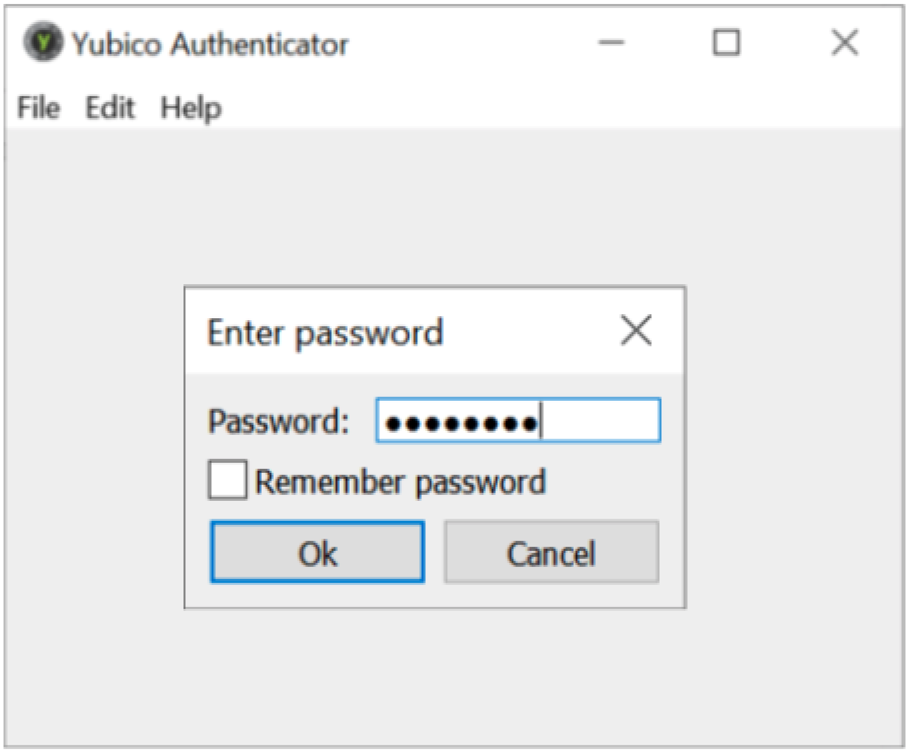


New password (blank for none): ●●●●●●●●●

Confirm password: ●●●●●●●●●

□ Remember password

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



Password: ●●●●●●●●●

□ Remember password

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

13.4.2 CSPN Approved Mode

To operate the YubiKey 5 in a CSPN approved mode, identify the user by a first factor authentication scheme (for example, username/password) according to the FIDO U2F standard [RD7]. The details for such a first factor authentication scheme are beyond the scope of this document.

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

13.4.3 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.

13.5 FIDO2

13.5.1 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.

13.5.2 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*)

13.5.3 FIDO2 With PIN Code

If WebAuthn User Verification is set to `Required` by the WebAuthn relying party, when registering the YubiKey 5 as a FIDO2 device, it prompts the your client to protect the FIDO2 credentials with a PIN code during the enrollment. Alternatively, you can also use the YubiKey Manager download to set a PIN code to protect the FIDO2 credentials. In both cases, the YubiKey 5 requires the PIN code for FIDO2 authentication.

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

13.5.4 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, identify the user by a first factor authentication scheme. For example, username/password.The details for such a first factor authentication scheme are beyond the scope of this document.

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

13.5.5 Technical Configuration

FIDO2 With PIN Code

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

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

By default, the touch sensor is also required for FIDO2, in addition to setting the PIN on the YubiKey.

Set FIDO2 PIN Code with ykman

Use the YubiKey Manager download to set a PIN code for the FIDO2 credentials on the YubiKey 5. When a PIN code is set, all FIDO2 credentials are protected by the same PIN code.

To set the PIN code with YubiKey Manager download:

1. Select the Applications from the menu.
2. Select the FIDO2 option.
3. In the GUI that appears, select the button **Set PIN**.

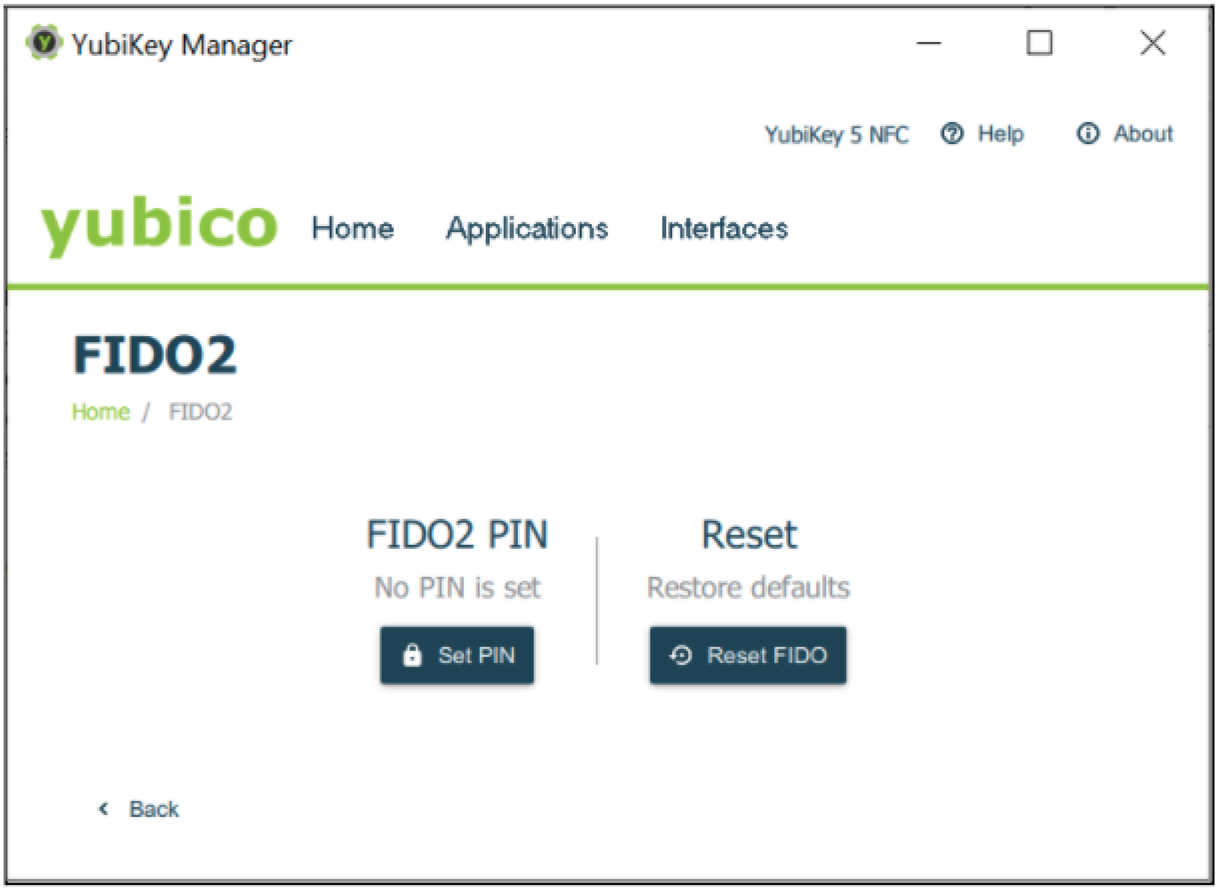
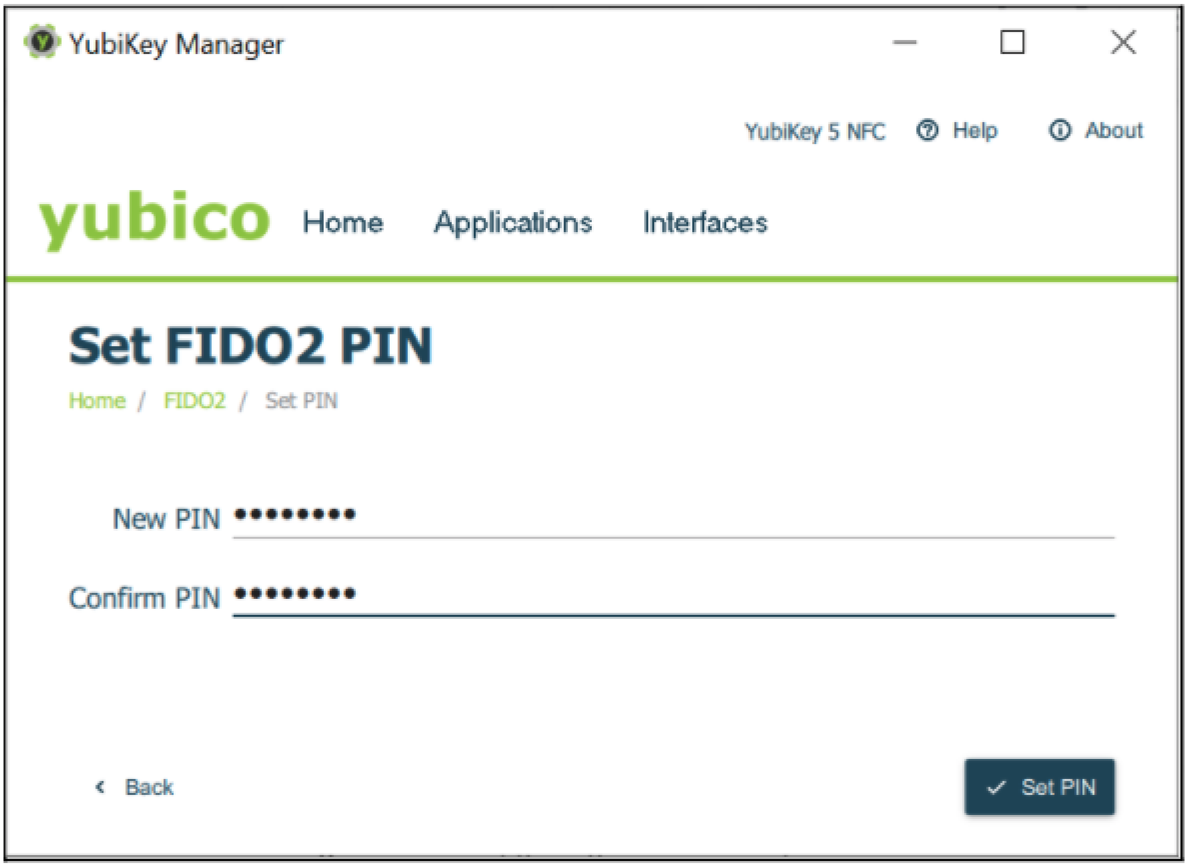


Fig. 12: **Figure 12** - Configuring the FIDO2 PIN with YubiKey Manager

4. In the next popup prompt that appears, set the new PIN and confirm this PIN for the FIDO2 application.



Set FIDO2 PIN

Home / FIDO2 / Set PIN

New PIN •••••••••

Confirm PIN •••••••••

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

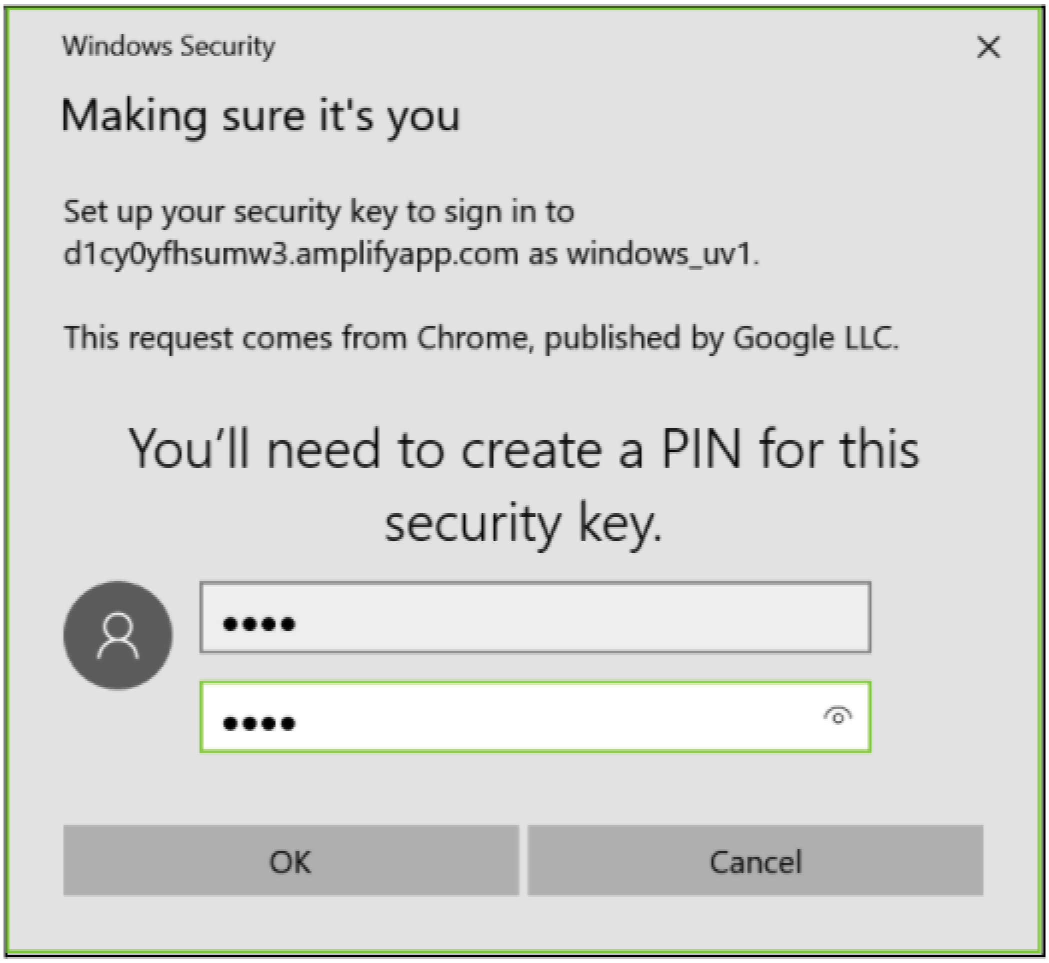
Set FIDO2 PIN Code From the Relying Party

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 prompts 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.



Windows Security

Making sure it's you

Set up your security key to sign in to d1cy0yfhsumw3.amplifyapp.com as windows_uv1.

This request comes from Chrome, published by Google LLC.

You'll need to create a PIN for this security key.

●●●●

●●●●

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 does not trigger the client to set any FIDO2 PIN code on the YubiKey 5.

If a FIDO2 PIN is not set through YubiKey Manager and the client is not triggered to set a FIDO2 PIN, this means the FIDO credentials are not protected by a PIN.

However, by default, touch is still required 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.

13.6 PIV

13.6.1 Feature Summary

The PIV application [RD4] can be used to authenticate, sign and decrypt. You can, 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 storing 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). This is used for various oversight functions. Set the PIN to a value between 6 and 8 bytes. Set the maximum number of retries between 1 to 255. The retries default value is 3.

To specify how often the PIN needs to be entered to access the credentials in a given slot, set a PIN policy for that slot. Set this policy when generating the key or when importing a key. The policy cannot be changed later.

You can configure the YubiKey 5 to require physical contact on the touch sensor, in addition to requiring the PIN. Similar to the PIN policy, set the touch policy when generating or importing a key.

13.6.2 CSPN Approved Mode

To operate the YubiKey's PIV application in approved CSPN mode, set the PIN code, PUK code, and management key for the PIV application. It is imperative that you change the default values of these codes before using the PIV application.

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

13.6.3 Technical Configuration

YubiKey Manager for PIN Configuration of PIV

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

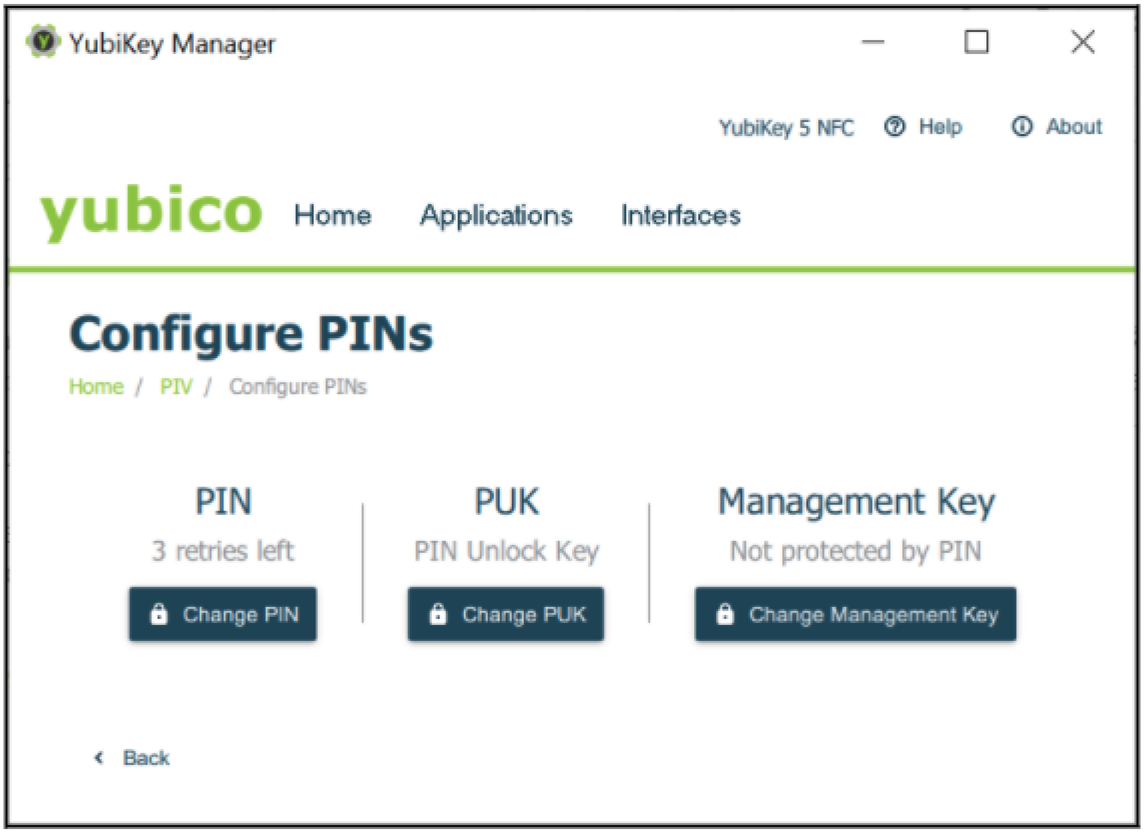


Fig. 15: **Figure 15** - Configuring the PIN, PUK, and management key for PIV

Changing the PIN Code

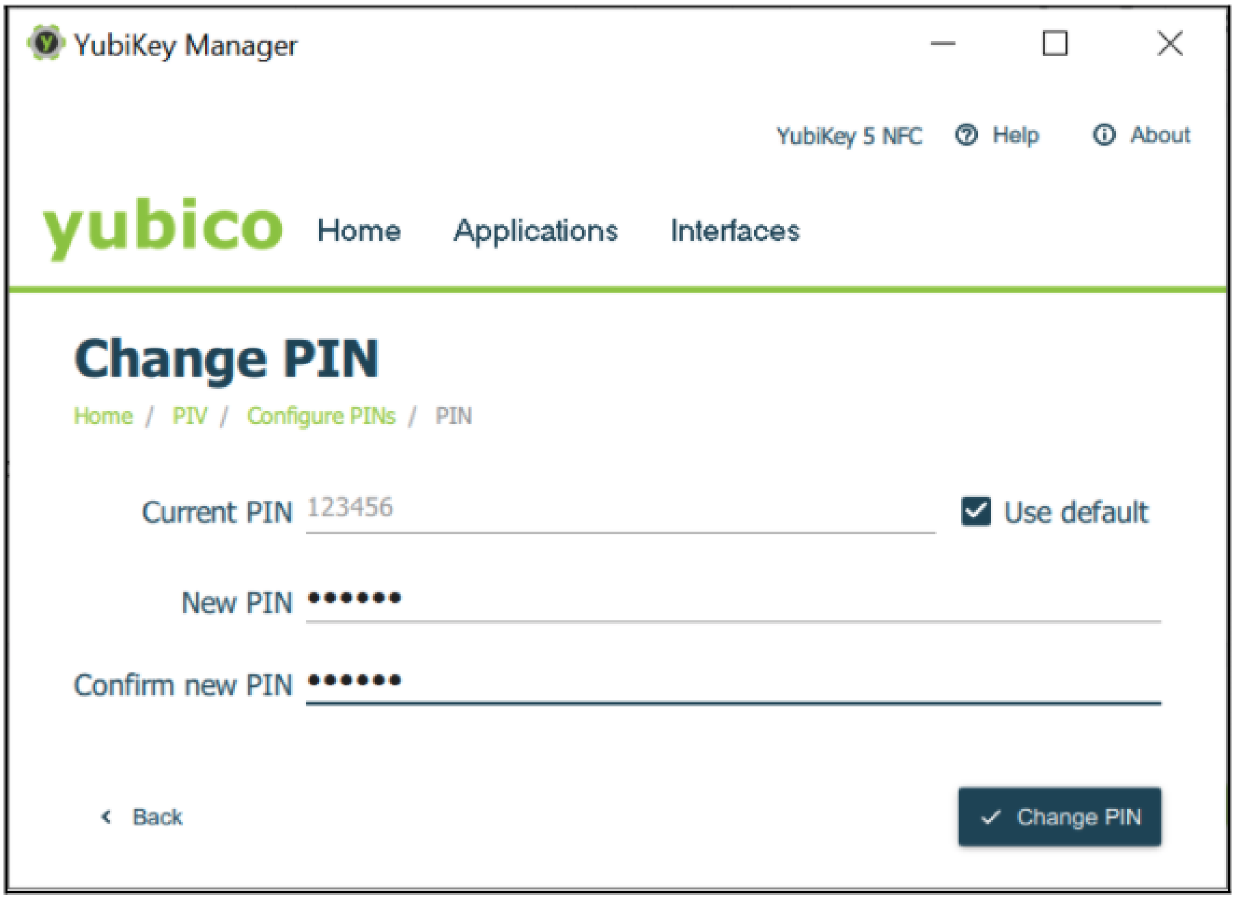
The PIN is used during normal operation to authorize an action such as creating a digital signature with any of the stored keys. Entering an incorrect PIN too many times and exceeding the retry counter, blocks the PIN and makes the PIV features unusable.

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. This allows for up to 8 ASCII characters. By default the PIN code is set to `123456`.

To change the PIN code, select the **Change PIN** button in the Configure PINs dialog box. The popup that appears in the YubiKey Manager, is shown in the following figure.

Change the current or default PIN to a new PIN with a length of 6-8 digits. To do this, enter the current PIN, enter a new PIN, confirm it, and then select the **Change PIN** button.

The default PIN code, `123456`, is pre-configured for slots 9a, 9c and 9d. For slot 9e, set to enforce the PIN policy using the command line YubiKey Manager when generating or importing the key-pair on the YubiKey 5. Use the command:



Change PIN

Home / PIV / Configure PINs / PIN

Current PIN 123456 ☑ Use default

New PIN ••••••

Confirm new PIN ••••••

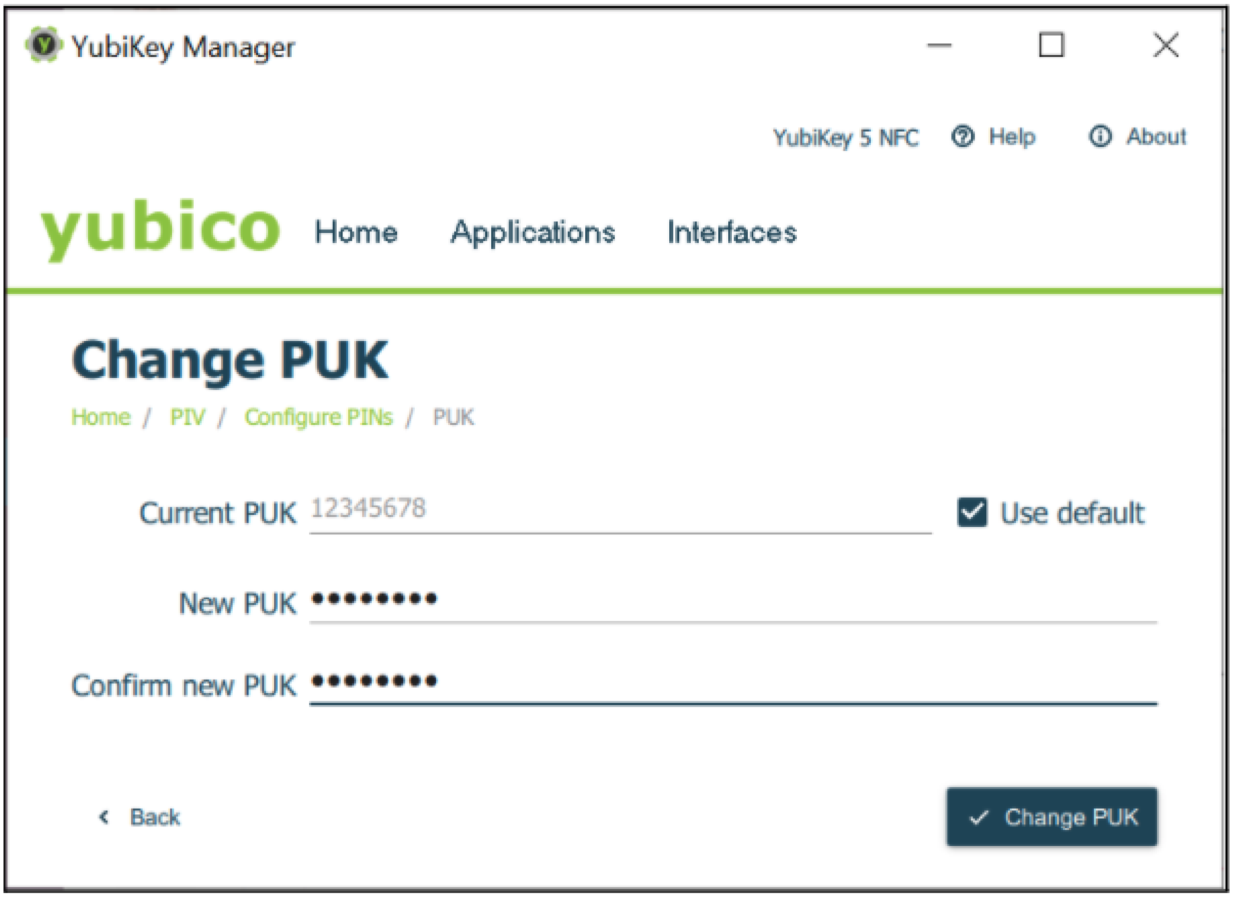
Fig. 16: **Figure 16** - Changing the PIN code for PIV

```
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 PIN entry attempts. By default the PUK is set to 12345678.

To change the PUK, select the **Change PUK** button in the Configure PINs dialog box. The popup that appears in the YubiKey Manager, is shown in the following figure.



Current PUK 12345678 ☑ Use default

New PUK •••••••••

Confirm new PUK •••••••••

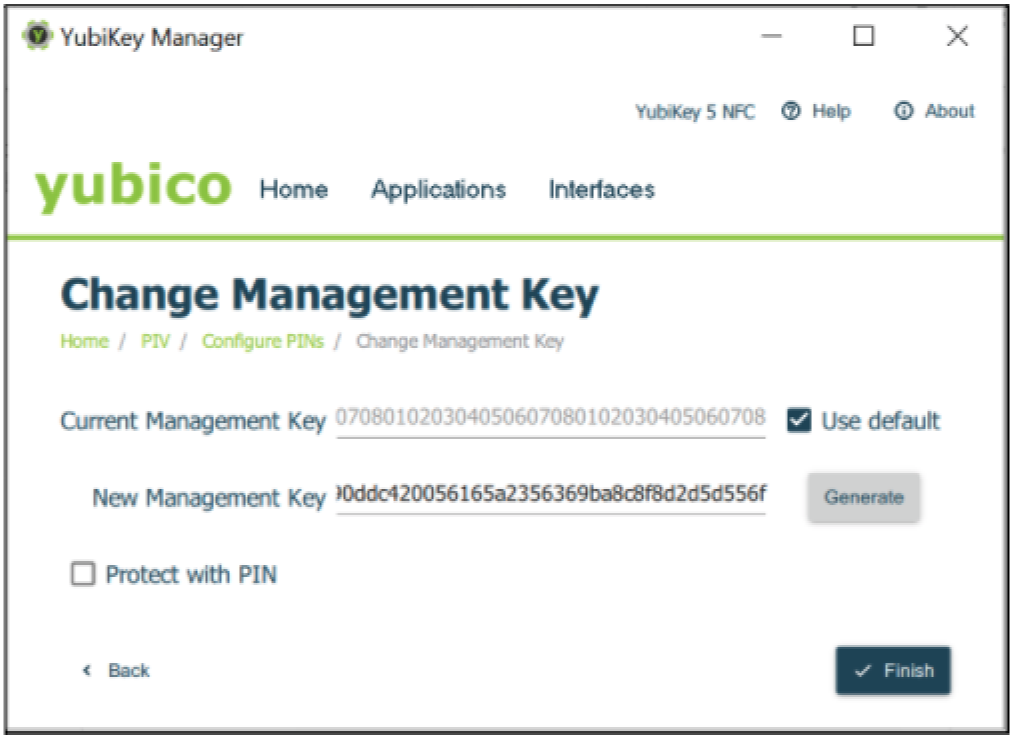
Fig. 17: **Figure 17** - Changing the PUK code for PIV

Change the default or current PUK to a new PUK with a length of 6-8 digits. To do this, enter the current PUK, the new PUK, confirm it, and then select 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`. Explicitly set a 24 byte key or use the YubiKey PIV Manager to generate one.

To change the management key press the **Change Management Key** button in the Configure PINs dialog box. The popup that appears in the YubiKey Manager, is shown in the following figure.



Change Management Key

Home / PIV / Configure PINs / Change Management Key

Current Management Key 070801020304050607080102030405060708 ☑ Use default

New Management Key)0ddc420056165a2356369ba8c8f8d2d5d556f Generate

□ Protect with PIN

‹ Back ✓ Finish

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

Change the current or default management key to a new management key with a length of 48 hexadecimal digits. To do this, in the Change Management Key dialog, enter the current management key, the new management key, and select the **Finish** button.

Click for Yubico Support.

CHAPTER

FOURTEEN

YUBIKEY BIO SERIES SPECIFICS

14.1 How the YubiKey Bio Works

For the full technical explanation of this from a developer perspective over FIDO, start with the Yubico's WebAuthn Developer Guide.

Note: In the following, the term *credentials* is referenced repeatedly. There are different kinds of credentials. To pursue all the distinctions, consult the FIDO2 page on the Fido Alliance web site.

14.1.1 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 you are authenticating to a WebAuthn/FIDO2 site, because the browser/client app *can* prompt for the PIN. Otherwise you 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 *Fingerprint 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.

14.1.2 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.

When using the YubiKey Bio - Multi-protocol Edition the client needs to support the PIV protocol and the client needs to have the Yubico Minidriver installed (version 4.6.0 or later).

Registration

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

Authentication

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

14.1.3 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 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 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, use discoverable credentials. 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 correct 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 are 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 needs to unblock the Yubikey. To do this, see the YubiKey Bio start page or by use the Yubico Authenticator for Desktop.

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 becomes **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, reset the FIDO2 application. For more details, see *FIDO2 PIN*.

Unblock
Unblock the YubiKey Bio's biometric function (its ability to read fingerprints) refer to the unblocking FAQ on the YubiKey Bio start page. Otherwise you can use any of the other methods given in *Troubleshooting and Tools*.

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*.

14.1.4 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 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 more **developer-oriented** information on this, see Discoverable Credentials / Resident Keys on Yubico's developer site.

14.2 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 - that is, software authenticators that in this case are also platform authenticators. The prompts in both Windows and Mac *might assume* you are 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 you are authenticating to a WebAuthn/FIDO2 site, because the browser/client app *can* prompt for the PIN. Otherwise you 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 *Fingerprint Tips*. For detailed information on FIDO2 PINs and their requirements, see Understanding YubiKey PINs.

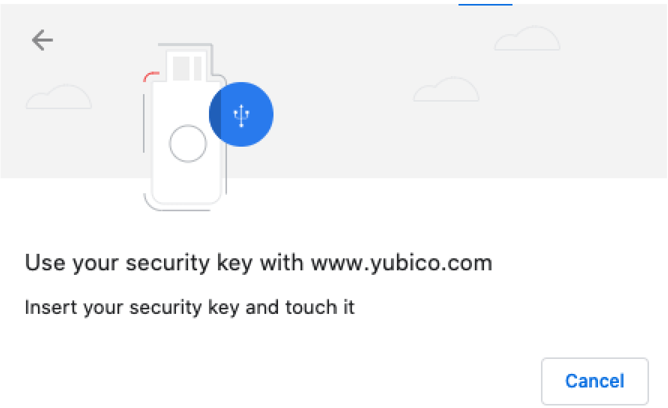
14.2.1 Enrolling the First Fingerprint

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, this wizard walk you through the PIN setup (if no PIN is set) and fingerprint enrollment:

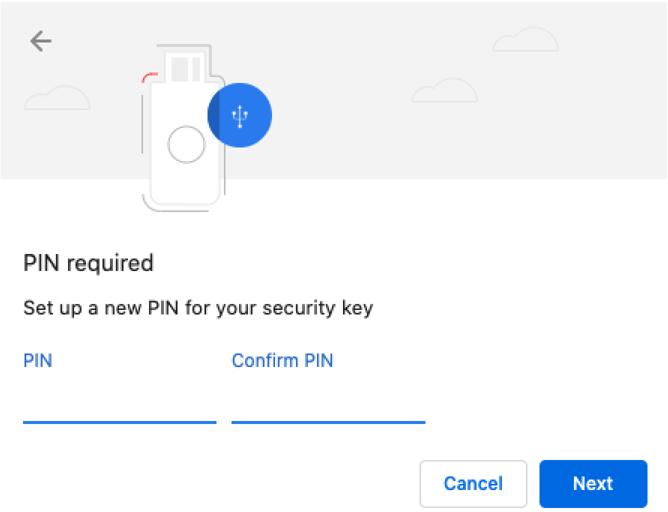


Use your security key with www.yubico.com

Insert your security key and touch it

Step 3

If the amber LED flashes slowly, it means either no fingerprint is enrolled or biometrics is blocked. If you have reason to believe biometrics are blocked, go to the appropriate link on the YubiKey Bio Series setup page or to *Troubleshooting and Tools*. Otherwise, *touch the key*:



PIN required

Set up a new PIN for your security key

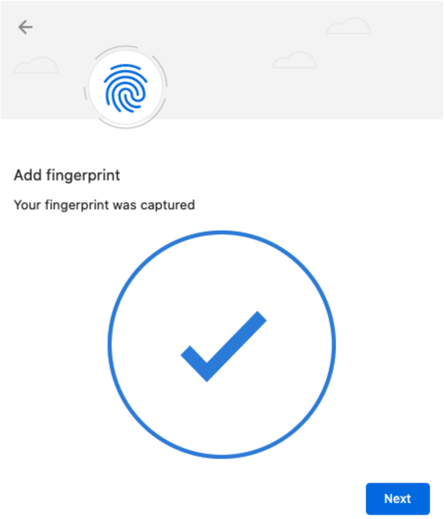
Step 4

If no PIN is set, set one by entering at least 4 digits, then confirm 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."

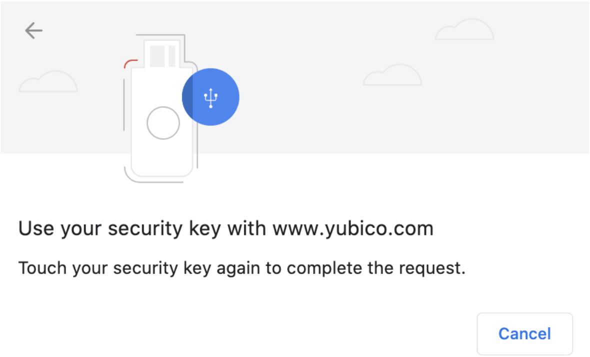


Add fingerprint

Your fingerprint was captured

Step 7

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



Use your security key with www.yubico.com

Touch your security key again to complete the request.

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.

14.2.2 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 and navigate to **Settings->Security->Advanced->Manage security keys**.

Step 2
Click **Fingerprints** and follow the instructions in the popup.

14.3 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 - that is, software authenticators that in this case are also platform authenticators. The prompts in both Windows and Mac *might assume* you are 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 you are authenticating to a WebAuthn/FIDO2 site, because the browser/client app *can* prompt for the PIN. Otherwise you must unblock biometrics by using either:

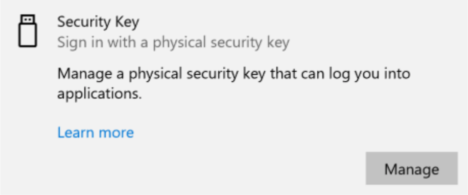
- The YubiKey Bio start page
- Yubico Authenticator for Desktop.

For information on the YubiKey Bio's sensor and tips on working with fingerprints see *Fingerprint 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**.


Security Key
Sign in with a physical security key

Manage a physical security key that can log you into applications.

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.

14.4 Fingerprint Tips

14.4.1 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.

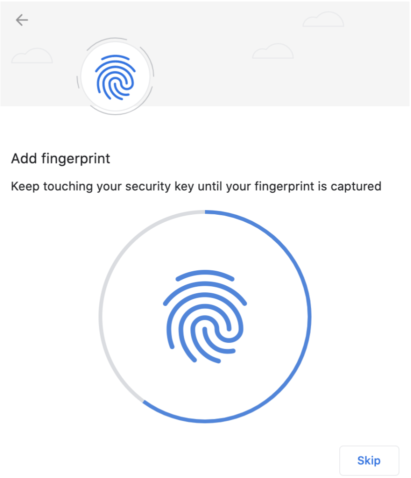
- If the key reacts to your touch by flashing or blinking the green LED, you used the right touch.
- If the amber LED flashes three times in quick succession, the attempt to match your fingerprint with the template was not successful.
- If the amber LED flashes slowly and continuously, it is in the biometrics 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.

14.4.2 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:

14.4.3 Fingerprint Orientation

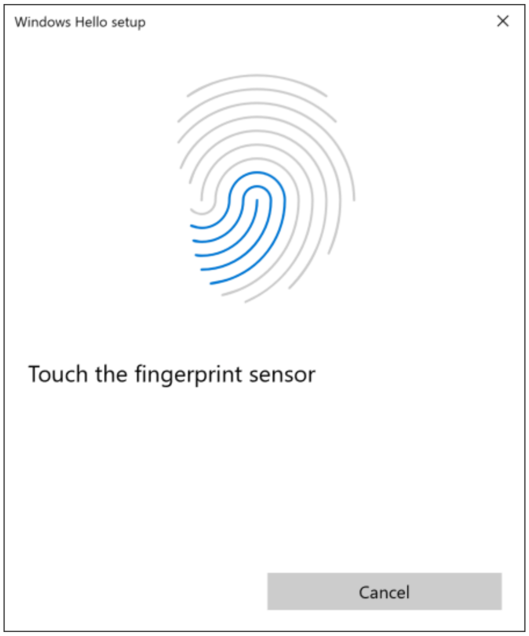
The YubiKey Bio supports 360 degree fingerprint reading, meaning that a fingerprint can be read from any angle once successfully enrolled.



Add fingerprint

Keep touching your security key until your fingerprint is captured

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



Windows Hello setup

Touch the fingerprint sensor

Fig. 2: Windows: Capturing the Fingerprint

14.4.4 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. Touch types:

Biometric
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 does not read the fingerprint.

Plain
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 is not 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.

Print quality
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.

Repeat reading
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 needs you to repeat enrolling.

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.

Stable 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.

Stable 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 enrolling a fingerprint, the green LED on your YubiKey Bio starts to flash. Start enrolling the fingerprint 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 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.

14.5 Troubleshooting and Tools

14.5.1 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 becomes **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, reset the FIDO2 application. For more details, see *FIDO2 PIN*.

Unblock
Unblock the YubiKey Bio's biometric function (its ability to read fingerprints) refer to the unblocking FAQ on the YubiKey Bio start page. Otherwise you can use any of the other methods given in *Troubleshooting and Tools*.

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.

14.5.2 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.

14.6 Requirements: Platform and Browser Compatibility

14.6.1 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 you when the YubiKey has failed to match the fingerprint, so you must watch for the YubiKey's blinking amber LED to indicate if 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 you are typically prompted to enter the PIN even if the key is not in the "biometrics blocked" state.

14.6.2 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, you might need to resort to computers (and not the mobile devices) to unblock biometrics.

14.7 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 *Troubleshooting and Tools*.

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, which has neither fingerprints nor PIN nor credentials, you must enroll your fingerprints again after resetting it and register your key again to your apps and services. See the relevant Enrolling chapter, either *Using Chrome to Enroll Fingerprints* or *Using Windows to Enroll Fingerprints*.

Note: 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 *Troubleshooting and Tools*.

14.8 Frequently Asked Questions

See the FAQs on the YubiKey Bio Start Page.

14.9 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 are used on a smartphone. 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. The only opportunity to input the PIN is after 3 unsuccessful attempts at matching a fingerprint with an enrolled finger.

14.9.1 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 unless you reset the FIDO2 application.
- If the FIDO2 PIN is entered incorrectly 3 times in a row, the key needs to be reinserted before it can 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 becomes locked, and FIDO2 authentication is not possible.

- To restore the FIDO2 functionality, reset the FIDO2 application.

  Note: Resetting the FIDO2 application also resets 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.

14.9.2 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)*.

14.9.3 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 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 more **developer-oriented** information on this, see Discoverable Credentials / Resident Keys on Yubico's developer site.

14.9.4 Supported Extensions

The YubiKey Bio supports 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 continue to work.

Note: **Developers**: For AAGUID values, see YubiKey Hardware FIDO2 AAGUIDs.

14.10 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.

14.10.1 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 (that is, authentication). See *Troubleshooting and Tools* for full instructions and more information.

Note: **Developers**: With regard to computer login tools that use FIDO U2F for second-factor authentication, some software might 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

14.10.2 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*.

14.10.3 Supported Extensions

The YubiKey Bio supports 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 continue to work.

Note: **Developers**: For AAGUID values, see YubiKey Hardware FIDO2 AAGUIDs.

14.11 YubiKey Bio and PIV

The YubiKey Bio - Multi-protocol Edition supports PIV in addition to FIDO U2F and FIDO2. The PIV application works as the PIV application on a YubiKey 5 Series, see *Smart Card (PIV Compatible)*, with some differences outlined below.

14.11.1 Device PIN

On the YubiKey Bio Multi-protocol Edition the PIN is shared between the PIV and FIDO2 applet. The PIN on a YubiKey Bio Multi-protocol Edition has to be between 6 and 8 characters.

Click for Yubico Support.

CHAPTER

FIFTEEN

ACRONYMS

2FA
Two-Factor Authentication

3DES
Triple Data Encryption Algorithm

AES
Advanced Encryption Standard

BSI
Bundesamt für Sicherheit in der Informationstechnik

CC
Common Criteria

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

CSPN
Certificat de Sécurité de Premier Niveau

CSR
Certificate Signing Request

CTAP2
Client to Authenticator Protocol v2

DES
Data Encryption Standard

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

NIST
National Institute of Standards and Technology

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

PIN
Personal Identification Number

PIV
Personal Identity Verification

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 Unblocking Key

RFC
Request For Comments

SHA
Secure Hash Algorithm

stdin
standard input - usually keyboard or CLI instructions

stdout
standard output - usually print to screen

TOTP
Time-based One-Time Password algorithm

U2F
Universal Second Factor

W3C
World Wide Web Consortium

X.509
The standard defining the format of a public key certificate

Click for Yubico Support.

CHAPTER

SIXTEEN

COPYRIGHT

16.1 Trademarks

Yubico and YubiKey are registered trademarks of Yubico AB. All other trademarks are the property of their respective owners.

16.2 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.

16.3 Contact Information

Yubico AB
Kungsgatan 44
111 35 Stockholm
Sweden

16.4 Getting Help

Documentation is continuously updated on https://docs.yubico.com/ (this site). Additional support resources are available in the Yubico Knowledge Base.

Click the links to:

- Submit a support request
- Contact our sales team

16.5 Feedback

Yubico values and welcomes your feedback. If you think you may have discovered a flaw in our product, please submit a support request at https://support.yubico.com/hc/en-us and provide as much detail as you can.

16.6 Document Updated

2024-08-09 17:37:57 UTC