YubiKey Technical Manual

Yubico

Nov 02, 2026

CONTENTS

1 Introduction to the YubiKey Series **1**
- 1.1 YubiKey 5 Series . . . 3
- 1.2 YubiKey 5 FIPS Series . . . 4
- 1.3 YubiKey 5 CCN Series . . . 5
- 1.4 YubiKey 5 Enhanced PIN Series . . . 5
- 1.5 YubiKey Bio Series . . . 6
- 1.6 Security Key Series . . . 7
- 1.7 YubiKey 5 CSPN Series . . . 8
- 1.8 Firmware Certifications . . . 8

2 Physical Attributes **11**
- 2.1 YubiKey NFC Form Factor . . . 11
- 2.2 YubiKey C NFC Form Factor . . . 12
- 2.3 YubiKey Nano Form Factor . . . 12
- 2.4 YubiKey 5C Form Factor . . . 13
- 2.5 YubiKey 5C Nano Form Factor . . . 13
- 2.6 YubiKey 5Ci Form Factor . . . 14
- 2.7 YubiKey 5 FIPS Series . . . 14
- 2.8 YubiKey 5 Enhanced PIN Series . . . 15
- 2.9 YubiKey 5 CCN Series . . . 16
- 2.10 YubiKey Bio Series . . . 17
- 2.11 Security Key Series . . . 18
- 2.12 YubiKey Connection Interfaces . . . 19
- 2.13 YubiKey ID Numbers . . . 23
- 2.14 Verify the YubiKey and Packaging . . . 23
- 2.15 Care and Cleaning . . . 23

3 Protocols and Applications **25**
- 3.1 FIDO Specifics . . . 25
- 3.2 Smart Card - PIV Compatible Specifics . . . 44
- 3.3 OATH Specifics . . . 52
- 3.4 OpenPGP Specifics . . . 53
- 3.5 OTP Specifics . . . 54
- 3.6 Secure Channel Protocol (SCP03 and SCP11) Specifics . . . 55
- 3.7 YubiHSM Auth Specifics . . . 70

4 YubiKey Firmware **75**
- 4.1 Firmware Overview . . . 75
- 4.2 5.8 Firmware Specifics . . . 85
- 4.3 5.7 - 5.6 Firmware Specifics . . . 97

4.4 Pre-5.6 Firmware Specifics . . . 106

5 YubiKey 5 FIPS Series Specifics 109
5.1 YubiKey 5 FIPS Series Features . . . 109
5.2 Deploying the YubiKey 5 FIPS Series . . . 112
5.3 YubiKey Firmware 5.4 Changes for FIPS . . . 113
5.4 YubiKey Firmware 5.7.x Changes for FIPS 140-3 . . . 114
5.5 FIPS Version Changes and Requirements Comparison . . . 116
5.6 FIPS 140-3 Configuration . . . 120
5.7 FIPS 140-2 Level 2 Configuration . . . 125
5.8 FIPS 140-2 Level 1 Configuration . . . 126
5.9 FIDO Configuration with FIPS . . . 126
5.10 OATH Configuration with FIPS . . . 128
5.11 OTP Configuration with FIPS . . . 128
5.12 PIV Configuration with FIPS . . . 130

6 YubiKey Bio Series Specifics 133
6.1 Using the YubiKey Bio Series . . . 133
6.2 Physical Attributes . . . 134
6.3 Protocols Supported . . . 134
6.4 Requirements: Platform and Browser Compatibility . . . 135
6.5 YubiKey Bio and FIDO2 . . . 135
6.6 YubiKey Bio and FIDO U2F . . . 138
6.7 YubiKey Bio and PIV . . . 139
6.8 How the YubiKey Bio Works . . . 139
6.9 User Experiences . . . 141
6.10 Understanding YubiKey Bio Enrollment . . . 142
6.11 Enrolling Fingerprints on Chrome . . . 143
6.12 Enrolling Fingerprints on Windows . . . 144
6.13 Unblocking and Resetting the YubiKey Bio . . . 145
6.14 Fingerprint Tips . . . 147
6.15 Troubleshooting and Tools . . . 150

7 NFC ID Calculation Technical Description 153
7.1 YubiKey for Door Access . . . 153
7.2 NFC ID Calculation for YubiKey v5.2.x and Earlier . . . 153
7.3 NFC ID Calculation for YubiKey v5.3.0 and Later . . . 153

8 Tools and Troubleshooting 155
8.1 Managing Applications . . . 155
8.2 Yubico Authenticator . . . 155
8.3 YubiKey Manager GUI / ykman CLI . . . 156
8.4 YubiKey Minidriver . . . 156
8.5 YubiKey Verification Site and FIDO Application Demo Site . . . 157
8.6 Troubleshooting . . . 157

9 YubiKey 5 CSPN Series Specifics 159
9.1 CSPN Specific Marking . . . 159
9.2 YubiKey 5 CSPN Series Firmware Version . . . 159
9.3 CSPN Mode Configuration . . . 160
9.4 One-Time Password - OTP . . . 162
9.5 OATH . . . 169
9.6 FIDO U2F . . . 171
9.7 FIDO2 . . . 172
9.8 PIV . . . 176

9.9 References 180

10 Acronyms **183**

11 Copyright **185**
11.1 Trademarks 185
11.2 Disclaimer 185
11.3 Contact Information 185
11.4 Getting Help 186
11.5 Feedback 186
11.6 Document Updated 186

CHAPTER

ONE

INTRODUCTION TO THE YUBIKEY SERIES

This guide is intended for technical users implementing YubiKey 5 Series passkeys to secure access and actions across their environments for their end users and organizations. See Yubico Solutions Initatives, Use Cases, and Technology for areas of integration.

YubiKey 5 FIPS Series 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 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.

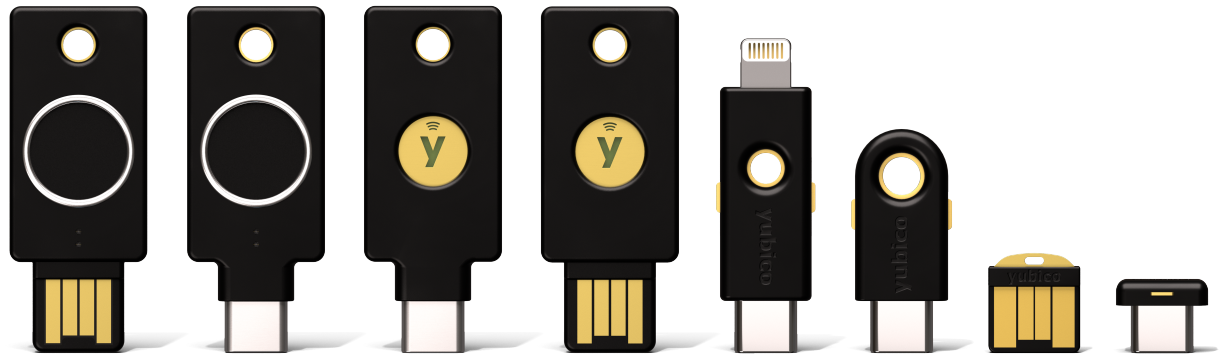
Where applicable, throughout this guide, the YubiKey 5 Series, the YubiKey 5 FIPS Series, the YubiKey 5 CCN Series, and the YubiKey 5 Enhanced PIN Series are referred to collectively as the **YubiKey 5 Series**, because all these YubiKeys share the same hardware base and many firmware features.

Also described in this guide are the YubiKey Bio Series and the Yubico Security Key Series.

This topic introduces:

- *YubiKey 5 Series*
- *YubiKey 5 FIPS Series*
- *YubiKey 5 CCN Series*
- *YubiKey 5 Enhanced PIN Series*
- *YubiKey Bio Series*
- *Security Key Series*
- *Firmware Certifications*

For a convenient way of comparing YubiKeys in 5 Series, Bio Series, and Security Keys Series, see Yubico's YubiKey Comparison Chart.



YubiKey Series Form Factors

The following table lists for each YubiKey Series: supported form factors (in order matching the image) and minimum firmware version where the YubiKey series first became available on the listed form factor.

Table 1: **YubiKey Series, Form Factors, and Minimum Firmware Versions**

YubiKey Series	5 Bio	5C Bio	5C NFC	5 NFC	5Ci	5C	5 Nano	5C Nano
Standard			5.2.x	5.0.x	5.3.x	5.0.x	5.0.x	5.0.x
FIPS			5.4.2	5.4.2	5.4.2	5.4.2	5.4.2	5.4.2
Enhanced PIN			5.7.4	5.7.4				
CCN			5.7.4	5.7.4				
Bio FIDO Edition	5.5.x	5.5.x						
Bio Multi-Protocol Ed.	5.5.x	5.5.x						
Security Key			5.4.x	5.0.x				
Security Key Enterprise Ed.			5.4.x	5.0.x				

* All form factors also have space for adding to a keychain, except 5 Nano and 5C Nano forms.

Where form factor codes indicate an interface type. See *YubiKey Connection Interfaces*.

- NFC - Near Field Communication: 5 NFC, 5C NFC
- C - USB-C connector: 5C Bio, 5C NFC, 5Ci, 5C, 5C Nano

- Ci - Apple lightning connector: 5Ci
- Bio - Finger print sensor: 5 Bio, 5C Bio
- No extra code - USB-A connector: 5 Bio, 5 NFC, 5 Nano

For a quick start, to register your YubiKey with supported applications and services, see Set up your YubiKey

1.1 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.

Protocols and Applications lists the YubiKey 5 Series functionalities and capabilities by protocol:

- *FIDO Specifics*
- *OATH Specifics*
- *OpenPGP Specifics*
- *OTP Specifics*
- *Smart Card - PIV Compatible Specifics*
- *Secure Channel Protocol (SCP03 and SCP11) Specifics*
- *YubiHSM Auth Specifics*

Note: It is the firmware version that determines which of the more specialized functionalities and capabilities are available on your YubiKey. See *Firmware Certifications*.

Yubico Authenticator is one of the tools most commonly used to configure YubiKeys, so for a complete breakdown of Yubico Authenticator functionality by platform and connection type for each YubiKey model, see the Yubico Authenticator Functionality table.

The Yubico `ykman` CLI (command line interface) is another cross-platform tool for advanced management and configuration of all YubiKey applications (FIDO2, FIDO U2F, PIV, Yubico OTP, YubiHSM Auth, OpenPGP, OATH, Security Domain). See the YubiKey Manager (ykman) CLI Guide.

For a list of YubiKey 5 Series specifications, see the YubiKey 5 Series Product Brief.

1.2 YubiKey 5 FIPS Series

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.

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 140-3 and FIPS 140-2. For the U.S. government, the default is that FIPS is **required**.

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. FIPS 140-3 validation enables government agencies and regulated industries to meet the highest authenticator assurance level 3 (AAL3) requirements from the NIST SP800-63B guidance .

Note:

- Effective May 22 2026, YubiKey 5 FIPS Series with firmware 5.7.4 was NIST certified for FIPS 140-3.
- Effective May 2026, YubiKey 5 FIPS Series with firmware 5.4.x and NIST FIPS 140-2 certification moved to Sunset List.
- Effective September 22, 2026, All NIST FIPS 140-2 certification will be moved to the Historical List.

Table 2: **Form Factors for YubiKey 5 FIPS Series**

Series	5 NFC	5C NFC	5 Nano	5C Nano	5Ci
YubiKey FIPS Series	USB-A + NFC	USB-C + NFC	Nano USB-A	Nano USB-C	Lightning + USB-C

For form factor description, see *YubiKey 5 FIPS Series*.

For a list of YubiKey 5 FIPS Series specifications, see the YubiKey 5 FIPS Series Product Brief.

See also in this guide, *YubiKey 5 FIPS Series Features* and *NIST Certification: FIPS*.

1.2.1 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.

1.3 YubiKey 5 CCN Series

The *YubiKey 5 CCN Series* is a standard YubiKey 5 Series key running firmware version 5.7.4 that is certified at the National Security Framework (ENS) Alta (High) level by the National Cryptologic Centre (CCN). This certification enables Spanish public organizations to comply with Royal Decree/Real Decreto 311/2022.

Note: YubiKey 5.8 is currently undergoing final re-certification.

The YubiKey 5 CCN Series has been added to the CPSTIC catalog, *Catálogo de Productos y Servicios de Seguridad de las Tecnologías de la Información y la Comunicación (CPSTIC)*, Yubikey 5 CCN Series 5.7.4, in compliance with Guía de Seguridad de las TIC, CCN-STIC 105.

To meet the CCN requirements, deploy the YubiKey CCN Series in accordance with the following:

- CCN compliance is focused on strong cryptographic authentication methods, specifically FIDO2 and PIV. Other protocols are not covered by CCN requirements.
- FIDO2 - Use PINs to protect all discoverable credentials.
- PIV - Use either RSA 3072 bytes and above or ECC based private keys.
- PIV - Change PIN, PUK, and Management keys from default. Block PUK codes if they are not required.

Table 3: **Form Factors for YubiKey 5 CCN Series**

Series	5 NFC	5C NFC
YubiKey CCN Series	USB-A + NFC	USB-C + NFC

For form factor description, see *YubiKey 5 CCN Series*.

See also Yubico Support:

- YubiKey 5 CCN Series
- YubiKey hardware FIDO2 AAGUIDs

1.4 YubiKey 5 Enhanced PIN Series

The *YubiKey 5 Enhanced PIN Series* provides a standard YubiKey 5 Series key with fixed settings for 6 character minimum PIN length, PIN complexity, and always on verification.

Based on the YubiKey 5 Series, the Enhanced PIN Series features include:

- Uses the FIDO (FIDO2/FIDO U2F) protocol. See *FIDO2 Features*.
- Ships with firmware version 5.7.4 or later. See *Firmware Overview*.
- Has 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 (Human Interface Device) interface.
- Enables access to the serial number over CCID running firmware version 5.7.x or later. See *CCID with USB*.

  The serial number is 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.

- Enforces *PIN Complexity*. This cannot be changed.
- Increases in the minimum FIDO pin length to 6. See *Minimum PIN Length*.
- Enforces user verification always on, *Always Require User Verification*.
- YubiKey 5 Enhanced PIN Series requires a subscription to YubiKey as a Service. Contact Yubico sales team for information about YubiKey as a Service.

Table 4: **Form Factors for YubiKey 5 Enhanced PIN Series**

Series	5 NFC	5C NFC
YubiKey Enhanced PIN Series	USB-A + NFC	USB-C + NFC

For form factor description, see *YubiKey 5 Enhanced PIN Series*.

For a list of YubiKey 5 Enhanced PIN Series specifications, see the YubiKey 5 Series Product Brief.

For AAGUID, see YubiKey hardware FIDO2 AAGUIDs.

1.5 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 following is a summary of the features; for complete explanations, see *YubiKey Bio Series Specifics*.

New York State's Department of Financial Services (DFS) advises that instead of using a traditional fingerprint or other biometric authentication system, the services regulated by DFS should consider using an authentication factor that employs technology with liveness detection or texture analysis to verify that a print or other biometric factor comes from a live person. This is what the new biometric touch feature provides.

Up to five fingerprints can be stored on a YubiKey Bio.

Table 5: **Form Factors for YubiKey Bio Series**

Series	5 Bio	5C Bio
YubiKey Bio Series	USB-A + bio sensor	USB-C + bio sensor
YubiKey Bio Series Multi-Protocol Edition *	USB-A + bio sensor	USB-C + bio sensor

* YubiKey Bio - Multi-Protocol Edition requires a subscription with YubiKey as a Service.

For form factor description, see *YubiKey Bio Series*.

For a list of YubiKey Bio Series specifications, see the YubiKey Bio Series Product Brief.

For AAGUIDs, see YubiKey hardware FIDO2 AAGUIDs.

1.5.1 YubiKey Bio Series - Multi-protocol Edition

The YubiKey Bio Multi-Protocol Edition is a new product that combines the ease of use of the YubiKey Bio FIDO Edition with a PIV-like smart card interface and unified PIN and fingerprint templates. This unique combination produces a portable authenticator with the convenience of biometric authentication and support for desktop login.

The YubiKey Bio Multi-protocol Edition takes a novel approach to PINs and fingerprints by using the same PIN and fingerprint templates for both FIDO and smart card. The resulting experience is seamless provisioning and use across both desktop sign-in and web use cases. This thereby eliminates the need for the user to enroll fingerprints multiple times or manage individual PINs.

YubiKey Bio Multi-protocol Edition requires a subscription to YubiKey as a Service. Contact Yubico sales team for information about YubiKey as a Service.

1.6 Security Key Series

The Security Key Series are only available with the FIDO (FIDO2/FIDO U2F) protocol. The latest FIDO updates are included with the latest YubiKey firmware on all Security Key Series.

Table 6: **Form Factors for Security Key Series**

Series	5 NFC	5C NFC
Security Key	USB-A + NFC	USB-C + NFC
Security Key Enterprise Edition *	USB-A + NFC	USB-C + NFC

* Security Key Series - Enterprise Edition requires a subscription with YubiKey as a Service.

For form factor description, see *Security Key Series*.

1.6.1 Security Key Series - Enterprise Edition

The Security Key Series - Enterprise Edition includes the features of the *Security Key Series* and has the following additional updates:

- 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.
- Access the serial number over CCID for Security Key - Enterprise Edition running firmware version 5.7.x or later. See *CCID with USB*.

  The serial number is 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. This change was introduced in 5.7.0.
- *PIN Complexity* turned on by default (and cannot be turned off).
- Increase in minimum FIDO pin length to 6. See *Minimum PIN Length*.
- Security Key Series - Enterprise Edition requires a subscription to YubiKey as a Service. Contact Yubico sales team for information about YubiKey as a Service.

Table 7: **Security Key Capabilities by Firmware Version**

Capability	5.7.x Enterprise Edition	5.7.x	5.4.x Enterprise Edition	5.0.x-5.4.x
Serial Number	Yes	No	Yes	No
Serial Number over CCID	Yes	No	No	No
PIN Complexity	Yes	No	No	No
FIDO Minimum PIN Length	6	4	4	4

Security Key Series - Enterprise Edition requires a subscription to YubiKey as a Service. Contact Yubico sales team for information about YubiKey as a Service.

For a list of Yubico Security Key Series specifications, see the Security Key Series Product Brief.

For AAGUIDs, see YubiKey hardware FIDO2 AAGUIDs.

1.7 YubiKey 5 CSPN Series

Important: The YubiKey 5 CSPN Series has been discontinued.

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.

1.8 Firmware Certifications

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 firmware, see:

- *5.8 Firmware Specifics*, most recent firmware release
- *5.7 - 5.6 Firmware Specifics*
- *Pre-5.6 Firmware Specifics*

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 with its intuitive and easy-to-use (GUI) interface or ykman the lightweight command line (CLI) software package installable on many OSs.

Note: Yubico periodically updates its firmware to take advantage of features and capabilities introduced into the ecosystem. YubiKeys are programmed in Yubico's facilities with the latest available firmware. Once programmed,

YubiKeys cannot be updated to another version. The firmware cannot be altered or removed from a YubiKey.

1.8.1 FIDO Authenticator Supported Certifications

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) Proposed Standard, February 26, 2026.

1.8.2 NIST Certification: FIPS

Yubico submitted the firmware for releases 5.7.4 to NIST and the organization approved the certification for FIPS 140-3 Security Levels 2. The certificate can be found at Cryptographic Module Validation Program (CMVP) Certificate #5291.

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 Features*.

1.8.3 CCN Certification: 5 Series NFC

Yubico submitted the firmware for release 5.7.4 to the Certification Organization for the National Cryptologic Centre (CCN) and the organization approved the certification. The CCN Certification is published in the CPSTIC catalog, *Catálogo de Productos y Servicios de Seguridad de las Tecnologías de la Información y la Comunicación (CPSTIC)*, Yubikey 5 CCN Series 5.7.4.

For more information about the YubiKey 5 CCN Series, see *YubiKey 5 CCN Series*.

1.8.4 ANSI Certification: CSPN

The YubiKey 5 CSPN Series has been discontinued.

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*.

CHAPTER

TWO

PHYSICAL ATTRIBUTES

The serial number is printed on the back of every YubiKey in the YubiKey 5 Series, YubiKey 5 FIPS Series, YubiKey 5 CCN Series, Yubikey 5 Enhanced PIN Series, Security Key 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, Security Key Series, and Bio Series except on the 5C Nano form factor, which is too small to accommodate the 2D barcode.

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

See the Yubico technical data sheet, What is a YubiKey? Product specification for additional certifications and specifications.

2.1 YubiKey NFC Form Factor



YubiKey - NFC plus USB-A Form

Series with this Form	Additional References
YubiYubiKey 5	*YubiKey 5 Series*
YubiKey FIPS	*YubiKey 5 FIPS Series*
YubiKey CCN	*YubiKey 5 CCN Series*
YubiKey Enhanced PIN	*YubiKey 5 Enhanced PIN Series*
Yubico Security Key	*Security Key Series*

See also, Support article, YubiKey 5 NFC.

2.2 YubiKey C NFC Form Factor



YubiKey - NFC plus USB-C Form

Series with this Form	Additional References
YubiYubiKey 5	*YubiKey 5 Series*
YubiKey FIPS	*YubiKey 5 FIPS Series*
YubiKey CCN	*YubiKey 5 CCN Series*
YubiKey Enhanced PIN	*YubiKey 5 Enhanced PIN Series*
Yubico Security Key	*Security Key Series*

See also, Support article, YubiKey 5 NFC.

2.3 YubiKey Nano Form Factor



YubiKey Nano - USB-A Form

Series with this Form	Additional References
YubiYubiKey 5	*YubiKey 5 Series*
YubiKey FIPS	*YubiKey 5 FIPS Series*
YubiHSM	*YubiHSM Auth Specifics*

See also, Support article, YubiKey 5 Nano.

2.4 YubiKey 5C Form Factor



YubiKey - USB-C Form

Series with this Form	Additional References
YubiYubiKey 5	*YubiKey 5 Series*
YubiKey FIPS	*YubiKey 5 FIPS Series*

See also, Support article, YubiKey 5C.

2.5 YubiKey 5C Nano Form Factor



YubiKey Nano - USB-C Form

Series with this Form	Additional References
YubiYubiKey 5	*YubiKey 5 Series*
YubiKey FIPS	*YubiKey 5 FIPS Series*

See also, Support article, YubiKey 5C Nano.

2.6 YubiKey 5Ci Form Factor



YubiKey - USB-C plus Lightning Form

Series with this Form	Additional References
YubiYubiKey 5	*YubiKey 5 Series*
YubiKey FIPS	*YubiKey 5 FIPS Series*

See also, Support article, YubiKey 5Ci.

2.7 YubiKey 5 FIPS Series

YubiKey 5 FIPS is available in all YubiKey 5 Series form factors.

Table 7: **YubiKey Series 5 FIPS Form Factors**

NFC plus USB-A	NFC plus USB-C	Nano USB-A	USB-C	Nano USB-C	USB-C i
					

Additional References:

- *YubiKey 5 FIPS Series*
- *YubiKey 5 FIPS Series Features*

2.7.1 FIPS-Specific Marking

All of the keys in the YubiKey 5 FIPS Series have the acronym **FIPS** inscribed on the backs of the YubiKeys, underneath the Data Matrix code. **v5** runs up on the left side of the Data Matrix code, except for the YubiKey 5C Nano, which has it on the right.

Table 8: **YubiKey Series 5 FIPS-Specific Markings**

NFC and USB-A	NFC and USB-C	Nano USB-A	USB-C	Nano USB-C	USB-C i
					

2.8 YubiKey 5 Enhanced PIN Series

Table 9: **Enhance PIN Form Factors**

NFC and USB-A	NFC and USB-C
	

Additional References:

- *YubiKey 5 Enhanced PIN Series*
- Support articles:
  - YubiKey hardware FIDO2 AAGUIDs
  - YubiKey 5 NFC - Enhanced PIN features
  - YubiKey 5C NFC - Enhanced PIN features
  - Enhanced PIN complexity

2.8.1 Enhanced PIN Specific Marking

All of the keys in the YubiKey 5 Enhanced PIN Series have the acronym **EPIN** and the serial number inscribed on the backs of the YubiKeys, underneath the Data Matrix code.



YubiKey Enhanced PIN Series Markings

2.9 YubiKey 5 CCN Series

Table 10: **CCN Form Factors**

NFC and USB-A	NFC and USB-C
	

Additional Reference:

- *YubiKey 5 CCN Series*

2.9.1 CCN Specific Marking

All of the keys in the YubiKey 5 CCN Series have the acronym **CCN** and the serial number inscribed on the backs of the YubiKeys, underneath the Data Matrix code.



YubiKey CCN Series Markings

2.10 YubiKey Bio Series

The Yubico Bio Key Series are a standalone series and are not within the YubiKey 5 Series. Though they share common functionality.

The YubiKey Bio series has two editions: FIDO Edition and YubiKey Bio Multi-protocol Edition. They are available with either a USB-A or USB-C connector in addition to the BIO sensor. NFC is not supported.

Table 11: **Bio Series Form Factors**

Bio Sensor and USB-A	Bio Sensor and USB-C
	

Additional References:

- *YubiKey Bio Series*
- *YubiKey Bio Series Specifics*

2.10.1 YubiKey Bio Series 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.

Table 12: **Bio Series Markings**

FIDO Edition March 2024 and later	FIDO Edition before March 2024 and Multi-protocol Edition
	

2.11 Security Key Series

The Yubico Security Key Series are a standalone series and are not within the YubiKey 5 Series. Though they share common functionality.

The Yubico Security Key series are available in form factors with either a USB-A or USB-C connector and the Near Field Communication (NFC) connection.

Table 13: **Security Key Series Form Factors**

NFC and USB-A	NFC and USB-C
	

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

Table 14: **Security Key Form Factor Byte Values**

Form Factor	YubiKey Value	Security Key Value, FW 5.4+
NFC and USB-A	0x01	0x41
NFC and USB-C	0x03	0x43

Additional References:

- *Security Key Series Overview*
- Support articles:
  - Security Key NFC
  - Security Key NFC - Enterprise Edition
  - Security Key C NFC
  - Security Key C NFC - Enterprise Edition

2.11.1 Security Key Series Specific Marking

All of the keys in the YubiKey 5 Security Key Series have the acronym **FIDO** inscribed on the backs of the Security Keys. This applies to both the Enterprise Edition and non-enterprise Security Keys.

As of January 2023, the markings on the back of the security keys are different for the non-enterprise and enterprise editions.

- *Non-Enterprise* edition of the security key series the back only has FIDO inscribed.
- *Enterprise Edition* security key series the back has FIDO, scan code, and serial number inscribed.

Table 15: **Security Key Series Markings**

Security Key Non-Enterprise	Security Key Enterprise
	

2.12 YubiKey Connection Interfaces

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

2.12.1 USB Connector

All of the models in the YubiKey 5 Series and Yubico Security Key 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 or USB C or both connectors are supported.

The USB PID and iProduct string changes depending on which of the USB interfaces are enabled. Additionally, Security Key Series and YubiKey Bio Series, PID and supported USB interfaces can be different depending on the YubiKey firmware version. The PID and iProduct strings are described in the YubiKey USB ID Values.

See also, Yubico SDK, Physical Interfaces.

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**.

CCID with USB

The Chip Card Interface Device (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.

As of YubiKey 5.8, Security Keys and YubiKey Bio - FIDO Edition now ship with the CCID interface enabled by default, reporting to the operating system as a combined FIDO+CCID device.

This change modifies the reported USB Product ID (PID) for our standard Security Keys and Bio FIDO keys. If you hardcode PID checks into your environment or just want to cross-reference your device, you can find a comprehensive list of all possible PID values at YubiKey USB ID Values.

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

See *FIDO over CCID*.

FIDO with USB

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).

As of YubiKey Firmware 5.8, FIDO over Chip Card Interface Device (CCID) allows CTAP 2.x commands to be carried using ISO7816 APDU (Application Protocol Data Unit) messaging over the USB CCID smart card interface instead of the USB HID (Human Interface Device) FIDO interface. For platforms, this provides a standards-defined alternative transport for CTAP that integrates with existing smart card infrastructure.

See *FIDO2 with USB-C or Lightning*.

OTP with USB

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.

2.12.2 NFC Interface

The Near Field Communication (NFC)-capable hardware security keys from Yubico provide an NFC wireless interface and a USB interface. 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.

Yubico NFC YubiKeys:

- YubiKey 5 and YubiKey 5C
- YubiKey 5 FIPS and YubiKey 5C FIPS
- YubiKey 5 Enhanced PIN and YubiKey 5C Enhanced PIN
- Security Key and Security Key C

For an example of the NDEF URI format, click the YubiKey NFC demo link. The `<OTP>` value is replaced with the OTP generated by the YubiKey.

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

For operations that require a touch, touch the YubiKey within the first 20 seconds of the operation request to succeed.

When using a desktop NFC reader:

- To help prevent unintended access, a YubiKey placed on a desktop NFC reader might power down unless the NFC reader power-cycles. This 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.

2.12.3 Apple Lightning® Connector

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 iOS SDK. See Yubico iOS SDK
- Register the app with Yubico. See Register HYour App.

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.

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 iOS SDK. See Yubico iOS SDK
- Register the app with Yubico. See Register HYour App.

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.

FIDO2 with USB-C or Lightning

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

iPad and iPad Pro

For users of keys in the YubiKey 5 Series, because the iPad Pro does not have a Lightning port but does have a USB port, support depends on what you want to do. All those aspects are covered by the second part of our Support article Getting Started with iOS.

From the developer perspective, support for the iPad Pro has some limitations. Consult 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 FIDO2 Security Keys on iOS or iPadOS.

2.13 YubiKey ID Numbers

Various security protocols and use cases require valid devices identification. To locate the ID numbers:

- For Authenticator Attestation Globally Unique Identifier (AAGUID) see YubiKey hardware FIDO2 AAGUIDs.
- For USB Product Identifier (PID), see How to find the USB Product ID (PID) of your YubiKey and YubiKey USB ID values.
- For U2F attestation Object Identifier (OID), see The YubiKey FIPS 4 Series Technical Manual, U2F Attestation.

2.14 Verify the YubiKey and Packaging

Each YubiKey Series has its own packaging style. Be sure you received the YubiKey you ordered. See YubiKey packaging for images and description of the packaging used for each YubiKey Series.

When you receive your YubiKey, check that the packaging has not been tampered with. Check that the:

- YubiKey arrives in the correct Yubico packaging
- Plastic shell and seal is not broken or torn
- Cardboard wrap is intact and shows no signs of wear or tear

If you suspect any tampering, Submit a request.

When in doubt, you can also easily test that your YubiKey is a genuine YubiKey.

1. Go to Verify your Yubikey.
2. Insert your Yubikey.
3. Click Verify Device.

   The response shows images your Yubikey form factor and firmware version.

2.15 Care and Cleaning

YubiKeys, including YubiKey Bio, are IP68-rated. This means you can expect the keys to not allow ingress of dust up to level 6 and water resistance up to level 8 as defined by the International Electrotechnical Commission, for ingress protection (IP). See European Standards, BS EN 60529:1992+A2:2013.

To clean the YubiKey, including YubiKey Bio sensors, use wipes impregnated with no more than 70% isopropyl alcohol.

CHAPTER

THREE

PROTOCOLS AND APPLICATIONS

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

3.1 FIDO Specifics

This section describes YubiKey integration with Fast Identity Online (FIDO): FIDO2, Web Authentication (WebAuthn), and FIDO Universal 2nd Factor (U2F).

3.1.1 FIDO2 Features

For an overview of the FIDO2 features that are supported with the 5.7.x and later firmware, see *5.7 - 5.6 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.

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 people to use a single device (the YubiKey) to provide two authentication factors: something they have - the YubiKey, and something they know (a PIN) or a unique physical attribute (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 registering 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 eight incorrect PIN attempts before the YubiKey FIDO2 application locks), there is no security benefit to 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. However, previous values for the PIN are not stored within the YubiKey, so if the current PIN is forgotten, an older PIN will not be recognized.

Locking FIDO2 Credentials

Note: By default, no PIN is set.

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. This is achieved by performing user verification (UV) at the time of authentication. Where UV is the user tapping the YubiKey sensors or the BIO sensor,and entering a PIN value that is stored on the YubiKey.

See *Minimum PIN Length*.

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. That is, the site or service validates the user through the stored discoverable credentials on YubiKey. Users authenticate their login with a tap on the Yubikey and do not need to enter a username.

Not all web sites or service providers offer support for discoverable credentials. In these cases, no information is stored on the YubiKey.

FIDO over CCID

As of YubiKey Firmware 5.8, FIDO over Chip Card Interface Device (CCID) allows CTAP 2.x commands to be carried using ISO7816 APDU (Application Protocol Data Unit) messaging over the USB CCID smart card interface instead of the USB HID (Human Interface Device) FIDO interface.

This means FIDO operations can run through the operating system's smart card stack (PC/SC), using the same communication model as traditional smart card applications. For platforms, this provides a standards-defined alternative transport for CTAP that integrates with existing smart card infrastructure. For RPs, no WebAuthn changes are required as transport selection is handled by the platform. However, broader transport support increases deployment reliability in managed environments.

CTAP 2.x supports transport binding using Application Protocol Data Unit (APDU) over USB CCID (ISO7816) and NFC (ISO14443). This allows FIDO commands to be carried through the smart card (PC/SC) stack instead of through the USB HID FIDO interface. SCP11b secure channel support is currently NFC-only. Extending it to CCID and USB enables transport-agnostic secure provisioning for all key types and can support services such as Yubico Remote Pre-Registration.

This means FIDO operations can run through the operating system's smart card stack (PC/SC), using the same communication model as traditional smart card applications. For platforms, this provides a standards-defined alternative transport for CTAP that integrates with existing smart card infrastructure. For RPs, no WebAuthn changes are required as transport selection is handled by the platform. However, broader transport support increases deployment reliability in managed environments.

Note: Where power consumption issues may occur, such as in with some MacBooks, consider disabling this option.

Default Values

PIN: None set.

YubiKeys are generally shipped without a PIN. The user sets the PIN. Some YubiKeys shipped through the YubiKey as a Service, can have a pre-registered PIN, that the user is forced to change upon first use.

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.

Supported Extensions

As of YubiKey Firmware 5.8, the YubiKey 5 Series supports CTAP 2.3 and 2.2 Authenticator API 2.3 and Defined Extensions 2.3.

As of YubiKey Firmware Prior 5.7.4, the YubiKey 5 Series supports CTAP 2.1 Authenticator API 2.1 and Defined Extensions 2.1.

For YubiKeys with Firmware Prior 5.7.4, 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**

Extension and feature	Firmware Versions 5.8.x	5.7.x	5.6.x	5.5.x	5.4.x	5.3.x	5.2.x	5.1.x	5.0.x
Standards and Algorithms									
ECC P256 Credentials	yes	yes	yes	yes	yes	yes	yes	yes	yes
Ed-DSA/Ed2 Credentials	yes	yes	yes	yes	yes	yes	yes		
ECC P384 Credentials	yes	yes	yes						
CTAP Extensions									

continues on next page

Table 1 – continued from previous page

Extension and feature	Firmware Versions 5.8.x	5.7.x	5.6.x	5.5.x	5.4.x	5.3.x	5.2.x	5.1.x	5.0.x
Credential Protection `credProt`	yes	yes	yes	yes	yes	yes	yes		
HMAC-Secret `hmac-sec`	yes	yes	yes	yes	yes	yes	yes		
HMAC-Secret `MakeCred` `hmac-sec`	yes								
Third Party Payment authentica `thirdPar`	yes								
CTAP Authenticator API									
PIN Protocol v2	yes	yes	yes	yes	yes				
Credential Blob `credBlob`	yes	yes	yes	yes					

continues on next page

Table 1 – continued from previous page

Extension and feature	Firmware Versions 5.8.x	5.7.x	5.6.x	5.5.x	5.4.x	5.3.x	5.2.x	5.1.x	5.0.x
	4kB	4kB	1kB	1kB					
Authentica Large Blob, `largeBlo` required									
Attestation									
Enterprise Attestation	yes	yes	yes						
	yes	yes							
Attestatio Formats `attestat`									
	yes								
Attestatio Formats Preference `attestat`									
PIN Management									
ForceChar	yes	yes	yes	yes					
	yes	yes	yes	yes					
Minimum PIN Length `minPinLe`									

continues on next page

Table 1 – continued from previous page

Extension and feature	Firmware Versions 5.8.x	5.7.x	5.6.x	5.5.x	5.4.x	5.3.x	5.2.x	5.1.x	5.0.x
Maximum PIN Length `maxPinLe`	yes								
PIN Complexity extension `pinCompl`	yes								
Count when user verifies PIN `uvCountS`	yes								
Credentials									
Credential Management	yes	yes	yes	yes	yes	yes	yes		
Always Require User Verificatic `alwaysUV`	yes	yes	yes	yes					
Make Credential UV Not Required	yes	yes	yes	yes					

continues on next page

Table 1 – continued from previous page

Extension and feature	Firmware Versions 5.8.x	5.7.x	5.6.x	5.5.x	5.4.x	5.3.x	5.2.x	5.1.x	5.0.x
Persistent PIN User Access Token (PPUAT)	yes								
Persistent Credential Management Read Only (PCMR)	yes								
YubiKey Functions									
Biometric Enrollment	yes	yes	yes	yes					
Built-in UV (fingerprint)	yes	yes	yes	yes					
Long Touch Reset	yes								
previewSign	yes								
YubiHSM Auth	yes 1	yes	yes	yes	yes				

Note:

1. YubiHSM Auth change without deleting and recreating.

CTAP Specifics

Client to Authenticator Protocol (CTAP) protocol, a specification in FIDO2, is supported on YubiKeys. CTAP 2.1 is supported on YubiKey firmware 5.7.x or later. CTAP 2.3 is supported on YubiKey firmware 5.8.x or later.

Because CTAP implementation is developer focused, for additional information, see Yubico's Developer site for Client To Authenticator Protocol (CTAP) - FIDO2 Developer Guide.

See also:

- In this guide, *FIDO Specifics*.
- FIDO Alliance, User Authentication Specifications Overview.
- Client to Authenticator Protocol (CTAP) Proposed Standard.

Enterprise Attestation with FIDO 2

CTAP 2.3 and YubiKey Firmware 5.8, supports a post-quantum enabled ecosystem, `attestationFormatsPreference` option in `authenticatorGetInfo`. This allows greater control and flexibility in determining the capabilities of authenticators. Currently more of a future proofing feature, it allows potential expansion as alternate attestation formats become available.

The YubiKey firmware 5.8 API communicates supported attestation formats, preparing for a post-quantum cryptography (PQC) enabled ecosystem. Though currently, there are no standard post-quantum attestation formats defined. If one is defined in the future, this `GetInfo` extension enables platforms to determine which formats are supported by a security key.

See CTAP 2.3 authenticatorGetInfo.

As of YubiKey firmware 5.7 through custom configured keys, Enterprise Attestation (EA) enables pre-defined Relying Parties (RPs), such as Identity Providers (IdPs), to read the YubiKey serial number 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. See *Enterprise Attestation Platform/RP Support* and *CTAP 2.1 Features*.

EA's ability to identify individual authenticators as opposed to just the type of authenticator changes the privacy model of the FIDO protocol. This makes the FIDO credential behave more like a certificate.

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 helps 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.

Note: On Enterprise Attestation enabled YubiKeys, if a FIDO reset is performed the Enterprise Attestation is disabled. To re-enable Enterprise Attestation, use `ykman` CLI. See YubiKey Manager (ykman) CLI User Guide, FIDO Commands.

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

Enterprise Attestation 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

As of YubiKey firmware 5.7.4, the Minimum PIN Length Extension `minPINLengthExtension` enables the minimum PIN length to be set on an authenticator. `minPINLengthExtension` enables:

- a Relying Party (RP) to enforce PIN length requirements, for example in regulated environments.
- IdPs to support FIDO registration self-enrollment processes by enforcing the configured minimum PIN length.

`minPINLength` is configurable only by platforms or by communicating with the YubiKey directly. We recommend using a local client tool such as the Yubico Authenticator or ykman.

IdPs, RPs pre-defined by the organization, and end users are able to query the `minPINLength` of the authenticator, but only if it is configured via an allowed list on the YubiKey.

Once set, the PIN length cannot be shortened until the authenticator is reset.

Generally, PIN length is between 4 and 63 alphanumeric characters, but the actual minimum PIN length varies depending on the firmware version, whether or not the YubiKey is a FIPS key, and whether *PIN Complexity* is added or not.

Table 2: **Minimum PIN Length**

YubiKey 5.7 and later with PIN complexity	6
YubiKey 5.7 and later without PIN complexity	4
YubiKey FIPS 5.7	8
YubiKey FIPS prior to 5.7	6
YubiKey prior to 5.7	4

On the YubiKey Bio Multi-protocol Edition, the PIN is shared between the PIV and FIDO2 applets. *PIN length requirements* depend on firmware version and YubiKey application status.

The special (developer-focused) requirements for the PIN are described in The FIDO2 PIN.

In YubiKey firmware version 5.7.4, the `minPINLength` extension resolves compliance requirements for organizations that need to enforce use of specific PIN lengths.

Prior to 5.7, enforcing PIN length was only possible with:

- YubiKey 5 FIPS Series authenticators certified for FIPS 140-2, which have a minimum PIN length of 6.
- YubiKey 5 FIPS Series Enhanced PIN and Security Key Series have minimum PIN length set to 6 by default.
- Custom configuration, which RPs could do when the authenticator had a custom AAGUID.

Changing Minimum PIN Length

To change the minimum PIN length, see Increasing the minimum PIN length.

See also, CTAP 2.3, Setting a minimum PIN Length.

- To re-attempt to enter the PIN after you have entered an incorrect PIN three times in succession, power-cycle the FIDO2 application.
- Once a FIDO2 PIN is set, it can be changed but removal of a FIDO2 PIN requires a FIDO2 reset.
- If the PIN is entered incorrectly eight times in succession, the FIDO2 application locks and FIDO2 authentication is no longer possible. To unlock the FIDO2 application, 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 the U2F sites.

Note: The YubiKey 5 supports FIDO2 credential management. This enables selectively deleting resident keys. See our article YubiKey 5.2 enhancements to FIDO 2 Support for details.

Minimum PIN Length 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.

ForceChangePIN

As of YubiKey firmware 5.7.4, `ForceChangePIN` (FIDO 2.1 specification) 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 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 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.

ForceChangePIN Platform Support

There is no need for explicit RP support for forcing PIN change. The `forceChangePIN` flag is set on the client/platform side. The client/platform triggers the change PIN flow. The PIN change can only be set by communicating directly with the YubiKey.

Chrome and some Chromium-based browsers support it. Windows 11 is required on Windows platforms.

Always Require User Verification

Always Require User Verification (UV), `alwaysuv`, was introduced to prompt users for user verification with each use (authentication and registration). This ensures consistent 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.

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

Always Require User Verification (UV) has a default setting for each YubiKey series.

Table 3: **Always Require User Verification Defaults by YubiKey**

YubiKey Series	Default	If YubiKey is reset	Notes
YubiKey 5 Series	Disabled	Reverts to disabled	Can toggle.
YubiKey 5 CCN Series	Disabled	Reverts to disabled	Can toggle.
YubiKey 5 Enhanced PIN Series	Enabled	Reverts to enabled	Can toggle.
YubiKey 5 FIPS Series	Enabled	Not applicable	Cannot be disabled.
YubiKey Bio Series	Enabled	Reverts to enabled	Can toggle. Always asks for biometrics and never only “plain touch”/User Presence (UP), even in a second factor flow when UV is not required.
Security Key Series	Disabled	Reverts to disabled	Can toggle.

Always User Verification Platform/RP Support

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

AlwaysUV and FIDO2

If you disable FIDO2 and leave AlwaysUV enabled, U2F fails.

To prevent U2F failing:

1. Disable AlwaysUV first.
2. Then disable FIDO2.

To correct U2F failure, if you disabled FIDO2 and did not disable AlwaysUV first:

1. Enable FIDO2.
2. Disable AlwaysUV.
3. Disable FIDO2.

Blob Storage

One of 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 Relaying Party (RP).

Large blob storage is:

- 4096 bytes on firmware 5.7.0 and later
- 1024 bytes on firmware 5.5.x and 5.6.x

Credential Blob

Credential Blobs, `credBlob`, 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` extension 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

Large blob, `authenticatorLargeBlob`, storage is 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 size of the Large Blob varies, depending on the YubiKey firmware version:

- 4096 bytes on firmware 5.7.0 and later
- 1024 bytes on firmware 5.5.x and 5.6.x.

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 enables the authenticator to be hardware-bound and to 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 and does 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.

Calculating the RPID hash

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 hash of an URL, apply the SHA-256 algorithms over the RPID. For example, the RPID `yubico.com` has the RPID hash of `378209b72defcba91dcbf854edb4daa648828a2cbd180afc77a74434655a1c7d`. This can be calculated by running `echo -n "yubico.com" | openssl dgst -sha256`, which returns the result `SHA2-256(stdin)= 378209b72defcba91dcbf854edb4daa648828a2cbd180afc77a74434655a1c7d`.

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

3.1.2 FIDO2 Management Tasks

This section describes:

- *Tools for Managing the FIDO2 Application*
- *Managing Discoverable Credentials*
- *Resetting the FIDO2 Application*
- *Setting or Changing the FIDO2 PIN*
- *Enrolling Fingerprints on the YubiKey Bio*

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.

ykman CLI for Windows, MacOS and Linux

Install the most recent version of the ykman, because the YubiKey Manager GUI includes an older version of the ykman CLI.

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.

For Windows, MacOS, Linux, download the Yubico Authenticator with its intuitive and easy-to-use graphical interface, the ykman, a lightweight software package installable on many OS, or the YubiKey Manager GUI, though it is not as robust as the other tools.

These are general purpose utilities that is able to configure many of the applications on the YubiKey in addition to FIDO2. They require 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.

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.

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.

Resetting the FIDO2 Application

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. Also note that FIDO2 reset performed over a Lightning connection is only supported for devices with firmware 5.7.4 and higher.

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. Also note that FIDO2 reset performed over a Lightning connection is only supported for devices with firmware 5.7.4 and later.

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

1. Download and install YubiKey Manager GUI.

   Alternatively, use either the Yubico Authenticator with its intuitive and easy-to-use graphical interface or the ykman, a lightweight software package installable on many OS.
2. Insert your YubiKey or Security Key into an available USB port on your computer.
3. Open YubiKey Manager.

   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.

   ---

4. Navigate to Applications > FIDO2.
5. Click "Reset FIDO" > "YES".
6. Follow the 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.

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.

Note: This does not apply to the YubiKey Bio Multi-protocol Edition where the two PINs are shared between the applications.

Setting or Changing the FIDO2 PIN on Windows

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 GUI.

   Alternatively, use either the Yubico Authenticator with its intuitive and easy-to-use graphical interface or the ykman, a lightweight software package installable on many OS.

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 “Set PIN” or “Change PIN”.
6. Follow the prompts 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.

Enrolling Fingerprints on the YubiKey Bio

Videos demonstrating how to enroll fingerprints in the YubiKey Bio can be found at Set up your YubiKey. Click the *enroll your fingerprint* link.

3.1.3 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.

3.2 Smart Card - PIV Compatible Specifics

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 Minidriver. See YubiKey Minidriver User Guide.

Note: The YubiKey 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.

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

3.2.1 Default Values

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

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

3.2.2 Supported Algorithms

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

Table 4: **Available PIV Algorithms per Firmware Version**

Algorithm (Identifier)	Firmware Versions 5.8.x	5.7.4	5.0.x-5.6.x
RSA-1024 (0x06)	Yes	Yes	Yes
RSA-2048 (0x07)	Yes	Yes	Yes
RSA-3072 (0x05)	Yes	Yes	–
RSA-4096 (0x16)	Yes	Yes	–
ECC P-256 (0x11)	Yes	Yes	Yes
ECC P-384 (0x14)	Yes	Yes	Yes
Ed25519/x25519 (0xe0)	Yes	Yes	–

3.2.3 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.

3.2.4 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 and later 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.

3.2.5 Import a Key

For more information, see Import asymmetric key pair.

3.2.6 Move a Key

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

Table 5: **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

3.2.7 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 6: **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

3.2.8 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.

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 form factor, encoded as a one-byte octet-string during manufacturing and returned as a one-byte value.

- NFC and USB-A keychain: 0x01
- Nano USB-A: 0x02
- NFC and USB-C keychain: 0x03
- Nano USB-C: 0x04
- Lightning and USB-C keychain: 0x05
- UNDEFINED: 0x00

Keychain form factors also have space for adding to a keychain. See *YubiKey 5 FIPS Series*.

3.2.9 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` 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 Yubico Authenticator on Windows 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 library

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 7: 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.

3.2.10 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 and later the YubiKey PIV Management Key is AES-192 by default.

3.3 OATH Specifics

For an overview of the OATH features that became available with the 5.7.x and later firmware, see *5.7 - 5.6 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.

Note: 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:

- **Yubico OTP and OATH-HOTP directly on the touch-triggered OTP function** on the YubiKey.
- **OATH-HOTP and OATH-TOTP directly on the OATH function** on the YubiKey (usually called OATH and used with Yubico Authenticator).
- 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; see *OATH Specifics* and *OTP Specifics*.

3.3.1 HOTP and TOTP

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

3.4 OpenPGP Specifics

For an overview of the OpenPGP features that became available with the 5.7.x and later firmware, see *5.7 - 5.6 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 and 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.

3.4.1 Elliptic Curve Cryptographic (ECC) Algorithms

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

In addition to 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 YubiKey 5.2 enhancements to OpenPGP 3.4 support.

3.4.2 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)

3.4.3 Default Values

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

3.5 OTP Specifics

For an overview of the OTP features that became available with the 5.7.x and later firmware, see *5.7 - 5.6 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.

Note: 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:

- **Yubico OTP and OATH-HOTP directly on the touch-triggered OTP function** on the YubiKey.
- **OATH-HOTP and OATH-TOTP directly on the OATH function** on the YubiKey (usually called OATH and used with Yubico Authenticator).
- 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; see *OATH Specifics* and *OTP Specifics*.

3.5.1 Yubico OTP

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

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

The character representation of the Yubico OTP is designed to handle a variety of keyboard layouts. It is crucial that the same code is generated if a YubiKey is inserted into a German computer with a QWERTZ layout, a French one with an AZERTY layout, or a US one with a QWERTY layout. The 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.

3.5.2 Static Password

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

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

3.5.3 HMAC-SHA1 Challenge-Response

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

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

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

3.6 Secure Channel Protocol (SCP03 and SCP11) Specifics

Secure Channel Protocol support was added to YubiKeys as of firmware version 5.3.0. Later versions of YubiKey firmware, as listed in this document, include and add to features and/or functional support.

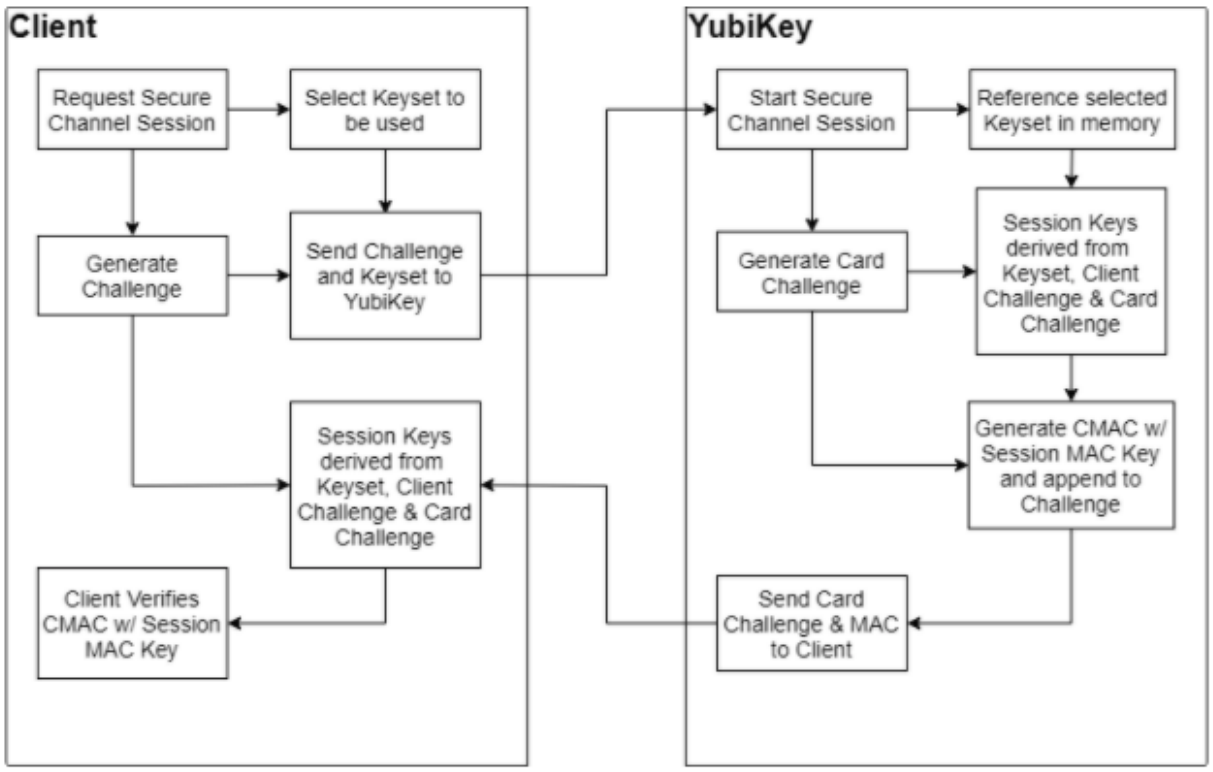
Yubico has implemented a subset of the (GlobalPlatform Secure Channel Protocol 03) specification, specifically, the most secure implementation including command and response message authentication code (MAC) and encryption. This is available in YubKey firmware 5.3.0 and later.

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 5.7.2 and later.

3.6.1 Yubico Secure Channel Technical Description

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

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).

Transport Keys and Session Keys

Table 8: **Transport 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.

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.

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.

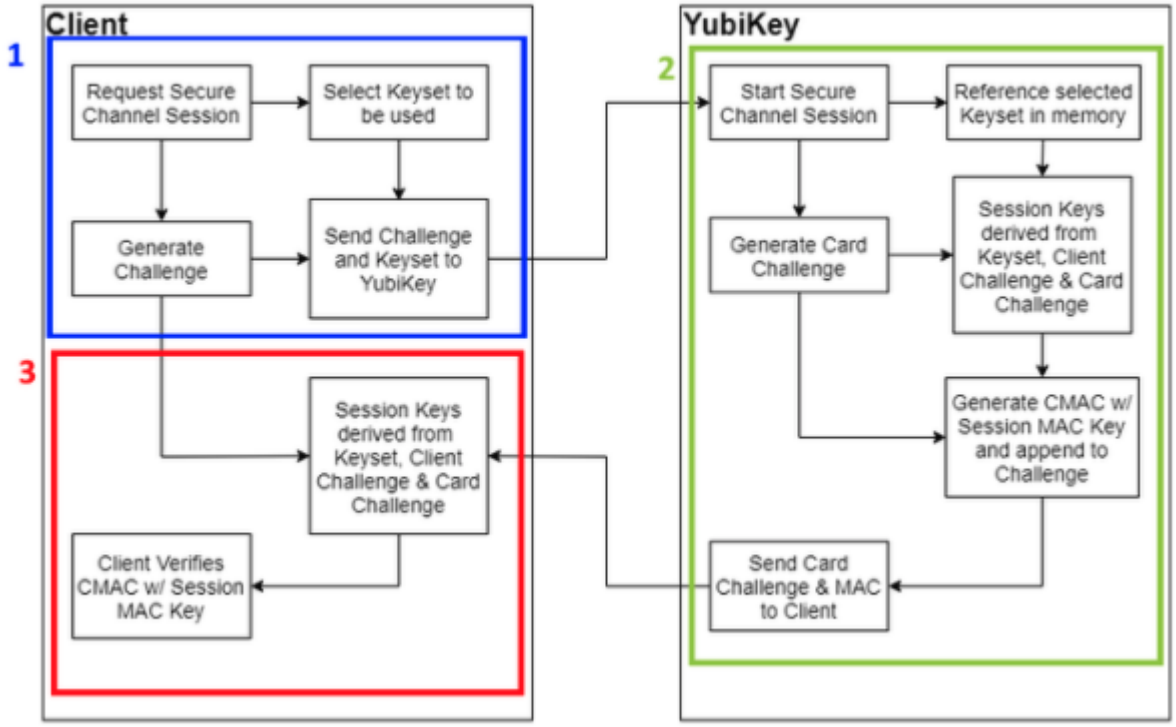
The Response MAC is generated using the encrypted response APDU from the YubiKey and the Session MAC key for Response. Each Response MAC also includes a value derived from the original command MAC sent from the client, providing a verification that the data included in the response corresponds to the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

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.



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.

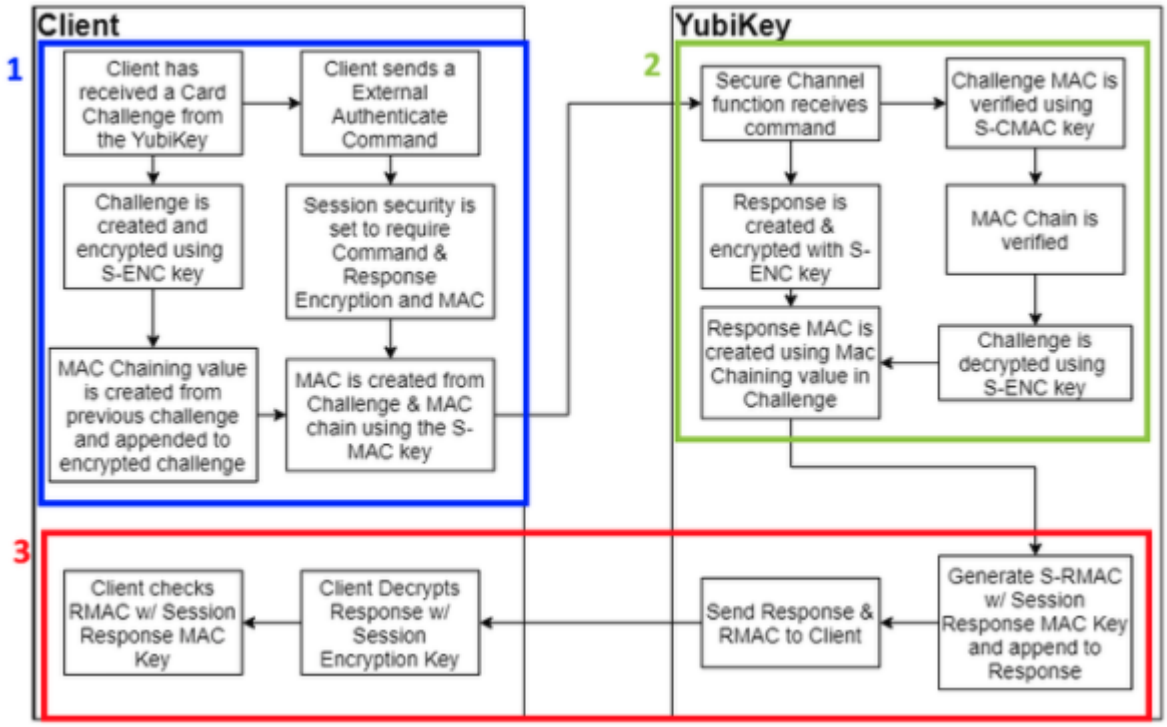
The YubiKey derives the Session keys for Encryption and MAC by generating a card challenge, then using that plus the challenge provided from the client and the selected Transport Key set.

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.



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

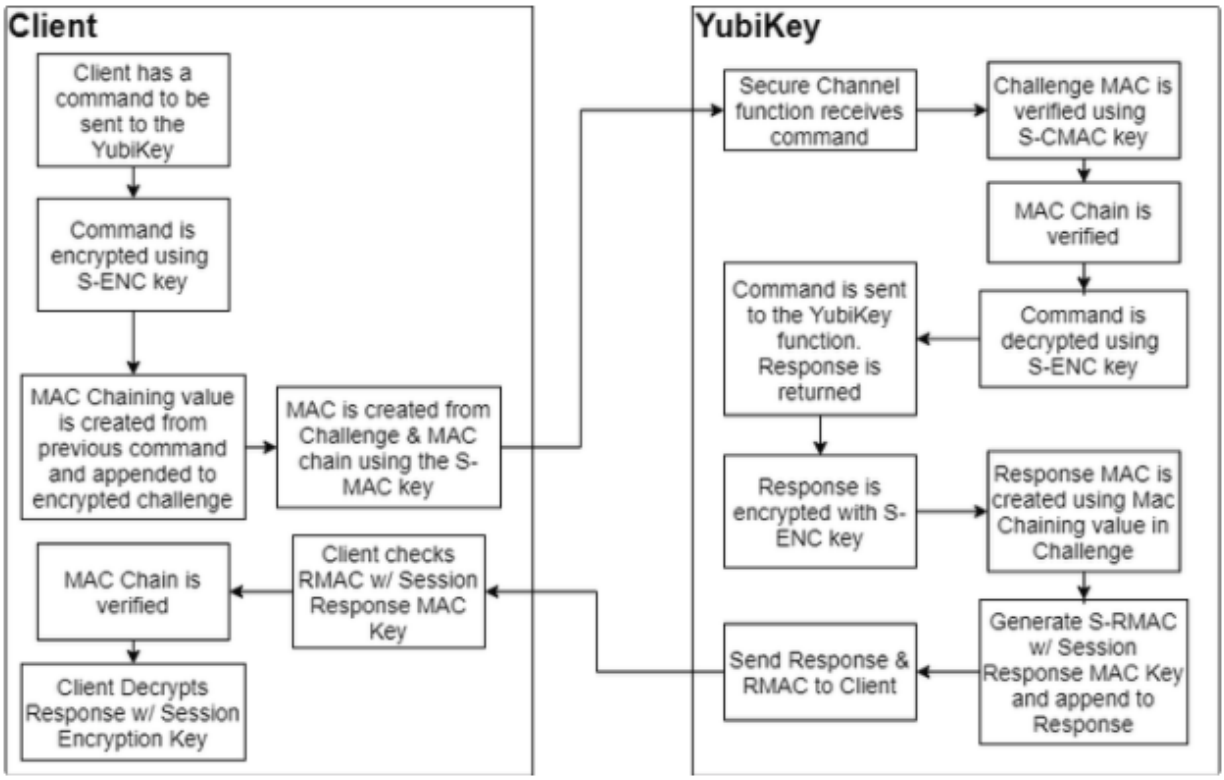
the MAC value and S-MAC key. The MAC Chain is then verified, confirms that the client has the same session keys and that a secure channel has been created. Then the challenge is decrypted using the S-ENC key and a response is created. In addition, the challenge from the client is used to create a new 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.

Communicating Over Secure Channel



Communicating Over Secure Channel Flow

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

Step 1

The command APDU is first encrypted using the Session Encryption Key (S-ENC). 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.

3.6.2 Yubico Secure Channel Key Diversification and Programming

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

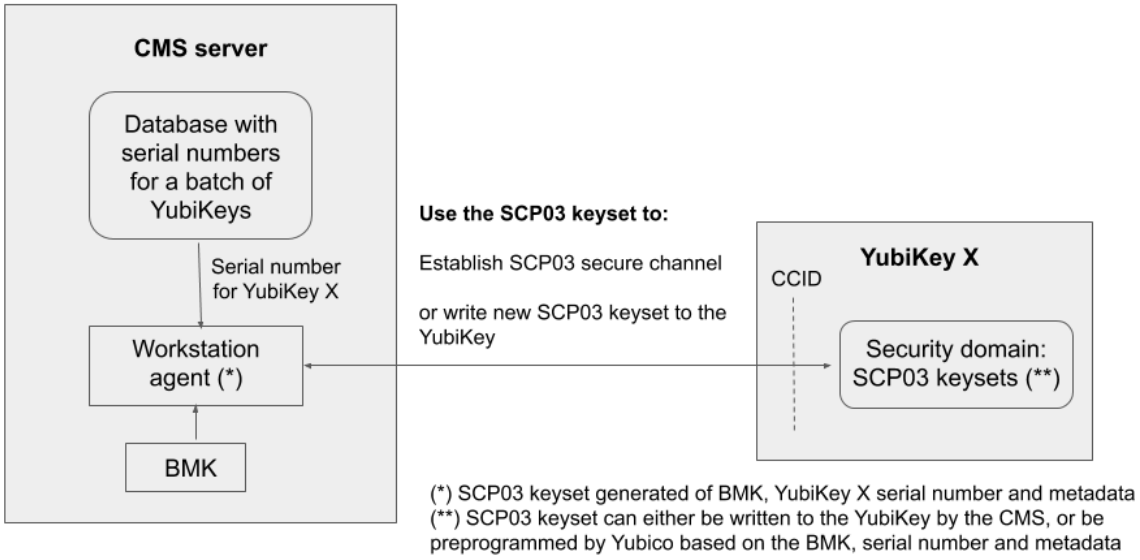
Secure Channel and Security Domains

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.

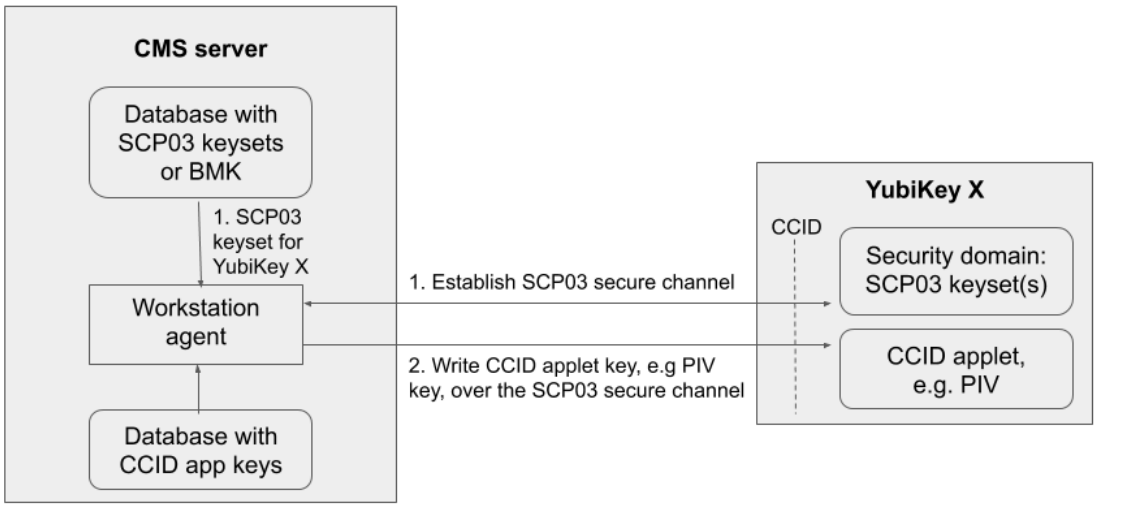
Writing CCID Application Keys over SCP03

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.

SCP03 key diversification



Writing CCID application keys over SCP03



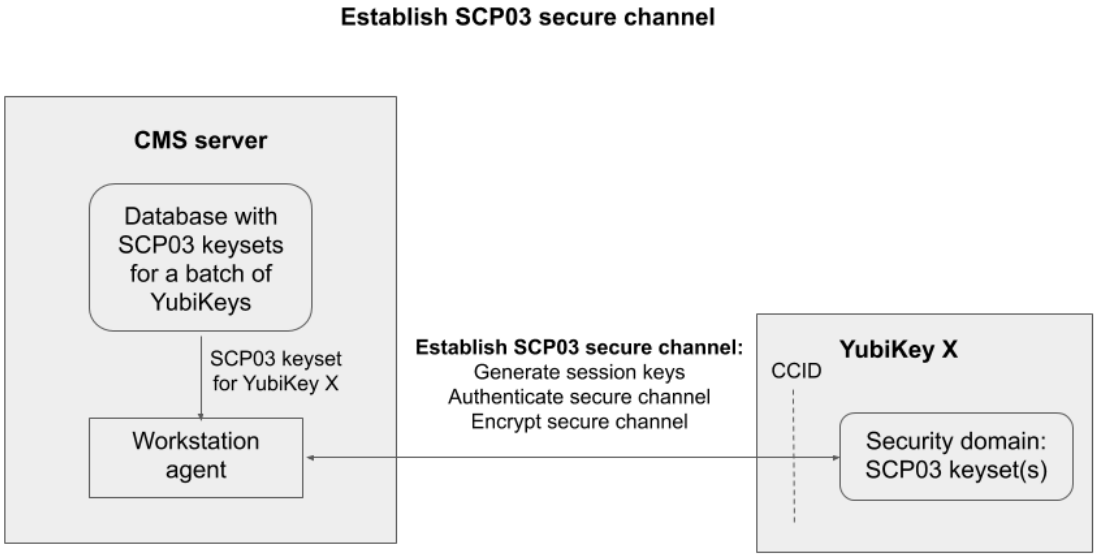
The YubiKey supports up to three secure channel transport key sets. This is to enable more granular control over the establishment of a secure channel to a specific device. The keys in each of these key sets can be overwritten when connected to the YubiKey. This allows for YubiKeys to be shipped with 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.

Key Diversification Option

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.

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.

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 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

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));}
```

Global Platform: 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.

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.

3.6.3 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.

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.

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`.

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.

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.

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.

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 00200080083132333435363fffff
```

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
```

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

3.7 YubiHSM Auth Specifics

YubiHSM Auth is supported by YubiKey firmware version 5.4.3 and above.

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 on YubiKey firmware versions, as listed:

- 5.4.3 or later firmware requires using ykman or Yubico Authenticator to enable it.
- 5.7.x or later firmware is enabled (on) by default.

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.

3.7.1 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).

3.7.2 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.

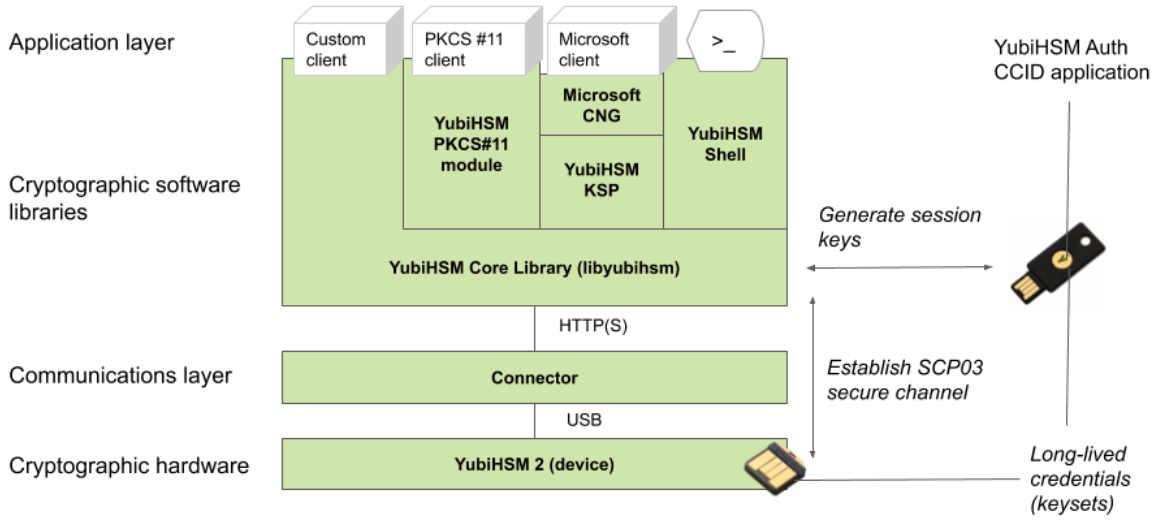
Those long-lived key sets are used by the YubiHSM Auth application to derive a set of three session-specific AES-128 keys using the challenge-response protocol as defined in SCP03:

- 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.

3.7.3 Architecture Overview

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



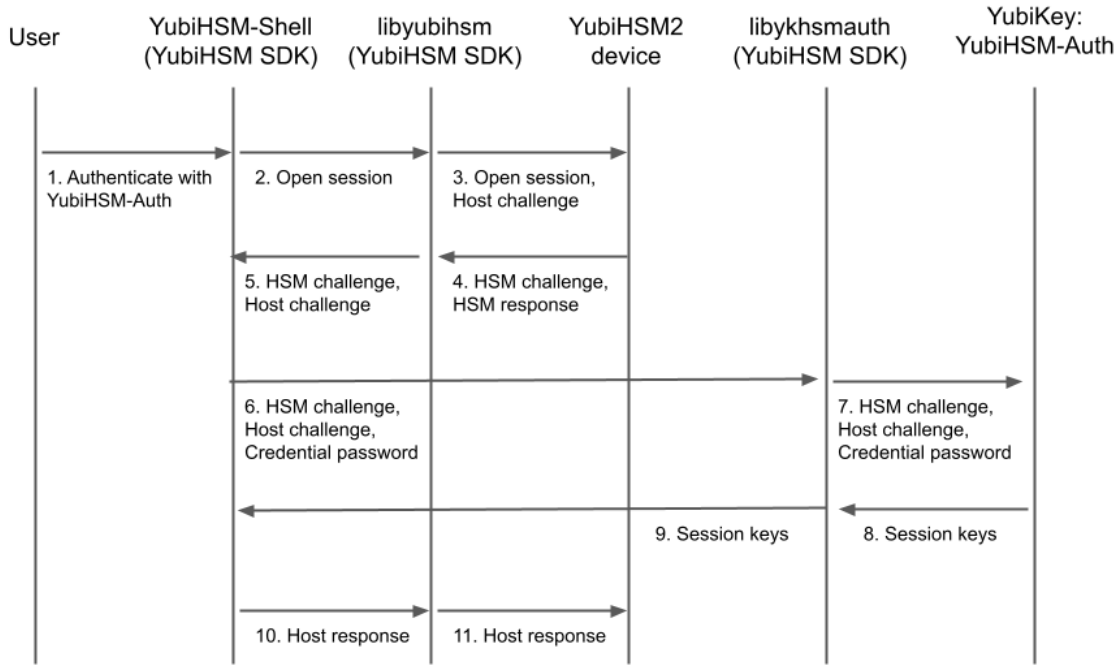
The identical long-lived credentials (key sets) are protected in both the YubiKey YubiHSM Auth application and in the YubiHSM 2. The YubiHSM-Shell software tool can be used for generating the key sets in the YubiHSM 2, and the YubiHSM-Auth software tool can be used for importing the same key sets to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

3.7.4 YubiHSM Auth Flowchart

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



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

3.7.5 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 2 Shell for authentication to the YubiHSM 2.

3.7.6 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 2 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.

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)
  - We support the **Yubico OTP and OATH-HOTP directly on the touch-triggered OTP function** on the YubiKey.
- OATH-TOTP (standard RFC6238)
  - 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 Specifics* and *OTP Specifics*.

CHAPTER

FOUR

YUBIKEY FIRMWARE

These topics describe YubiKey 5 firmware versions and features released with each version.

4.1 Firmware Overview

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 with its intuitive and easy-to-use (GUI) interface or ykman the lightweight command line (CLI) software package installable on many OSs.

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

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

Form Factor	YubiKey Form Name	Firmware Versions						
		5.8	5.7.x	5.5.x	5.4.x	5.3.x	5.2.x	5.0.x
USB-A + NFC	5 NFC	Yes	Yes	Yes	Yes	Yes	Yes	Yes
USB-C + NFC	5C NFC	Yes	Yes	Yes	Yes	Yes	Yes	
USB-A Nano	5 Nano	Yes	Yes	Yes	Yes	Yes	Yes	Yes
USB-C Nano	5C Nano	Yes	Yes	Yes	Yes	Yes	Yes	Yes
USB-A + bio sensor	5 Bio	Yes	Yes	Yes				
USB-C + bio sensor	5C Bio	Yes	Yes	Yes				
USB-C + Lightning	5Ci	Yes	Yes	Yes	Yes	Yes		

4.1.1 YubiKey 5 Series

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

5.8 Firmware - FIDO2 and CTAP 2.3

The 5.8. firmware for the YubiKey 5 Series has a number of features that are available for the first time on the YubiKey 5. The changes and additions are described in detail in *5.8 Firmware Specifics*. In addition to the features that are directly accessible, there are a number of features that require partner support.

5.7.4 Firmware - FIPS Series

The 5.7.4 new features include:

- Certified to NIST's Cryptographic Module Validation Program for FIPS 140-3 validation.
- Enterprise Attestation to support use cases such as derived FIDO credentials
- FIDO2, PIV and OpenPGP minimum PIN length is now 8
- PIN complexity is on by default to adhere to NIST Special Publication 800-63B (and 800-63B-4)
- Larger keys sizes provide better protection than smaller key sizes until Post-Quantum-Cryptography is mature.

5.7 and 5.6 Firmware Prior to 5.7.4

The 5.7 and 5.6 new features include:

- The NFC function with firmware 5.7 is off by default. Plugging the YubiKey into the device in activates the NFC function. For more detail on this specific feature, see *Restricted NFC*.
- The 5.7. firmware for the YubiKey 5 Series has a number of features that are available for the first time on the multi-protocol YubiKey 5.

4.1.2 Firmware Capability Matrices

For form factors released with which firmware versions, see *Introduction to the YubiKey Series*.

YubiKey 5 Series

Table 2: **Features Available per Firmware Version**

Features	Firmware Versions						
	5.8.x	5.7.x	5.5.x	5.4.x	5.3.x	5.2.x	5.0.x
AlwaysUV	Yes	Yes					

continues on next page

Table 2 – continued from previous page

Features	Firmware Versions						
	5.8.x	5.7.x	5.5.x	5.4.x	5.3.x	5.2.x	5.0.x
Attestation Enterprise	Yes	Yes					
Blob Storage	Yes	Yes					
Serial Number	Yes	Yes	Yes	Yes	Yes	Yes	Yes
YubiHSM Auth	Yes	Yes	Yes				
CTAP							
CTAP 2.3 Conditional Mediation, PPUAT, PCMR	Yes						
Pseudo-Random (PRF)	Yes						
Reset LongTouch	Yes						
Reset Transport	Yes						
Third Party Payment	Yes						
FIDO							
FIDO2 CTAP 2.3	Yes						
FIDO U2F	Yes	Yes	Yes	Yes	Yes	Yes	Yes
FIDO2/WebA	Yes	Yes	Yes	Yes	Yes	Yes	Yes
FIDO2 Credential Storage	100	100	25	25	25	25	25
FIDO2 PIN Mgmt	Yes	Yes					
FIDO2 over CCID	Yes						
Protocols							
OATH	Yes	Yes	Yes	Yes	Yes	Yes	Yes

continues on next page

Table 2 – continued from previous page

Features	Firmware Versions						
	5.8.x	5.7.x	5.5.x	5.4.x	5.3.x	5.2.x	5.0.x
OATH Credential Storage	64	64	32	32	32	32	32
OpenPGP version	3.4	3.4	3.4	3.4	3.4	2.1	2.1
OTP	Yes	Yes	Yes	Yes	Yes	Yes	Yes
PIV/Smart Card	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SCP03	Yes	Yes	Yes	Yes			
SCP11	Yes	5.7.4+					

YubiKey 5 FIPS Series

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

Features	Firmware Versions		
	5.7.4	5.4.3	5.4.2
AlwaysUV			
Attestation Enterprise			
Blob Storage			
Serial Number	Yes	Yes	Yes
YubiHSM Auth	Yes	Yes	Yes
CTAP			
CTAP 2.3 Conditional Mediation, PPUAT, PCMR	Yes		
Pseudo-Random (PRF)			
Reset LongTouch			
Reset Transport			
Third Party Payment			
FIDO			
FIDO2 CTAP 2.3			
FIDO U2F	Yes	Yes	Yes
FIDO2/WebAuthn	Yes	Yes	Yes

continues on next page

Table 3 – continued from previous page

Features	Firmware Versions		
	5.7.4	5.4.3	5.4.2
FIDO2 Credential Storage	100	100	25
FIDO2 PIN Mgmt			
FIDO2 over CCID	Yes		
Protocols			
OATH	Yes	Yes	Yes
OATH Credential Storage			
OpenPGP version	3.4	3.4	3.4
OTP	Yes	Yes	Yes
PIV/Smart Card	Yes	Yes	Yes
SCP03	Yes	Yes	Yes
SCP11	Yes		

Note: YubiKey FIPS series remains on firmware version 5.7.4 while FIPS certification for firmware 5.8 is processing.

YubiKey 5 Series - Enhanced PIN

YubiKey 5 Enhanced PIN Series requires a subscription to YubiKey as a Service. Contact Yubico sales team for information about YubiKey as a Service.

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

Features	Firmware Versions	
	5.8.x	5.7.4
AlwaysUV	Yes	Yes
Attestation Enterprise	Yes	Yes
Blob Storage	Yes	Yes
Serial Number	Yes	Yes
YubiHSM Auth	Yes	Yes
CTAP		

continues on next page

Table 4 – continued from previous page

Features	Firmware Versions 5.8.x	5.7.4
CTAP 2.3 Conditional Mediation, PPUAT, PCMR	Yes	
Pseudo-Random (PRF)	Yes	
Reset LongTouch	Yes	
Reset Transport	Yes	
Third Party Payment	Yes	
FIDO		
FIDO2 CTAP 2.3	Yes	
FIDO U2F	Yes	Yes
FIDO2/WebAuthn	Yes	Yes
FIDO2 Credential Storage	100	100
FIDO2 PIN Mgmt	Yes	Yes
FIDO2 over CCID	Yes	
Protocols		
OATH	Yes	Yes
OATH Credential Storage	64	64
OpenPGP version	3.4	3.4
OTP	Yes	Yes
PIV/Smart Card	Yes	Yes
SCP03	Yes	Yes
SCP11	Yes	Yes

YubiKey 5 CCN Series

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

Features	Firmware Versions 5.8.x	5.7.4
AlwaysUV	Yes	Yes

continues on next page

Table 5 – continued from previous page

Features	Firmware Versions	
	5.8.x	5.7.4
Attestation Enterprise	Yes	Yes
Blob Storage	Yes	Yes
Serial Number	Yes	Yes
YubiHSM Auth	Yes	Yes
CTAP		
CTAP 2.3 Conditional Mediation, PPUAT, PCMR		
Pseudo-Random (PRF)		
Reset LongTouch		
Reset Transport		
Third Party Payment		
FIDO		
FIDO2 CTAP 2.3		
FIDO U2F	Yes	Yes
FIDO2/WebAuthn	Yes	Yes
FIDO2 Credential Storage	100	100
FIDO2 PIN Mgmt	Yes	Yes
FIDO2 over CCID		
Protocols		
OATH	Yes	Yes
OATH Credential Storage	64	64
OpenPGP version	3.4	3.4
OTP	Yes	Yes
PIV/Smart Card	Yes	Yes
SCP03	Yes	Yes
SCP11	Yes	Yes

YubiKey Bio Series

The listed firmware versions apply to both YubiKey Bio Series and YubiKey Bio Series - Multi-Protocol editions, unless indicated.

Yubico Bio Series - Multi-Protocol Edition requires a subscription to YubiKey as a Service. Contact Yubico sales team for information about YubiKey as a Service.

Table 6: **Features and Form Factors Available per Firmware Version**

Features	Firmware Versions			
	5.8.x	5.7.x	5.6.x	5.5.x
AlwaysUV	Yes			
Attestation Enterprise	Yes			
Blob Storage	Yes			
Serial Number	Yes	Yes	Yes	Yes
YubiHSM Auth	Yes			
CTAP				
CTAP 2.3 Conditional Mediation, PPUAT, PCMR	Yes			
Pseudo-Random (PRF)	Yes			
Reset LongTouch	Yes			
Reset Transport	Yes			
Third Party Payment	Yes			
FIDO				
FIDO2 CTAP 2.3	Yes			
FIDO U2F	Yes	Yes	Yes	Yes
FIDO2/WebAuthn	Yes	Yes	Yes	Yes
FIDO2 Credential Storage	100	100	25	25
FIDO2 PIN Mgmt	Yes			
FIDO2 over CCID	Yes			
Protocols				
OATH	Yes	Yes	Yes	Yes
OATH Credential Storage	Yes			
OpenPGP version	3.4	3.4	3.4	
OTP	Yes	Yes	Yes	Yes

continues on next page

Table 6 – continued from previous page

Features	Firmware Versions			
	5.8.x	5.7.x	5.6.x	5.5.x
PIV/Smart Card	Multi-Protocol	Multi-Protocol		
SCP03	Multi-Protocol	Multi-Protocol	Multi-Protocol	
SCP11	Multi-Protocol	Multi-Protocol 5.7.2+		

Security Key Series

The listed firmware versions apply to both Security Key Series and Security Series Enterprise editions.

Security Key Series - Enterprise Edition requires a subscription to YubiKey as a Service. Contact Yubico sales team for information about YubiKey as a Service.

Table 7: **Features and Form Factors Available per Firmware Version**

Features	Firmware Versions			
	5.8	5.7.x	5.4.x	5.0.x-5.2.x
AlwaysUV	Yes	Yes		
Attestation Enterprise	Enterprise Edition	Enterprise Edition		
Blob Storage	Yes	Yes		
Serial Number	Enterprise Edition	Enterprise Edition	Enterprise Edition	
YubiHSM Auth	Yes			
CTAP				

continues on next page

Table 7 – continued from previous page

Features	Firmware Versions			
	5.8	5.7.x	5.4.x	5.0.x-5.2.x
CTAP 2.3 Conditional Mediation, PPUAT, PCMR	Yes			
Pseudo-Random (PRF)	Yes			
Reset LongTouch	Yes			
Reset Transport	Yes			
Third Party Payment	Yes			
FIDO				
FIDO2 CTAP 2.3	Yes			
FIDO U2F	Yes	Yes	Yes	Yes
FIDO2/WebAuthn	Yes	Yes	Yes	Yes
FIDO2 Credential Storage	100	100	25	25
FIDO2 PIN Mgmt	Yes	Yes		
FIDO2 over CCID	Yes			
Protocols				
OATH	Yes	Yes	Yes	Yes
OATH Credential Storage	Yes			
OpenPGP version	3.4	3.4	3.4	
OTP	Yes	Yes	Yes	Yes
PIV/Smart Card	Yes			
SCP03	Enterprise Edition	Enterprise Edition		
SCP11	Enterprise Edition	Enterprise Edition 5.7.4+		

SCP03, SCP11, Enterprise Attestation, and Serial Number Support are only available on the Security Key Series Enterprise Edition.

4.2 5.8 Firmware Specifics

This section describes features enabled by the 5.8 firmware. For additional details specific to supported protocols, see *Protocols and Applications*. This firmware update is a toolkit expansion for developer authentication workflows, see Yubico's YubiLab GitHub Repository Build with Us.

- The 5.8 firmware for the YubiKey 5 Series has a number of new and improved FIDO2 features that are available for the first time on the multi-protocol YubiKey 5.
- YubiKey 5.8 supports all of the capabilities of YubiKey 5.7.4 and expands the use of passkeys specific for developer and platform use cases.
- There have been no material changes to the other protocols, such as PIV and YubiOTP in firmware 5.8.

4.2.1 5.8 Features Summary

YubiKey 5.8 firmware enables hardware-backed root of trust for secure digital identity and financial transactions, including privacy-preserving, human-in-the-loop authorization, and passkey-based signatures.

In addition to the YubiKey 5 Series, the features and capabilities listed in the table below, apply to both Enterprise and non-enterprise editions of the Security Keys Series, and both Multi-Protocol and FIDO editions of the YubiKey Bio Series, unless otherwise indicated.

Table 8: **YubiKey 5 Series Firmware 5.8 capabilities**

Firmware 5.8 Capabilities CTAP 2.3, where noted	YubiKey 5 (Std, CCN)	YubiKey 5 Enhanced PIN	Security Key Series	YubiKey Bio Series
Secure payments, Document Signing, Digital Wallets, Credential, Agentic approval				
Secure Payment Confirmation (SPC)	Yes	Yes	Yes	Yes
Third-party payment CTAP extension `thirdPartyPayment`	Yes	Yes	Yes	Yes
WebAuthn Signing Extension `previewSign` (Preview only)	Yes	Yes	Yes	Yes

continues on next page

Table 8 – continued from previous page

Firmware 5.8 Capabilities CTAP 2.3, where noted	YubiKey 5 (Std, CCN)	YubiKey 5 Enhanced PIN	Security Key Series	YubiKey Bio Series
Asynchronous Remote Key Generation (ARKG)	Yes	Yes	Yes	Yes
Pseudo-Random Function (PRF) support				
`MakeCredential` CTAP extension `hmac-secret-mc`	Yes	Yes	Yes	Yes
Enterprise Credential Management, Multiple Environments				
Persistent Credential Management Read Only (PCMR) CTAP `pinUvAuthToken` permission	Yes	Yes	Yes	Yes
Persistent PIN User Access Token (PPUAT)	Yes	Yes	Yes	Yes
Enroll YubiKey through Enroll app (LEA)	Yes	Yes	Yes	Yes
FIDO over Chip Card Interface Device (CCID), USB & smart card SCP11b secure channel	Yes	Yes	Yes	Yes
YubiHSM Auth credential editing	Yes	Yes	–	–

continues on next page

Table 8 – continued from previous page

Firmware 5.8 Capabilities CTAP 2.3, where noted	YubiKey 5 (Std, CCN)	YubiKey 5 Enhanced PIN	Security Key Series	YubiKey Bio Series
Conditional Mediation for Security Keys				
CTAP authenticatorGetI API encCredStoreState	Yes	Yes	Yes	Yes
CTAP authenticatorGetI API encIdentifier	Yes	Yes	Yes	Yes
PIN and User Verification				
CTAP authenticatorGetI API maxPINLength	Yes	Yes	Yes	Multi-Protocol Edition only
CTAP authenticatorGetI API pinComplexityPoli	Yes	Yes	Yes	Yes
CTAP authenticatorGetI API pinComplexityPoli	Yes	Yes	Yes	Yes
CTAP authenticatorGetI API uvCountSinceLastP	–	–	–	Yes
YubiKey Reset Discovery				

continues on next page

Table 8 – continued from previous page

Firmware 5.8 Capabilities CTAP 2.3, where noted	YubiKey 5 (Std, CCN)	YubiKey 5 Enhanced PIN	Security Key Series	YubiKey Bio Series
CTAP `authenticatorGetI` API `longTouchForReset`	Yes	Yes	Yes	Yes
CTAP `authenticatorGetI` API `transportsForRese`	Yes	Yes	Yes	Yes
Enterprise Attestation: RPID storage, Multiple IdPs, Post-Quantum Cryptography (PQC)				
Enterprise Attestation RPIDs from 2 to 16.	Yes	Yes	Enterprise Edition only	Yes
CTAP `authenticatorGetI` API `attestationFormat`	Yes	Yes	Enterprise Edition only	Yes
CTAP `authenticatorMake` API `attestationFormat`	Yes	Yes	Enterprise Edition only	Yes

Note:

1. YubiKey FIPS series remains on the NIST 140-3 validated firmware version 5.7.4. This means that YubiKeys on 5.8 are not FIPS keys, though they share the cryptographic module major functions.
2. Security Key Series - Enterprise Edition and Yubico Bio Series - Multi-Protocol Edition, both require subscription with YubiKey as a Service.
3. ARKG-related `previewSign` code and test utilities are experimental. Do not use this for production cryptographic guidance.

4. As of YubiKey 5.8, YubiHSM Auth credential can be changed without deleting it, then recreating it.

4.2.2 CTAP 2.3 Features

The Client to Authenticator Protocol (CTAP) 2.3 update enhances YubiKey functionality by improving user experience, enterprise scalability, and platform integration. Key benefits include streamlined passkey workflows with Conditional Mediation, seamless PRF-based credential management, support for Secure Payment Confirmation, expanded RPID storage for enterprises, and improved PIN management and security features.

By implementing CTAP 2.3, Yubico addresses key customer challenges across user experience, enterprise credential management, and modern workflows like payments and wallet integration.

These features deliver:

- **Simplified UX**: Seamless passkey workflows, reducing friction for end-users.
- **Enterprise Scalability**: Increased flexibility and ease of management for complex enterprise environments.
- **Future-Ready Innovation**: Support for emerging standards like PRF-based wallets and Secure Payment Confirmation.
- **Improved Security and Integration**: Enhanced platform features and PIN management workflows, ensuring long-term usability and security.
- **Long touch for Reset**: user setting of long touch reset allowing the user to enable long touch for reset if it is disabled by default (in our case it is a prepers only option and off by default). See Long touch for Reset.
- **PIN Complexity Extension**: `pinComplexityPolicy` extension. See PIN Complexity Extension (pinComplexityPolicy).

CTAP 2.3 additional references:

- FIDO Alliance CTAP FIDO2 Specifications for `authenticator` API descriptions.
- Yubico's .NET YubiKey SDK: YubiKey API reference
- Yubico's developer topic, CTAP 2.3: Protocol Refinements and PIN Complexity Improvements.

4.2.3 Attestation Formats

YubiKey Firmware 5.8 with CTAP 2.3 supports a post-quantum enabled ecosystem, `attestationFormatsPreference` option in `authenticatorGetInfo`. This allows greater control and flexibility in determining the capabilities of authenticators. Currently more of a future proofing feature, it allows potential expansion as alternate attestation formats become available.

`attestationFormats` (0x16) communicates supported attestation formats, enabling platforms to prepare for future features like post-quantum cryptography (PQC). Allows us to skip doing attestation signatures that the platform then strips if format = `none`.

See CTAP 2.3 authenticatorGetInfo.

See *Enterprise Attestation with FIDO 2*.

4.2.4 Conditional Mediation for Security Keys

The YubiKey 5.8 contains a number of Client to Authenticator Protocol (CTAP) 2.3 features that can collectively be used to enable conditional mediation for credentials that are stored in security keys, instead of software-based passkey providers. Use cases include:

- Enable platforms to list credentials stored on currently connected hardware security keys during conditional mediation for passkeys.
- Enable platforms to cache credentials, that have been previously used to successfully authenticate to a specific relying party (RP), to list credentials stored on known hardware security keys during conditional mediation for passkeys, without storing identifiers that can be used to identify or correlate security keys.

CTAP APIs include:

- `encCredStoreState`
- `encIdentifier`
- `encIdentifier` (0x19)

By implementing these CTAP 2.3 and YubiKey firmware 5.8 options, platforms can deliver a much smoother user experience, reducing the number of user interactions needed, while mitigating tracking, as well as expanding options for safe credential selection and identification, without granting the ability to delete credentials.

Platforms and applications can enumerate the credentials on a hardware key without risk of deletion and reduce the number of authentication events required, allowing IDPs and Password Managers to deliver a smooth flow between applications and services.

See CTAP authenticatorGetInfo.

encCredStoreState

Introduces an indicator in `GetInfo` that allows the platform to know if the state of the credentials on the key have changed since the last time that they were cached. This allows platforms a method for checking if the credential has changed when they use Persistent PIN User Access Token (PPUAT) to enable autofill and secure payment confirmation flows.

The value is a byte value containing `iv || ct`.

Where:

`ct` is the AES-128-CBC encryption of (128-bit credential store state) using HKDF-SHA-256:

- salt = 32 zero bytes
- IKM = persistentPinUvAuthToken
- L = 16
- info = “encCredStoreState”

`iv` The encryption iv must be regenerated for each output of `GetInfo`

encIdentifier

- Provides a stable, device-unique identifier to a platform but it is always encrypted with a per-device secret (`persistentPinUvAuthToken`) to prevent cross-origin tracking. Each time `GetInfo` is called, the authenticator returns a value that changes. A platform with a valid Persistent PIN User Access Token (PPUAT) can validate this value, confirming its authorization.
- Returned in `GetInfo` as part of support for conditional mediation with hardware authenticators.

encIdentifier (0x19)

The `GetInfo` feature supports privacy-preserving cached credential management, helping platforms identify the correct authenticator without persistent tracking.

4.2.5 Persistent PIN User Access Token (PPUAT)

YubiKey Firmware 5.8 with CTAP 2.3 provides a new type of access token that can be acquired via PIN Protocol v2. This protocol enables the platform to acquire a long term access token that can be used to identify individual credentials on a hardware security key.

`PersistentPinUvAuthToken` (PPUAT) enables platforms to list discoverable credentials from YubiKeys without requiring repeated PIN entry.

This means that users can provide the passkey validation on login and the platform can use the persistent PIN to authenticate the user over multiple clicks and buried menus. Users do not have to constantly re-authenticate each move within a logged in session.

The access token simplifies workflows by delivering an autofill-like experience for YubiKey passkeys, aligning security key usability with password managers and platform credentials.

See CTAP authenticatorClientPIN.

4.2.6 Persistent Credential Management Read Only (PCMR) Permission

YubiKey Firmware 5.8 with CTAP 2.3 provides new permission that can be assigned to a Persistent Credential Management Read Only (PCMR) to enable reading credential information only.

Used in conjunction with PPUATs to grant long-term read-only permission to credentials on an authenticator. `PCMR` is a `pinUvAuthToken` permission that allows platforms to display security key credentials alongside on-platform passkeys, supporting modern Conditional Mediation UX. For example, password manager-like dropdowns.

See CTAP authenticatorCredentialManagement.

4.2.7 FIDO over CCID and USB

YubiKey Firmware 5.8 supports FIDO over Chip Card Interface Device (CCID) allowing CTAP 2.x commands to be carried using ISO7816 Application Protocol Data Unit (APDU) messaging over the USB CCID smart card interface in addition to the USB Human Interface Device (HID) FIDO interface.

This provides a standards-defined alternative transport for CTAP that leverages the OS smart card stack, and enables secure channel establishment (SCP11b) beyond NFC. It adds flexibility and functionality for remote enrollment of devices, including expanding options on the iOS platform.

See *FIDO2 with USB-C or Lightning* and *FIDO over CCID*.

4.2.8 PIN and User Verification

YubiKey Firmware 5.8 with CTAP 2.3 PIN-related `GetInfo` enhancements used to enable the platform to guide the user towards having more success with creating and remembering their PIN. FIDO 2 PIN and user verification feature updates.

These new enhancements allow customers, especially those subject to high-assurance security standards, to mandate and enforce strong PIN policies and authentication frequency events. Enabling these options within the hardware means customers do not have to solely rely on their application platforms to enforce policies.

Platforms can now query the hardware to assess PIN minimum and maximum lengths, whether complexity is enabled and monitor how often the PIN is entered.

When implemented, the platforms can control and modify the UX experience for users as well as ensuring security policies are met and maintained.

See CTAP authenticatorGetInfo.

maxPINLength

Specifies the maximum PIN length (beyond the baseline byte spec limit) the authenticator enforces for ClientPIN, measured in Unicode code points. This enables platforms to validate PIN length client-side and provide accurate user guidance before submission.

Allows platforms to provide users-guidance regarding what is acceptable for a PIN, and better user experience when attempting to set a PIN that will be rejected by the authenticator because it exceeds the maximum length.

Note: Specification constraint. Even if `maxPINLength` is set, the authenticator must still restrict the PIN to 63 bytes or fewer.

`maxPINLength` (0x1D) ensures platforms respect PIN length limits. For example, max 8 for PIV compatibility. This improves user experience with PIN setup.

pinComplexityPolicy

A true/false value that indicates whether an authenticator has additional PIN complexity restrictions besides minimum and maximum PIN lengths.

Allows the platform to inform the user that additional PIN complexity requirements are enforced, so that they are more likely to choose a PIN that complies with the authenticator's PIN policy. May be used by the platform to decide whether to attempt to process the `pinComplexityPolicyURL`, or display a generic message if the platform does not support `pinComplexityPolicyURL`.

`pinComplexityPolicy` (0x1B) enables platforms to know that the authenticator is enforcing stronger PIN policies (reducing weak PIN usage and improving security) and therefore can provide better UX when a user tries to set a PIN that does not meet the complexity requirements.

pinComplexityPolicyURL

Optional URL that allows platforms to provide users with a specific link to PIN complexity rules, ensuring clarity and compliance with authenticator requirements.

Allows the platform to provide a way for the end user to understand what PIN complexity policies are in force for the specific device they are using, to guide the user to setting a PIN that will not be rejected by the authenticator.

`pinComplexityPolicyURL` (0x1C) allows platforms to provide users with a specific link to PIN complexity rules, ensuring clarity and compliance with authenticator requirements. For example, `yubi.co/pin`.

uvCountSinceLastPinEntry

Allows platforms to occasionally prompt PIN entry after repeated biometric usage, preventing PIN forgetfulness. It is defined as a counter: the number of consecutive successful user verification (UV) attempts since the authenticator last required the user to re-enter their PIN.

Enables "PIN recheck" policies. For example, require PIN after *N* biometric uses.

`uvCountSinceLastPinEntry` (0x17) allows platforms to occasionally prompt PIN entry after repeated biometric usage, preventing PIN forgetfulness.

4.2.9 Digital Wallet and Credential Management with PRF

YubiKey Firmware 5.8 with CTAP 2.3 enables Pseudo-Random Function (PRF) during make credential (`hmac-secret-mc`). This allows dual-HMAC keys to be created seamlessly during credential setup instead of requiring two separate steps.

Applications, such as Dashlane, that rely on PRF for end-to-end encrypted synced passkeys, make seamless integration critical. However, setting up credentials that rely on PRF requires multiple user interactions, and creates adoption barriers for lifecycle management of wallets and synced passkeys.

`hmac-secret-mc` extension reduces friction during setup, enabling smooth passkey lifecycle management and digital wallet adoption, keeping YubiKeys relevant in evolving ecosystems.

CTAP references:

- authenticatorClientPIN.
- PersistentPinUvAuthToken.

4.2.10 Pseudo-Random Function (PRF) Support

Defined as a WebAuthn extension (PRF) in WebAuthn Level 3, PRF is used to provide per-credential high-entropy secrets that can be used to derive symmetric encryption keys. The YubiKey provides PRF support via the `hmac-secret` extension, which was introduced in CTAP 2.0 and has been a part of the YubiKey firmware since 2019.

YubiKey Firmware 5.8 with CTAP 2.3 `hmac-secret-mc` extension:

- Expands the PRF functionality with the `hmac-secret-mc` function. This allows developers to generate encryption keys immediately upon account creation and reduce the number of times a user needs to interact with the key. For example, removes the double-tap.
- Developers can deliver smoother flows when they need to create both authentication and encryption keys at the same time.

Use cases:

- Deriving strong symmetric keys: Use PRF output as a user's local encryption key for cloud storage.
- Application-specific keys: Derive one secret for authentication, another for encrypting files, and another for signing transactions (all from the same credential).
- Federated/delegated identity: Payment provider derives its own secret from the merchant's credential, with no shared state needed.

CTAP references:

- hmac-secret.
- hmac-secret-mc.

hmac-secret

A mandatory CTAP extension for authenticators starting in CTAP 2.1.

Provides per-credential, per-RP high-entropy secrets that can be derived during `getAssertion`.

hmac-secret-mc

Extension introduced in CTAP 2.3.

The `hmac-secret-mc` extension allows for the derivation of secrets during `makeCredential`, reducing the number of times that users need to interact with an authenticator to create a credential and then immediately use that credential to generate encryption keys.

4.2.11 Reset-Related Capability Discovery

YubiKey Firmware 5.8 with CTAP 2.3 delivers enhanced reset device protection. Security is improved by allowing platforms to identify what reset options are available on the authenticator and tailor the user flow and messaging to reflect that. For example, reset over a non-wireless connection if only USB is offered as an option. Additionally devices can be delivered with a longer presence check enforced, ensuring the destructive action is intentional.

The `transportsForReset`, `GetInfo` bit field is set in pre-pers, and that value determines which transports can be used for reset. For example, USB, NFC, both, or neither.

See CTAP authenticatorGetInfo.

transportsForReset (0x1A)

Improves reset security by requiring a non-wireless connection, preventing specialized attacks on FIDO authenticators that rely on the ability to reset a device using NFC without the knowledge or consent of the device owner.

Allows the platform to direct the user to connect a device over an allowed transport for reset to be successful. Addressing emerging FIDO2 reset restrictions over NFC.

The default value should be that all available transports (USB, Lightning, NFC) are available for reset, however, there shall exist an interface to turn off interfaces for reset (e.g. not allow reset over NFC). This should be a "pre-pers" option. For example, it cannot be changed once the key has been programmed.

longTouchForReset (0x18)

Improves reset security by requiring deliberate long-touch actions instead of a simple touch. This is currently defined as five seconds so as not to break Windows 10 while still providing evidence of a deliberate action. Default is still for the shorter touch.

Allows the platform to direct the end user to use a gesture other than the default short touch to indicate they intend to reset their device. Without this value, platforms cannot inform users the actions they need to take to successfully reset a security key that requires a long touch to reset.

The default value should be that a short touch is used for reset to not cause customer confusion, however, there shall exist an interface to change to long touch for reset. This should be a "pre-pers" option, that is, it cannot be changed once the key has been programmed.

4.2.12 RPID Storage for Enterprises

YubiKey Firmware 5.8 with CTAP 2.3 increases storage capacity for Enterprise Attestation Relying Party ID (RPID)s from 2 to 16 without affecting certificate storage. It simplifies enterprise testing, deployment, and operations, ensuring smoother credential management across complex environments. This allows customers to:

- Support both test and production environments. For example, `test-tenant.acme.com` and `prod-tenant.acme.com`.
- Support Enterprise Attestation on multiple identity providers.

4.2.13 Secure Payment Confirmation

YubiKey Firmware 5.8 with CTAP 2.3 provides a `ThirdPartyPayment` extension per-credential bit flag enables YubiKeys to be used for cross-domain credentials without redirects, as required by Secure Payment Confirmation (SPC).

Enables YubiKeys to support modern, frictionless payment flows, increasing usability and business adoption for secure payments.

See CTAP thirdPartyPayment.

4.2.14 Signing Extension (Preview)

This feature is currently delivered as a preview option with YubiKey Firmware 5.8 and is likely to change before the final specifications are agreed.

Proposed by Yubico for inclusion in WebAuthn L4, the `previewSign` extension is intended for use cases that need to sign an un-modified (not including client or authenticator data) message with a hardware security key. Like the `thirdPartyPayment` extension, this uses a separate credential which may not be used for normal authentication.

Asynchronous Remote Key Generation (ARKG) with the raw signing, `previewSign` extension, as implemented for the FUNKE EUDI Wallet Prototypes, enables end users to use a YubiKey as a root of trust in a wallet scenario and for additional use cases that require signing of arbitrary data. For example:

- The ability to use hardware security keys to sign raw data with signature schemes that are not suitable for authentication, but are suitable for digital wallet applications.
- Provides hardware-backed ECDSA P-256 signing over application-defined data via the `previewSign` extension. Keys are generated using ARKG; relying parties may integrate ARKG into protocol designs explicitly for key unlinkability or just use the resulting keys like any other ECDSA P-256 key. Additional algorithms or derivation models may be supported in the future.

By extending signing operations beyond just user authentication new use cases are opened up, the signing extension allows for solutions to sign raw data without needing to include client or authenticator details. For example:

- Enables customers to offer comprehensive solution for digital wallets and also for additional use cases where signing arbitrary data is needed.
- By delivering this preview version of the Signing Extension, customers can build out new use cases and solutions ready for full adoption

4.2.15 Third-party payment extension

YubiKey Firmware 5.8 supports CTAP 2.3 Third Party Payment, `thirdPartyPayment`, extension for third-party payment.

See CTAP thirdPartyPayment.

Defined as a WebAuthn extension, `thirdPartyPayment`, can be requested by the RP in `navigator.credentials.create()` or `navigator.credentials.get()` and is supported via CTAP 2.2 if the authenticator advertises it in the `GetInfo` extensions. It is returned in the authenticator responses if the extension is recognized and supported.

`thirdPartyPayment` allows an RP to declare that a credential is being created for a payment use case on behalf of a third party. It provides a privacy-preserving signal that the credential may be scoped or treated differently for payment flows, enabling trusted payment UX without leaking sensitive data.

These extensions focus on giving vendors the tools to be able to present streamlined, secure payment flows to customers, reducing payment friction by removing redirects and confusing UX messages. Historically a credential has been tied to the platform where it was created, meaning merchants would need to redirect users to external sites to perform payment, creating extra steps and confusion.

Use cases:

- Delegated payment credential creation
  - Example: The user creates a passkey with their bank. With the `thirdPartyPayment` extension set to true, the user can now use that bank passkey on a merchant site to validate the transaction with the bank.
  - Authenticator understands this is a "third party payment" credential and can surface appropriate UX. For example, payment icon, branding, and warnings.
- Improved payment UX in browsers/platforms
  - Allows the platform (browser/OS) to display "Use this passkey to pay with Provider X" instead of a generic login prompt.
  - Prevents confusing UX where a payment credential looks identical to a normal website credential.

4.2.16 YubiHSM Auth credential editing

Prior to YubiKey firmware 5.8, after a credential is created in the YubiHSM Auth component it could not be updated. With YubiKey Firmware 5.8 you can update an existing credential if:

- In possession of the authkey, or
- In possession of the current password

This helps in deployment situations when a credential is set to an initial value by for example an admin, then changed later after the end user receives their key.

See Yubico .NET YubiKey SDK: User's Manual, YubiHSM Auth Overview.

4.3 5.7 - 5.6 Firmware Specifics

This section provides detailed descriptions of the features enabled by the 5.7 firmware. Also referenced are features included firmware version 5.6.x.

- *CTAP 2.1 Features*
- *FIDO2 Extensions*
  - *Enterprise Attestation with FIDO 2*
  - *Minimum PIN Length*
  - *ForceChangePIN*
  - *Always Require User Verification*
  - *Blob Storage*
- *FIDO Level 2*
- *PIV Enhancements*
- *PIN Complexity*
- *Expanded Storage (FIDO2 and OATH)*
- *Restricted NFC*
- *Yubico Crypto Library*

4.3.1 5.7.4, 5.7.x, 5.6.x Firmware

Table 9: **Capabilities Available per YubiKey 5 Series Firmware 5.7.x**

CTAP 2.1 Capabilities FW 5.7, unless noted	YubiKey 5 (Std, CCN)	YubiKey 5 FIPS Series	YubiKey 5 Enhanced PIN	Security Key Series	YubiKey Bio Series
PIN Management Flexibility					
Set temporary FIDO2 PIN User must change on next use	Yes	Yes	Yes	Enterprise, non-enterprise	Multi-protocol, FIDO
Configure FIDO2 minimum PIN length	Yes	Yes	Yes	Enterprise, non-enterprise	Multi-protocol, FIDO

continues on next page

Table 9 – continued from previous page

CTAP 2.1 Capabilities FW 5.7, unless noted	YubiKey 5 (Std, CCN)	YubiKey 5 FIPS Series	YubiKey 5 Enhanced PIN	Security Key Series	YubiKey Bio Series
Enhanced PIN complexity	Custom configured keys only	Yes	Yes	Enterprise	Custom configured keys only
Default minimum PIN length	4	8 FW 5.7.4 6 FW 5.4.x	6	6 Enterprise, 4 non-enterprise	6 Multi-protocol 4 FIDO
Enhanced Asset Tracking					
Enterprise Attestation capable Requires custom configuration	Yes	Yes	Yes	Enterprise	Multi-protocol, FIDO
Serial number retrievable by client software in Windows without Admin rights	Yes Also earlier firmware	Yes	Yes	Enterprise	Multi-protocol, FIDO
Enhanced Smart Card Capabilities (PIV)					
RSA-3072 and RSA-4096 support	Yes	Yes	Yes		Multi-protocol, FIDO
Ed25519 and X25519 key types	Yes	Yes	Yes		Multi-protocol, FIDO
Enhanced Credential Storage					
Supports 100 passkeys	Yes	Yes	Yes	Enterprise, non-enterprise	Multi-protocol, FIDO
Supports 24 PIV certificates	Yes	Yes	Yes		Multi-protocol
Supports 64 OATH credentials	Yes	Yes	Yes		

continues on next page

Table 9 – continued from previous page

CTAP 2.1 Capabilities FW 5.7, unless noted	YubiKey 5 (Std, CCN)	YubiKey 5 FIPS Series	YubiKey 5 Enhanced PIN	Security Key Series	YubiKey Bio Series
Supports 2 OTP seeds	Yes	Yes	Yes		

Note: FIPS series is NIST certified with YubiKey version 5.7.4 firmware.

5.7.4 Firmware - FIPS Series

YubiKeys 5 FIPS Series with firmware 5.7.4 is validated and certified for FIPS 140-3 Security Level 2, Physical Level 3 by the Cryptographic Module Validation Program (CMVP) of the National Institute of Standards and Technology (NIST). See Cryptographic Module Validation Program (CMVP) Certificate #5291 and *YubiKey Firmware 5.7.x Changes for FIPS 140-3*.

Yubico's aim in releasing this new firmware is to bring the new enterprise-focused features to users that require FIPS-certified authenticators.

The FIPS 140-3 requirements are very different from those of FIPS 140-2. For a detailed description of those requirements, see *YubiKey Firmware 5.7.x Changes for FIPS 140-3* and *YubiKey 5 FIPS Series Features*.

5.7 and 5.6 Firmware Prior to 5.7.4

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

In addition to the features that are directly accessible, there are a number of features that require partner support.

A couple examples of a feature made available by firmware:

- The NFC function with firmware 5.7 and later is disabled during shipping to prevent tampering. Plugging the YubiKey into the device in activates the NFC function. For more detail on this specific feature, see *Restricted NFC*.
- The 5.7 and later firmware for the YubiKey 5 Series has a number of features that are available for the multi-protocol YubiKey 5.

4.3.2 CTAP 2.1 Features

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 to Sites or Services*) and `uvRetries` introduced in 5.5.0.

In firmware 5.7 and later 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.

4.3.3 FIDO2 Extensions

Enterprise Attestation

As of YubiKey Firmware 5.7.4, 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.

See *Enterprise Attestation with FIDO 2*.

Minimum PIN Length and Minimum PIN Length Extension

In YubiKey firmware version 5.7.4, the `minPINLength` extension resolves compliance requirements for organizations that need to enforce use of specific PIN lengths.

Prior to 5.7, enforcing PIN length was only possible with:

- YubiKey 5 FIPS Series authenticators certified for FIPS 140-2, which have a minimum PIN length of 6.
- YubiKey 5 FIPS Series Enhanced PIN and Security Key Series have minimum PIN length set to 6 by default.
- Custom configuration, which RPs could do when the authenticator had a custom AAGUID.

See *Minimum PIN Length*.

Always Require User Verification

Always Require User Verification (UV), `alwaysuv` was introduced to prompt users for user verification (UV) each time. This ensures consistent behavior between different platforms and RPs.

This feature is enabled by default for:

- YubiKey Bio Series
- YubiKey Series 5 Enhanced PIN

See *Always Require User Verification*.

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.

Large blob storage is:

- 4096 bytes on firmware 5.7.0 and later
- 1024 bytes on firmware 5.5.x and 5.6.x

See *Blob Storage*, *Credential Blob*, *Large Blob*.

4.3.4 FIDO Level 2

As of YubiKey Firmware 5.7.4, all YubiKeys with firmware version 5.7 and later have achieved FIDO Level 2 certification for assurance of attestable hardware-bound credentials. Certification enables YubiKeys for use with e-government use cases (citizen-facing) and corporate compliance mandates that require FIDO L2 certification.

To check the FIDO certification status for all keys and firmware versions see YubiKey hardware FIDO2 AAGUIDs.

4.3.5 PIV Enhancements

Additional Key Types Supported

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

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

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 and later 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.

Beginning with firmware 5.4.x, the management key type held in PIV slot 9b expanded 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.

To summarize, standard YubiKey 5 Series keys with firmware 5.7.x and later use AES-192 for the management key by default. TDES, along with AES-128 and AES-256, are supported as options. YubiKey 5 FIPS Series keys with firmware 5.7.x and later allow AES only, with AES-192 as the default.

YubiKeys with firmware 5.4.x through 5.6.x use TDES for the management key by default, and AES-128, AES-192, and AES-256 are supported as options.

YubiKeys with firmware 5.3.x and older support TDES only.

For additional technical information, see **PIV AES Management Key** in *Smart Card - PIV Compatible Specifics*.

Advanced Key Management Functions

With the 5.7.x and later 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

As of YubiKey Firmware 5.7.4, 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

As of YubiKey Firmware 5.7.4, 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 and are also supported for 5.7 and later firmware.

Table 10: **List of Tags for Import Data**

Algorithms	Key Element	Tag
RSA-1024 (0x06) RSA-2048 (0x07)	prime P	0x01
RSA-3072 (0x05)* RSA-4096 (0x16)*	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

As of YubiKey Firmware 5.7.4, keys can be moved from any slot except F9 (attestation) to any other slot except F9 using the instruction 0xF6.

Table 11: **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

As of YubiKey Firmware 5.7.4, any key can be deleted from any slot, including F9 (Attestation) using the instruction `0xF6` with a value of `0xFF` for P1.

Table 12: **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

YubiKey PIV Metadata

YubiKey 5 PIV metadata enables services and client software to obtain information about PIV keys from a central location, which means, as of YubiKey Firmware 5.7.4, 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 was already available starting with the 5.3.0 firmware. For details, see the Get Metadata section of the PIV extensions.

4.3.6 PIN Complexity

PIN complexity enforces common PIN rules. It has to have a minimum 6 characters. It cannot use repeated sequential characters. It cannot be one of the commonly used and therefore easily guessed PINs. See the list below, *Blocked PINs*.

As of YubiKey Firmware 5.7.4, the PIN complexity feature prevents users from adopting simple patterns or common PINs. Blocking these type of PINs significantly reduces the risk of users setting easily guessable PINs on their devices.

PIN complexity is available on some YubiKeys with firmware version 5.7.0 and later. For more details on feature support across the various YubiKey series, see the *Capabilities Available per YubiKey 5 Series Firmware 5.7.x*.

PIN complexity is enabled by default and cannot be disabled on the following series:

- Security Key Series - Enterprise Edition. Requires subscription with YubiKey as a Service. See *Security Key Series*.
- YubiKey 5 Enhanced PIN Series. Requires subscription with YubiKey as a Service. See *YubiKey 5 Enhanced PIN Series*.

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 all follow the same set of rules.
- For the protocols on the YubiKey, PIN complexity is applied to the listed PIN type:
  - FIDO2 - PIN
  - PIV - PIN and PUK
  - OpenPGP - user PIN, admin PIN, and reset code
  - yubihsm-auth - credential PINs
  - YubiKey - access codes

Unicode characters

In firmware 5.7 and later, 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.

Blocked PINs

PINs are blocked, that is cannot be used, if they have any of the following:

- Are less than 6 characters
- Contain only one unicode character, for example: `111111`
- Are on the blocklist (commonly used PINs that are easily guessed):
  - 123456
  - 123123
  - 654321
  - 123321
  - 112233

- 121212
- 123456789
- password
- qwerty
- 12345678
- 1234567
- 520520
- 123654
- 1234567890
- 159753
- qwerty123
- abc123
- password1
- iloveyou
- 1q2w3e4r

4.3.7 Expanded Storage (FIDO2 and OATH)

As of YubiKey Firmware 5.7.4, 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

4.3.8 Restricted NFC

Restricted NFC mode prevents wireless device manipulation before a YubiKey NFC with the 5.7 and later 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 and later 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 and later NFC causes 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 ykman config nfc.

4.3.9 Yubico Crypto Library

As of YubiKey Firmware 5.7.4, and now available, is a library Yubico has developed over the past few years in-house that performs the underlying cryptographic operations (encryption, signing, etc.) for RSA and ECC.

4.3.10 Yubico Root Certificate Authority (CA)

Yubico's root certificate authority (CA) was updated in early 2025. Starting with firmware 5.7.4, all YubiKeys will be signed by the new root CA.

4.4 Pre-5.6 Firmware Specifics

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

4.4.1 Firmware 5.4.3: YubiHSM Auth

YubiHSM Auth is a CCID application that can store long-lived credentials (AES keys) that are used to establish secure sessions to a YubiHSM 2. By providing an external challenge, a derivation scheme that yields three session keys is executed. The session keys are not stored on the YubiKey but simply output as a result. The session keys can be used for authentication to the YubiHSM 2. The authentication scheme is based on SCP03 (see *Firmware 5.3.0: Secure Channel* above). Each long-lived YubiHSM Auth credential is protected by a user access code that has to be provided to authenticate each session. Storing and deleting credentials requires a separate admin access code.

For detailed information, see *YubiHSM Auth Specifics*.

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.

4.4.2 Firmware 5.3.0: Secure Channel

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

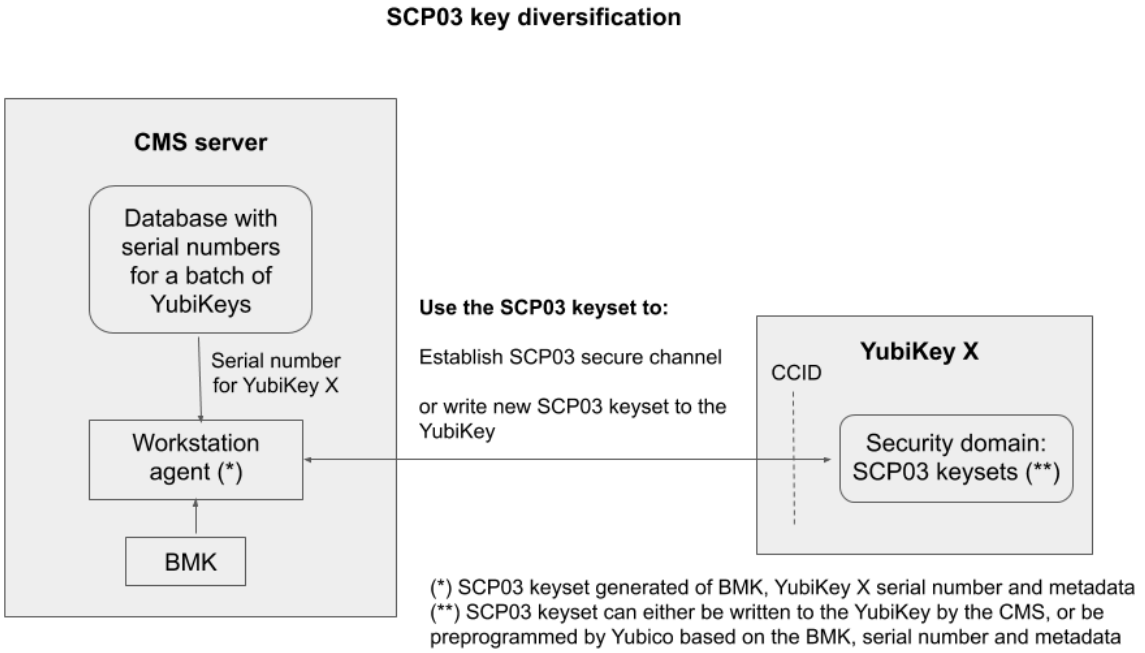
SCP03 (Secure Channel Protocol 03), which is part of the GlobalPlatform standard, is a framework for mutual authentication and encrypted transport between hosts and secure elements in smart cards. This protocol for secure channel is implemented on YubiKeys as of Yubico 5.3.0 firmware.

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.

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*.

4.4.3 Firmware 5.3.0: NFC ID Calculation Changed

Crucial to vendors of physical access control systems and door protection systems utilizing NFC readers, the modification of the YubiKey NFC ID calculation enables NFC readers and access management systems (door locks) using the NFC ID tag to identify NFC-enabled YubiKeys, including those without serial numbers. It is now calculated so that a unique string is returned in the first part of the NFC ID. Until the release of the 5.3.0 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*.

CHAPTER

FIVE

YUBIKEY 5 FIPS SERIES SPECIFICS

These topics describe YubiKey 5 FIPS series support for FIPS 140-2 through FIPS 140-3.

5.1 YubiKey 5 FIPS Series Features

NIST classified the YubiKey 5 Series FIPS as "composite authenticators". As such, no device in that series can be taken out of the FIPS-approved mode after initialization without zeroizing the function. This means that once the YubiKey is correctly configured, it remains in the correct configuration. As long as the crypto officer ensures that the YubiKey 5 Series FIPS devices are correctly configured at initialization, they remain in FIPS-approved mode. See *Deploying the YubiKey 5 FIPS Series*.

Initialization is a bit different depending on the FIPS version you are using. Each version requires more stringent initialization than the previous level. See *YubiKey Firmware 5.7.x Changes for FIPS 140-3*, *FIPS 140-2 Level 2 Configuration*, or *FIPS 140-2 Level 1 Configuration*.

5.1.1 NIST Version Certification

The YubiKey 5 cryptographic module applies across YubiKey firmware versions.

- YubiKey 5 FIPS Series firmware version 5.7.4 is is certified as an authenticator under FIPS 140-3, Level 2.
- YubiKey 5 FIPS Series hardware with 5.4.3 and 5.4.2 firmware is certified as an authenticator under FIPS 140-2, both Level 1 and Level 2 (Physical Security Level 3).

The YubiKey 5 FIPS Series meets the highest authenticator assurance Level 3 (AAL3) of NIST SP800-63B guidance.

Note:

- Effective May 22 2026, YubiKey 5 FIPS Series firmware 5.7.4 was NIST certified for FIPS 140-3.
- YubiKey 5 FIPS Series 5.4.x FIPS 140-2 certification moved to the Sunset List, May 2026 and will be moved to the Historical List, September 22, 2026.

Per NIST document FIPS 140-3 Transition Effort, YubiKey 5 FIPS Series certification is affected as follows:

- Existing deployments - No change. Authenticators certified under FIPS 140-2 can continue to be used in existing deployments after sunset.
- New deployments - FIPS 140-2 is being replaced by FIPS 140-3. New submissions must fulfill the FIPS 140-3 requirements. Authenticators deployed in new environments must have a valid certificate that is not expired. Auditors typically recommend that authenticators certified under FIPS 140-2 are not used in new deployments.

5.1.2 Supported Cryptographic Module Major Functions

The following table lists the FIPS cryptographic modules and protocols supported for the YubiKey 5 FIPS Series firmware versions.

Table 1: **Cryptographic Functions and NIST Version per Firmware Version**

Function	Firmware Versions 5.7.4	5.4.3	5.4.2
FIDO Universal 2nd Factor (U2F)	•	•	•
FIDO2 WebAuthn	140-3	140-2	140-2
OATH OTP authentication	140-3	140-2	140-2
One Time Password (Yubico OTP)	•	140-2	140-2
OpenPGP (version 3.4)	140-3	140-2	•
PIV-compatible smart card	140-3	140-2	140-2
SCP03	140-3	140-2	140-2
SCP11	140-3	•	•
YubiHSM Auth	140-3	140-2	•

Note:

1. Do not use U2F with FIPS. Instead use the FIDO2. U2F is disabled on FIPS 140-3 keys. Be sure to disable U2F on FIPS 140-2 keys. See *U2F with FIPS*.
2. For Yubico OTP, YubiKeys are labeled, enabled, and usable on FIPS 140-3 devices. However, it is not certified as FIPS.

5.1.3 FIPS Form Factors

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. These FIPS YubiKey values apply since YubiKey firmware 5.4.

Table 2: **FIPS Form Factor Byte Values**

YubiKey	Undefined	5 NFC	5 Nano	5C NFC	5C Nano	5Ci
Connections	N/A	NFC + USB-A	Nano USB-A	NFC + USB-C	Nano USB-C	Lightning + USB-C
Value	0x00	0x01	0x02	0x03	0x04	0x05
FIPS Value	N/A	0x81	0x82	0x83	0x84	0x85

5.1.4 Credentials and Permitted Values

For FIPS 140-3 Level 2 and FIPS 140-2 Level 2, the table below lists the credentials required, allowed values, and credential owner for the supported applications.

Table 3: **Credentials Required and Allowed**

Application	Credential	Credential Owner	Permitted Values FIPS	
			140-3	140-2
OATH	Authentication Key	Crypto Officer	14-64 byte HMAC SHA1, SHA256 key	
One Time Password (OTP)	OTP Slot 1 OTP Slot 2	Crypto Officer		6 byte access codes
OpenPGP	User Password (PW1)	Authenticated User	8-127 byte PIN	6-127 byte PIN
	Admin Password (PW3)	Crypto Officer	8-127 byte PIN	6-127 byte PIN
PIV Smart Card	Management Key	Crypto Officer	AES-128, AES-192, AES-256	3-key TDES, AES-128,192, AES-256 key
	PUK	Authenticated User	8 byte PIN	6-8 byte PIN

continues on next page

Table 3 – continued from previous page

Application	Credential	Credential Owner	Permitted Values FIPS 140-3	Permitted Values FIPS 140-2
	PIN	Authenticated User	8 byte PIN	6-8 byte PIN
WebAuthn	PIN	Authenticated User	8-63 byte PIN	6-32 byte PIN

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

- *FIDO Configuration with FIPS* (WebAuthn)
- *OTP Configuration with FIPS*
- *OATH Configuration with FIPS*
- *OpenPGP Specifics*
- *PIV Configuration with FIPS*

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

5.2 Deploying the YubiKey 5 FIPS Series

The YubiKey 5 FIPS Series chipset firmware 5.7.4 is certified under FIPS 140-3 Overall Level 2, Physical Security Level 3.

Physical Security Level 3 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.

See NIST SP 800-63-B for guidance on the level required for your deployment.

5.2.1 New and Existing FIPS Deployments

New FIPS compliant submissions must fulfill the FIPS 140-3 requirements. Use YubiKey 5 FIPS Series firmware 5.7.4.

Existing deployments can continue with YubiKey 5 FIPS Series keys firmware 5.4.2 and 5.4.3, certified under FIPS 140-2 Level 1 and FIPS 140-2 Level 2

Note: YubiKey 5 FIPS Series 5.4.x FIPS 140-2 certification moved to the Sunset List, May 2026 and will be moved to the Historical List, September 22, 2026. Existing FIPS 140-2 deployments can continue after sunset.

The YubiKey 5 FIPS Series keys firmware 5.4.2 and 5.4.3, certified under FIPS 140-2 Level 1 and FIPS 140-2 Level 2, have two certificates, each corresponding to a different level of certification, but both certificates apply to the same keys.

Depending on which certification the YubiKey 5 FIPS Series is being deployed under, there are different requirements and initialization steps for securing the various functions. Each version requires more stringent initialization than the previous level. See the topic for the FIPS version you are using:

- *YubiKey Firmware 5.7.x Changes for FIPS 140-3* and *FIPS 140-3 Configuration*
- *FIPS 140-2 Level 2 Configuration*
- *FIPS 140-2 Level 1 Configuration*

5.3 YubiKey Firmware 5.4 Changes for FIPS

For a description of the FIPS-specific aspects of the YubiKey 5 FIPS Series with the 5.7.4 firmware, see *YubiKey Firmware 5.7.x Changes for FIPS 140-3*.

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 4: **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 YubiKey hardware FIDO2 AAGUIDs
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 certificates applicable to the specific authenticator. See *FIDO Authenticator Supported Certifications*.
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 *FIPS Form Factors*.

5.4 YubiKey Firmware 5.7.x Changes for FIPS 140-3

Important: YubiKeys are shipped with FIPS approved mode OFF. The protocols and applications listed for FIPS 140-3 cannot create credentials until the YubiKeys are put in FIPS approved mode. See *Initializing the Approved Mode*.

The YubiKey 5 FIPS Series 5.7.4 firmware includes a number of changes for FIPS 140-3. The most notable of these changes is that the FIPS-specific requirements are now enforced by the YubiKey.

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

Note: FIPS 140-2 is a regulation that was deprecated in May 2026. Auditors typically recommend that authenticators certified under FIPS 140-2 not be deployed in new deployments. They can continue to be used in existing deployments, but authenticators deployed in new environments must have a valid certificate that is not expired. FIPS 140-3 replaces FIPS 140-2, therefore any new submissions must fulfill the FIPS 140-3 requirements.

5.4.1 High-Level FIPS 140-3 Changes to YubiKey

- **Hardware-Enforced Compliance**: The most notable change in FIPS 140-3 is that FIPS-specific requirements are now strictly enforced by the YubiKey itself.
- **No Pre-Validation Credential Creation**: The device explicitly refuses to create credentials across any application until it is securely configured and in FIPS Approved Mode.
- **Stricter PIN Requirements**: FIDO2, PIV, and OpenPGP now strictly enforce an 8-character minimum PIN length. PIN complexity is enabled by default to adhere to NIST SP800-63b guidelines.
- **Larger Key Sizes**: To meet enterprise and Department of Defense (DoD) demands, the YubiKey supports larger key sizes to provide superior protection until Post-Quantum Cryptography matures.

For FIPS 140-3 configuration changes and additional FIPS Approved Mode information, see *FIPS 140-3 Configuration*.

5.4.2 FIDO2 Changes for FIPS 140-3

- Requires a FIDO2 PIN of at least 8 characters.
- Always Require User Verification (UV), `alwaysUV`, is permanently enabled and strictly enforced for operations.
- U2F functionality is disabled on FIPS-capable devices under the 140-3 standard. Use FIDO2 functions instead.
- PIN Protocol v2 must be over NFC.
- Requires the application is in FIPS Approved Mode to create credentials. The device refuses to create credentials until it is in FIPS Approved Mode.

See *FIDO Initializing Approved Mode*.

5.4.3 OATH Changes for FIPS 140-3

- Requires an access code of at least 14 bytes.
- Executing `SET CODE` and `PUT` commands over NFC, requires a secure channel (SCP03 or SCP11).
- Requires the application is in FIPS Approved Mode to create credentials. The device refuses to create credentials until it is in FIPS Approved Mode.

See *OATH Initializing Approved Mode*.

5.4.4 Open PGP Changes for FIPS 140-3

- Requires changing the default User PIN, Admin PIN, and Reset Code (if set) to a minimum of 8 characters. Changing user PIN, admin PIN or Reset Code to a value shorter than 8 characters is blocked.
- RSA decryption, X25519, and SECP256k1 are blocked.
- Requires all operations over NFC go through a secure channel (SCP03 or SCP11).
- Requires the application is in FIPS Approved Mode to create credentials. The device refuses to create credentials until it is in FIPS Approved Mode.

See *OpenPGP Initializing Approved Mode*.

5.4.5 PIV Changes for FIPS 140-3

- Requires changing the default PIN and PUK to 8-character values. Requires changing the Management Key to an AES key.
- RSA1024, TDES (3DES), and X25519 are blocked algorithms. Cannot set the Management Key to TDES.
- Requires all operations performed over NFC go through a secure channel (SCP03 or SCP11).
- Requires the application is in FIPS Approved Mode to create credentials. The device refuses to create credentials until it is in FIPS Approved Mode.

See *PIV Initializing Approved Mode*.

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

- Requires changing the default key set.
- Requires the application is in FIPS Approved Mode to create credentials.

  Until the application is in FIPS 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.

See *Security Domain Initializing Approved Mode*.

5.4.7 YubiHSM Auth Changes for FIPS 140-3

- Requires changing the default admin code.
- Requires all operations performed over NFC go through a secure channel (SCP03 or SCP11).
- Requires the application is in FIPS Approved Mode to create credentials. The device refuses to create credentials until it is in FIPS Approved Mode.

See *HSMAuth Initializing Approved Mode*

5.5 FIPS Version Changes and Requirements Comparison

The table highlights changes between FIPS 140-3 Level 3, 140-2 Level 2, and 140-2 Level 1.

Each newer version and level of the FIPS 140 includes the requirements of the previous version and level. Therefore each left most column lists only the differences between the next column. For example, FIPS 140-3 Level 2 includes requirements of FIPS 140-2 Level 2 and FIPS 140-2 Level 1. FIPS 140-2 Level 2 includes requirements of FIPS 140-2 Level 1.

Table 5: **YubiKey Functions in FIPS**

YubiKey Function	FIPS 140-3 Level 2 Firmware 5.7.4	FIPS 140-2 Level 2 Firmware 5.4.3	FIPS 140-2 Level 1 Firmware 5.4.2
FIDO2	- FIDO2 PIN required with minimum 8 characters - alwaysUV permanently enabled - U2F disabled on FIPS-capable devices. Use FIDO2 functions instead. - PIN Protocol v2 required over NFC. - Requires application be in FIPS Approved Mode to create credentials	- Set a PIN. - Set Credential Protection to level 2 for all discoverable credentials. - Credential Registration is not allowed over NFC.	No additional requirements

continues on next page

Table 5 – continued from previous page

YubiKey Function	FIPS 140-3 Level 2 Firmware 5.7.4	FIPS 140-2 Level 2 Firmware 5.4.3	FIPS 140-2 Level 1 Firmware 5.4.2
OATH	- Access code required with minimum 14 bytes. - Configuration over NFC requires `SET CODE`, `PUT` commands go through a secure channel (SCP03 or SCP11) - Requires application be in FIPS Approved Mode to create credentials	- 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.	If writing a credential over NFC, use a secure channel.
OpenPGP	- User PIN, Admin PIN, and Reset Code (if set) must be minimum of 8 characters - RSA decryption, X25519 and SECP256k1 are blocked. - All operations over NFC must go through a secure channel (SCP03 or SCP11) - Requires application be in FIPS Approved Mode to create credentials	No additional requirements	No additional requirements

continues on next page

Table 5 – continued from previous page

YubiKey Function	FIPS 140-3 Level 2 Firmware 5.7.4	FIPS 140-2 Level 2 Firmware 5.4.3	FIPS 140-2 Level 1 Firmware 5.4.2
OTP Touch-Triggered	Not validated for FIPS 140-3	- Set Access code for both OTP slots. - If updating a configuration of either OTP slot or the NDEF behavior, use a secure channel.	If writing a configuration to a slot over NFC, use a secure channel.
PIV	- Requires changing default PIN and PUK to 8 character minimum. - Requires changing Management Key to AES key. - RSA1024, TDES (3DES), and X25519 are blocked. - Cannot set the Management Key to TDES. - All operations over NFC must go through a secure channel (SCP03 or SCP11) - Requires application be in FIPS Approved Mode to create credentials	- Change Management key, PIN, PUK from default values. - For any operation with the PIV function over NFC, use a secure channel.	If importing a key or setting the management key, use a secure channel.

continues on next page

Table 5 – continued from previous page

YubiKey Function	FIPS 140-3 Level 2 Firmware 5.7.4	FIPS 140-2 Level 2 Firmware 5.4.3	FIPS 140-2 Level 1 Firmware 5.4.2
Secure Domain Channel	- SCP03 and SCP11 - Requires changing default key set - USB restriction: Until the application is in FIPS Approved Mode, the default key set can only be used to establish a secure channel with the Security Domain itself, only for the purpose of loading a new key set, and this operation must be performed exclusively over USB.	- SCP03 only - Requires changing default key set	- SCP03 only - Requires changing default key set
U2F	U2F disabled on FIPS-capable devices	- Must be not be used. - Recommendation: Disable and use the FIDO2 function instead.	No additional requirements
YubiHSM Auth	- Requires changing default admin code - All operations over NFC must go through a secure channel (SCP03 or SCP11) - Requires application be in FIPS Approved Mode to create credentials	No additional requirements	No additional requirements

5.6 FIPS 140-3 Configuration

The YubiKey 5 Cryptographic Module is a single-chip installed within the YubiKey 5. To configure the YubiKey 5 Cryptographic Module for FIPS 140-3 approved use, follow the instructions below.

PIV, OpenPGP, OATH, FIDO, Security Domain (SD) and HSMAuth functional units must be in FIPS Approved mode for operational use.

To initialize the module in Approved mode, use `ykman` CLI.

- Download and install the latest version of `ykman`, see ykman releases CLI (yubikey-manager).

  We recommend, for GUI access, you use the Yubico Authenticator. The YubiKey Manager GUI is end-of-life.

  Note: Do not use the version of `ykman` installed with the YubiKey Manager GUI, it included an older version of the `ykman` CLI.

- See YubiKey Manager (ykman) CLI User Guide and Yubico Authenticator User Guide.

Note:

- All CLI examples provided below assume either a new or freshly reset device.
- All access codes and keys used are the default values.
- All examples are run from the command line.

5.6.1 Initializing the Approved Mode

The following topics describe how to set the supported protocols into FIPS approved mode.

FIDO Initializing Approved Mode

Requirements

Putting a FIDO2 application in FIPS Approved Mode, requires that you:

- Set the FIDO2 PIN

Set a PIN

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

Example

```
ykman fido access change-pin --new-pin 32145699
```

HSMAuth Initializing Approved Mode

Requirements

Putting a YubiHSM Auth application in FIPS Approved Mode, requires that you:

- Change the default admin code.

Change the default management key

```
ykman hsmauth access change-management-key -m <MGMT-KEY> -n <new-MGMT-KEY>
```

Example

```
ykman hsmauth access change-management-key -m "00000000000000000000000000000000" -n
↪"59e48ecde5a5aeeb3dd2be861ee198a8"
```

OATH Initializing Approved Mode

Requirements

Putting an OATH application in FIPS Approved Mode, requires that you:

- Set the access code
- Set the access code to at least 14 bytes.

Set an authentication key

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

Example

```
ykman oath access change -n 32145699
```

OpenPGP Initializing Approved Mode

Requirements

Putting an OpenPGP application in FIPS Approved Mode, requires that you:

- Change the default user PIN to a minimum 8 character value.
- Change the default admin PIN to a minimum 8 character value.
- If the Reset Code is set, ensure it is at least 8 characters.

Change the default user PIN

```
ykman openpgp access change-pin -P <PIN> -n <new-PIN>
```

Example

```
ykman openpgp access change-pin -P 123456 -n 32145699
```

Change the default admin PIN

```
ykman openpgp access change-admin-pin -a <admin-PIN> -n <new-admin-PIN>
```

Example

```
ykman openpgp access change-admin-pin -a 12345678 -n 32145699
```

PIV Initializing Approved Mode

Requirements

Putting a PIV application in FIPS Approved Mode, requires that you:

- Change the default PIN to an 8 character value.
- Change the default PUK and ensure it is an 8 character value.
- Change the default Management Key to an AES key.

Change the default PIN

```
ykman piv access change-pin -P <PIN> -n <new-PIN>
```

Example

```
ykman piv access change-pin -P 123456 -n 32145699
```

Change the default PUK

```
ykman piv access change-puk -p <PUK> -n <new-PUK>
```

Example

```
ykman piv access change-puk -p 12345678 -n 32145699
```

Change the default management key

```
ykman piv access change-management-key -m <MGMT> -n <new-MGMT> -a AES192
```

Example

```
ykman piv access change-management-key -m␣
↪010203040506070801020304050607080102030405060708 -n␣
↪b0bba5c8f76297f680a4731b200fcb6afb8052c34a42fbf1 -a AES192
```

Security Domain Initializing Approved Mode

The following applies to Security Domains: SCP03 and SCP11.

Requirements

Putting a Security Domain application in FIPS Approved Mode, requires that you:

- Change the default key set.

Change the default key set

```
ykman --scp <KEY1:KEY2:KEY3> sd keys import 0x01 2 <new-KEY1:new-KEY2:new-KEY3>
```

Example

```
ykman --scp␣
↪404142434445464748494a4b4c4d4e4f:404142434445464748494a4b4c4d4e4f:404142434445464748494a4b4c4d4e4f␣
↪sd keys import 0x01 2␣
↪f4e4d4c4b4a4948474645444342414 04:f4e4d4c4b4a4948474645444342414 04:f4e4d4c4b4a4948474645444342414 04
```

5.6.2 Zeroisation Procedure

To zeroise the module, use the following commands.

FIDO Zeroisation

```
ykman fido reset --force
```

Note: When using the `--force` flag, the command must be run immediately after inserting the YubiKey.

HSMAuth Zeroisation

```
ykman hsmauth reset --force
```

OATH Zeroisation

```
ykman oath reset --force
```

OpenPGP Zeroisation

```
ykman openpgp reset --force
```

PIV Zeroisation

```
ykman piv reset --force
```

Security Domain Zeroisation

```
ykman sd reset --force
```

5.7 FIPS 140-2 Level 2 Configuration

The YubiKey 5 FIPS 140-2 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.

5.7.1 FIPS 140-2 Level 2 Configuration

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).)

FIPS Approved Mode

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 *Yubico 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 **ykman** CLI. Install the most recent version of the `ykman`. See ykman releases.

- Download and install the latest version of YubiKey Manager (ykman), from ykman Releases.
- Reference the YubiKey Manager (ykman) CLI Guide.

  For GUI access, we recommend you use the Yubico Authenticator. The YubiKey Manager GUI is end of life.

  OTP, OATH, and WebAuthn
  : To be in a FIPS-approved mode, the OTP, OATH, and FIDO2 (WebAuthn) applications must have their respective credentials set.

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

  U2F
  : 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.

5.8 FIPS 140-2 Level 1 Configuration

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

5.9 FIDO Configuration with FIPS

YubiKey 5 FIPS Series support FIDO U2F and FIDO2 WebAuthn.

5.9.1 FIDO2 (WebAuthn)

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* and *FIDO2 Features*.

FIDO2 (WebAuthn) 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 ykman CLI with the following 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.

5.9.2 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.

U2F with FIPS

The YubiKey 5 U2F FIPS application is disabled for FIPS 140-3 devices and cannot be used in FIPS 140-2 Level 2.

For FIPS 140-2 devices, 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.

5.10 OATH Configuration with FIPS

The YubiKey 5 FIPS OATH application can store up to 64 OATH credentials for FIPS 140-3 and 32 OATH credentials for FIPS 140-2, 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.

See also, *OATH Specifics*.

5.10.1 OATH FIPS-approved Mode

This OATH FIPS approved mode supported for both FIPS 140-3 and FIPS 140-2.

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 ykman CLI.

- Download and install the latest version of YubiKey Manager (ykman), from ykman Releases.
- Reference the YubiKey Manager (ykman) CLI Guide.

  For GUI access, we recommend you use the Yubico Authenticator. The YubiKey Manager GUI is end of life.

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

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

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

5.11 OTP Configuration with FIPS

Note: OTP content in this section applies to FIPS 140-2 devices. FIPS 140-3 devices are not validated for OTP.

The OTP application provides two programmable slots, each of which can hold one of the types of credentials. For information about YubiKey 5 series details with OTP, see *OTP Specifics*.

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

5.11.1 Testing OTP Credentials

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.

5.11.2 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.

5.11.3 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.

OTP FIPS-approved Mode with FIPS 140-2 Level 2

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.

Set Access Codes

The crypto officer can set an access code to the OTP slots using `ykman`, see ykman releases.

- Download and install the latest version of YubiKey Manager (ykman), from ykman Releases.
- Reference the YubiKey Manager (ykman) CLI Guide.

  For GUI access, we recommend you use the Yubico Authenticator. The YubiKey Manager GUI is end of life.

Applying an Access Code

To apply an access code to a configuration using the ykman 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 ykman CLI, see the ykman CLI and YubiKey Manager GUI Guide, OTP section.

Filling a Blank OTP Slot

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.

5.12 PIV Configuration with FIPS

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.

For full description of YubiKey 5 with PIV, see *Smart Card - PIV Compatible Specifics*.

5.12.1 FIPS Exceptions

For YubiKey 5 FIPS Series, some exceptions apply:

- Do not use non-NIST-approved curves
- Do not use the following keys:

  FIPS 140-3

  - RSA 1024-bit
  - Triple DES
  - X25519

  FIPS 140-2

  - RSA 1024-bit
  - 3072-bit keys.

  These restrictions apply to Attestation as well.
- PIN policy = `none` cannot be used. Select either `once` or `always`.

5.12.2 Supported Algorithms

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

Table 6: **Supported PIV Algorithms per FIPS Version**

Algorithm (Identifier)	FIPS Version
RSA-1024 (0x06)	140-2
RSA-2048 (0x07)	140-2
RSA-3072 (0x05)	140-3
RSA-4096 (0x16)	140-3
ECC P-256 (0x11)	140-2
ECC P-384 (0x14)	140-2
Ed25519/x25519 (0xe0)	–

5.12.3 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`)

5.12.4 PIV FIPS Approved Mode

This PIV FIPS approved mode supported for both FIPS 140-3 and FIPS 140-2.

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 and the 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. See YubiKey Minidriver User Guide.

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 ykman CLI.

- Download and install the latest version of YubiKey Manager (ykman), from ykman Releases.
- Reference the YubiKey Manager (ykman) CLI Guide.

  For GUI access, we recommend you use the Yubico Authenticator. The YubiKey Manager GUI is end of life.

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:

- FIPS 140-3: AES-128, AES-192 or AES-256.
- FIPS 140-2: 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.

CHAPTER

SIX

YUBIKEY BIO SERIES SPECIFICS

6.1 Using the YubiKey Bio Series

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

- *How the YubiKey Bio Works*
- Descriptions of the different *User Experiences* with the various protocols
- Full instructions for enrolling fingerprints using platform support:
  - *Enrolling Fingerprints on Chrome* and
  - *Enrolling Fingerprints on Windows*
- Descriptions of the methods by which the protocols are supported, in:
  - *YubiKey Bio and FIDO2*
  - *YubiKey Bio and FIDO U2F*
  - *YubiKey Bio and PIV*
- A brief explanation of the role the Yubico Authenticator plays in managing the YubiKey Bio.

To simply add a PIN and fingerprint to your YubiKey Bio Series, without a lot of details, see Yubico's setup page, scroll down the page to the YubiKey Bio note and click the *enroll your fingerprint* link. **Be sure to add a PIN before you add your fingerprints**.

6.1.1 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.

6.2 Physical Attributes

For physical attributes that apply to all YubiKeys, see *Physical Attributes* and the Yubico data sheet, What is a YubiKey? Product specification.

6.2.1 Sensors

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

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

6.2.2 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.

6.2.3 USB Connector

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.

6.3 Protocols Supported

6.3.1 FIDO Protocols

All keys in the YubiKey Bio Series support WebAuthn sites and applications that support the FIDO2 and FIDO U2F protocols. This includes the YubiKey Bio.

- FIDO2 (WebAuthn) – See *YubiKey Bio and FIDO2*
- FIDO U2F – See *YubiKey Bio and FIDO U2F*

Each application can be enabled and disabled independently. However, 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*.

6.3.2 PIV Protocol

The YubiKey Bio Multi-protocol Edition supports the PIV protocol and works with sites and applications that support PIV/smart card interfaces. See *YubiKey Bio and PIV*.

6.4 Requirements: Platform and Browser Compatibility

6.4.1 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 do not tell you when the YubiKey has failed to match the fingerprint. To identify a failed authentication, check the YubiKey. If its LED is blinking amber, the biometric authentication attempt failed. This is not a problem on Windows 11.

Browsers such as Firefox and Safari have limited CTAP 2.1 implementation. These browsers typically prompt to enter the PIN even when the YubiKey is not in `biometrics blocked` state.

6.4.2 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 in `biometrics blocked` state and requires the PIN, you might need to resort to a computer (as opposed to a mobile device) to unblock biometrics (see: *Unblocking and Resetting the YubiKey Bio*).

6.5 YubiKey Bio and FIDO2

FIDO2 / WebAuthn builds upon FIDO U2F, and is the standard that enables the replacing password-based authentication with the more secure public key cryptography. See WebAuthn Introduction and Client to Authenticator Protocol (CTAP) 2.1.

The YubiKey Bio Series - FIDO Edition and the Multi-protocol Edition support all FIDO2 scenarios supported by the YubiKey 5 Series and the Security Key Series. They can be used in both passwordless and second factor authentication scenarios. In both scenarios the fingerprint is used *in lieu of* the PIN, the way biometrics are used on a smartphone. There are some scenarios in which the PIN is required; for example, 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 fingerprint.

6.5.1 Fingerprint Authentication

When biometrics is not blocked on the YubiKey Bio, the user has three opportunities to present a valid fingerprint by touching the fingerprint sensor. The platform - for example, the browser or desktop client - notifies the user if the fingerprint was recognized, that is, the attempt was successful. If unsuccessful, the YubiKey Bio's amber LED blinks 3 times.

After three unsuccessful fingerprint attempts the platform prompts for PIN entry. For more information on the FIDO2 PIN see *FIDO2 PIN*.

6.5.2 Discoverable Credentials

Like FIDO U2F, the FIDO2 standard (see WebAuthn Introduction) 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 (firmware 5.5 and 5.6) or 100 discoverable credentials (firmware 5.7 and later). To manage them, see *Credential Management*.

6.5.3 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.

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 discoverable credentials (firmware 5.5 and 5.6) or 100 discoverable credentials (firmware 5.7 and later).

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 YubiKey 5.2 enhancements to FIDO 2 support.

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

6.5.4 FIDO2 PIN

The FIDO2 PIN is necessary for:

- Enrolling fingerprints
- Managing enrolled fingerprints
- Fallback after failure to match fingerprint with template.
- 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 three times in a row, the key needs to be reinserted before it can accept additional PIN entry attempts. Reinserting "reboots" the key.

- To find out how many retries you have left, use the Yubico Authenticator tool or the ykman tool.
- If the PIN is entered incorrectly eight times in a row (3+3+2), the FIDO2 application locks, and FIDO2 authentication is not possible.
- To restore the FIDO2 functionality, reset the FIDO2 application.

For YubiKey Bio Series - FIDO Edition keys, the FIDO2 PIN must be between 4 and 63 characters in length. For YubiKey Bio Series - Multi-protocol Edition keys, PIN length requirements depend on firmware version and YubiKey application status:

YubiKey Bio Series - Multi-protocol Edition PIN Length Requirements		
Firmware version:	PIV application enabled:	PIV application disabled:
Prior to 5.7.4	6-8 characters	4-63 characters
5.7.4 and later	6-8 characters	6-63 characters

For more information, see Understanding YubiKey PINS and *Device PIN*.

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.

6.5.5 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)*.

6.5.6 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.

6.6 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.

6.6.1 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 Support site and look for articles about the YubiKey Bio.

6.6.2 FIDO U2F Superseded 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*.

6.6.3 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.

6.7 YubiKey Bio and PIV

The YubiKey Bio Multi-protocol Edition supports PIV in addition to FIDO U2F and FIDO2. The PIV application works the same way as the PIV application on a YubiKey 5 Series, see *Smart Card - PIV Compatible Specifics*, with the differences set out below.

6.7.1 Yubico Minidriver and YubiKey Bio

With the PIV protocol on some operating systems and browsers, using the biometric (fingerprint) option on YubiKey Bio requires client software or middleware. To access the PIV application on the YubiKey Bio Multi-protocol Edition, if fingerprint authentication for cryptographic operations does not work, enter the YubiKey PIN.

For Windows operating systems, Yubico provides biometric supporting middleware, the YubiKey Smart Card Minidriver. See:

- Download from Smart card drivers and tools.
- YubiKey Minidriver User Guide.
- Deploying the YubiKey Smart Card Minidriver to workstations and servers.

6.7.2 Device PIN

On the YubiKey Bio Multi-protocol Edition, the PIN is shared between the PIV and FIDO2 applets. *PIN length requirements* depend on firmware version and YubiKey application status.

6.7.3 Device PUK

On the YubiKey Bio Multi-protocol Edition, because the PIN is shared between FIDO2 and PIV, the PUK is blocked by default. This means that a device that has a blocked PIN on PIV cannot be unblocked by the PUK. Instead, it must be reset (see: *Resetting*).

6.8 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.

6.8.1 Enrollment

Before you can start using the YubiKey Bio with services and applications, you need to first set a *FIDO2 PIN*. Once a PIN is set, biometric functionality (recommended but not required for use with passkeys) can be enabled by enrolling 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 any of the following:

- The YubiKey Bio start page
- The Yubico Authenticator
- The ykman Releases page.

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/products/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.

6.8.2 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.

YubiKey Bio Multi-protocol Edition
: When using the YubiKey Bio Multi-protocol Edition the client needs to support the PIV protocol. For Windows, the client can use the Yubico Minidriver installed (version 4.6.0 or later). See YubiKey Minidriver User Guide.

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.

6.9 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.

6.9.1 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.

6.9.2 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.

6.9.3 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, because U2F has no concept of PIN.

If the fingerprint match is unsuccessful, you must unblock biometrics by using any of the following:

- The YubiKey set up page
- The Yubico Authenticator tool
- The ykman tool.

6.10 Understanding YubiKey Bio Enrollment

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 any of the following:

- The YubiKey Bio start page
- The Yubico Authenticator
- The ykman Releases page.

6.10.1 Adding a PIN to YubiKey Bio

Before you add fingerprints to your YubiKey Bio, it is highly recommended that you create a PIN for the YubiKey as a backup to the fingerprint authentication. See Support article, Understanding YubiKey PINs: The Basics.

The wizards to add security keys provided in the Windows and Chrome (for macOS, Linux, and Chrome OS devices) include an option to add, change, or reset the YubiKey PIN.

The Yubico Authenticator also provides a method for adding PINs to your YubiKey Bio. To use Yubico Authenticator for setting YubiKey PINs, see:

- Creating a FIDO2 PIN on desktop and Android
- Creating a FIDO2 PIN on iOS/iPadOS

- Download and Install the Yubico Authenticator

  Note that on Windows 10, the Yubico Authenticator must be run in Administrator mode.

6.11 Enrolling Fingerprints on Chrome

Set a PIN and enroll the *first* fingerprint using the Chrome browser on a macOS, Linux or Chrome OS device. To enroll one or more fingerprints use the Chrome settings as described in *Setting Up Your YubiKey Bio on Chrome*.

For information on the YubiKey Bio sensor and tips on working with fingerprints see *Fingerprint Tips*. For detailed information on FIDO2 PINs and their requirements, see Understanding YubiKey PINS.

6.11.1 Setting Up Your YubiKey Bio on Chrome

See the link and video Set up your YubiKey > Enroll your Fingerprint > Fingerprint Enrollment > macOS / Linux / Chrome OS.

1. Access security settings.

   From a Chrome browser, click the three vertical dots in the upper right of the browser window and select **Settings** > **Privacy and Security** > **Security**. Scroll to select **Manage security keys**.

   Optionally, paste `chrome://settings/securityKeys` into the Chrome address field.

2. Create PIN on your YubiKey.

   If your YubiKey does not have a PIN, set it now.

   a. Click **Create a PIN**.

   b. Insert and touch the YubiKey BIO.

   c. Enter and confirm a PIN. Then click **Save**.

   d. At the Your PIN was created message, click **OK**.

3. Enroll your fingerprint.

   In the the Manage wizard click **Fingerprints** and follow the prompts.

   a. Insert YubiKey Bio.

   b. Enter the security key PIN. Then click **Continue**.

   c. Click **Add** to add fingerprints.

   d. Touch the fingerprint sensor on the YubiKey Bio until you see **Your fingerprint was captured**. Then click **Continue**.

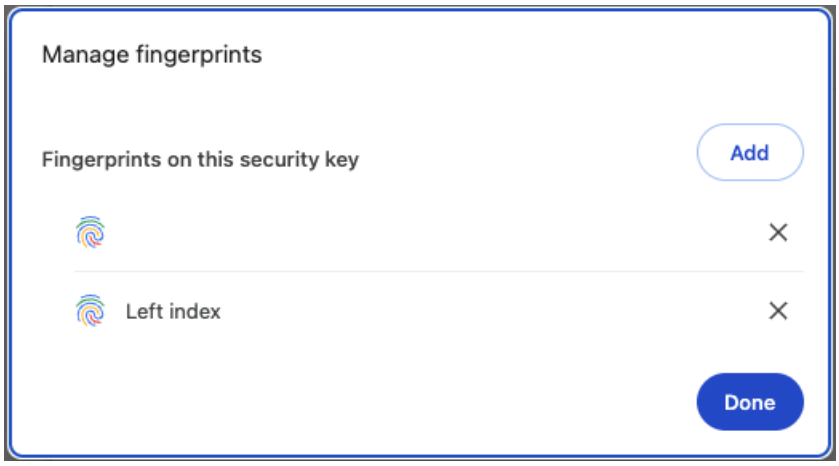
      Touch the fingerprint sensor and the bezel. You are prompted to touch the sensor several times. Change the angle of finger to sensor slightly each time. If the fingerprint you enroll is smaller than the sensor, apply some pressure to help ensure a good image capture.

   e. Enter a name for the fingerprint. For example, Right Thumb. Click **Done**.

4. Enroll additional fingerprints.

   Repeat this Step 3 to enroll different fingers from either hand. The YubiKey Bio holds a total number of 5 enrolled fingerprints.

   If you are fingerprint management panel, start with Step 3c, click **Add**.



Chrome on macOS, Linux, and Chrome OS: Capturing the Fingerprint

The YubiKey Bio now has a template for that fingerprint.

6.12 Enrolling Fingerprints on Windows

Set a PIN and enroll fingerprints on a YubiKey Bio using the Sign-in options on a Windows 10 or Windows 11 system.

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.

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.

6.12.1 Setting Up Your YubiKey Bio on Windows

See the link and video Set up your YubiKey > Enroll your Fingerprint > Fingerprint Enrollment > Windows 10 and Windows 11.

1. Access security settings.

   From Windows, open **Settings** > **Accounts** > **Sign-in Options** > **Security Key**. Expand Security Key field, then click **Manage**.

2. Insert your YubiKey Bio into your device USB port. Then tap your YubiKey Bio.

3. Create PIN on your YubiKey.

   If your YubiKey does not have a PIN, set it now. Click **Add** Security Key PIN. Enter a PIN you select in both fields and click **OK**.

4. Enroll your fingerprint. In the the Windows Setup wizard:

   a. Click Security Key Fingerprint **Set up** and follow the prompts.

   b. Enter the security key PIN. Then click **OK**.



c. Touch the fingerprint sensor.

   Touch the fingerprint sensor and the bezel. You are prompted to touch the sensor several times. Change the angle of finger to sensor slightly each time. If the fingerprint you enroll is smaller than the sensor, apply some pressure to help ensure a good image capture.

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

5. Enroll additional fingerprints.

   Repeat this Step 4 to enroll different fingers from either hand. The YubiKey Bio holds a total number of 5 enrolled fingerprints.

   The YubiKey Bio now has a template for that fingerprint.

6.13 Unblocking and Resetting the YubiKey Bio

The main cause for the biometric function to block is failure to match the fingerprint three times in a row. If the YubiKey Bio locks because the biometric function was blocked, you can just unblock it instead of resetting it: see *Troubleshooting and Tools*.

6.13.1 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
To unblock the YubiKey Bio's biometric function (its ability to read fingerprints) by using any of the following:

- The YubiKey Bio start page
- The Yubico Authenticator
- The YubiKey Manager (ykman) Releases Releases page.

6.13.2 Resetting

Resetting your YubiKey Bio deletes all credentials, the PIN, and stored fingerprint templates.

Resetting the key is not the same as unblocking it. Resetting the FIDO2 and FIDO U2F applications returns the key to the factory default state. In this default state, the key has neither fingerprints nor PIN nor credentials. After resetting it, you must therefore enroll your fingerprints again and register your key again to your apps and services. See the relevant Enrolling chapter, either *Enrolling Fingerprints on Chrome* or *Enrolling Fingerprints on Windows*.

Note: The YubiKey Bio Multi-protocol Edition behaves differently from other YubiKey Bios. To ensure that devices and applications do not unintentionally destroy credentials on that key, if both PIV and FIDO are enabled, the native reset commands for these protocols will be disabled and replaced with a single new reset command for both applications.

Therefore, if you try to run the individual protocol resets on a YubiKey Bio Multi-protocol Edition, you will get an error message. Conversely, if you try to run the reset command for the YubiKey Bio Multi-protocol Edition on a YubiKey Bio that is not the Multi-protocol Edition, you will also get an error message.

Because the PIN and fingerprint templates have been unified, the individual PIV and FIDO2 protocols can only be enabled or disabled on a freshly reset device. It is not possible to enable or disable protocols after a PIN is set.

Resetting Your YubiKey Bio with the Yubico Authenticator and Other Tools

With a YubiKey Bio that is not the Multi-protocol Edition, resetting means resetting the FIDO application. You can perform a FIDO reset using any of the following:

- The YubiKey set up page
- The Yubico Authenticator tool
- The ykman tool
- Windows Sign-in options
- The Chrome browser settings.

YubiKey Bio Multi-protocol Edition

To reset the YubiKey Bio Multi-protocol Edition please use the Yubico Authenticator tool.

6.13.3 Managing Credentials to Sites or Services

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 discoverable credentials (firmware 5.5 and 5.6) or 100 discoverable credentials (firmware 5.7 and later).

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 YubiKey 5.2 enhancements to FIDO 2 support.

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

6.14 Fingerprint Tips

6.14.1 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.

6.14.2 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:

6.14.3 Fingerprint Orientation

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

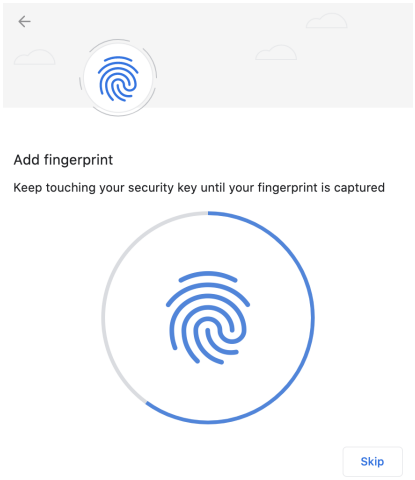


Fig. 1: Chrome on macOS, Linux, and Chrome OS: Capturing the Fingerprint

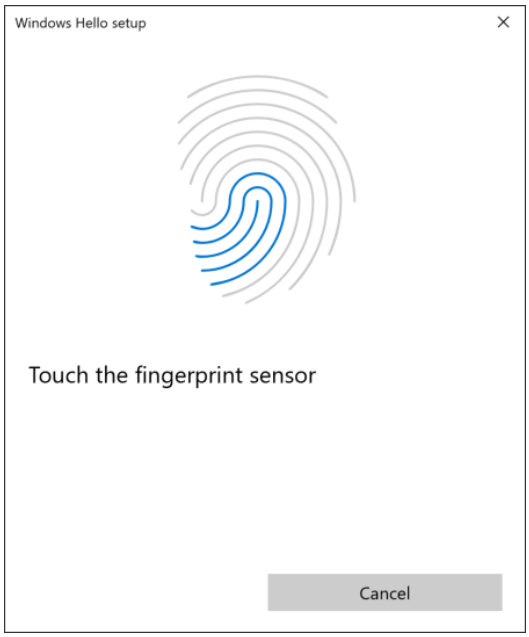


Fig. 2: Windows: Capturing the Fingerprint

6.14.4 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.

6.15 Troubleshooting and Tools

6.15.1 Troubleshooting

The primary source for troubleshooting tips is the FAQ on the YubiKey set up 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
To unblock the YubiKey Bio's biometric function (its ability to read fingerprints) by using any of the following:

- The YubiKey Bio start page
- The Yubico Authenticator
- The YubiKey Manager (ykman) Releases Releases page.

Unblocking/Unlocking

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
To unblock the YubiKey Bio's biometric function (its ability to read fingerprints) by using any of the following:

- The YubiKey Bio start page
- The Yubico Authenticator
- The YubiKey Manager (ykman) Releases Releases page.

For resetting, see *Resetting*.

Other Issues

For any issues with a YubiKey Bio, refer to the Yubico Support 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 Submit a request with Yubico Support.

6.15.2 Tools

Yubico Authenticator

The Yubico Authenticator tool can be used to manage the YubiKey Bio. It is open source and cross-platform, running on Windows, macOS, and Linux. Note that the iOS and Android versions of Yubico Authenticator cannot be used to manage the YubiKey Bio.

CHAPTER

SEVEN

NFC ID CALCULATION TECHNICAL DESCRIPTION

7.1 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.

7.2 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.

7.3 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.

Note: FIDO Reset over NFC on Windows If you have a YubiKey with the PIV capability enabled and you have never reset the FIDO2 application, you might find that your first attempt to reset the FIDO2 application fails with an error message. On the second attempt the application will be reset successfully.

CHAPTER

EIGHT

TOOLS AND TROUBLESHOOTING

8.1 Managing Applications

8.1.1 Enabling/Disabling

To find out which applications are enabled on which interface, you can use either the Yubico Authenticator, see Yubico Authenticator User Guide > Settings, or the ykman CLI, see ykman CLI and YubiKey Manager GUI Guide.

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

8.1.2 Locking

Once the desired applications have been selected, you can set a lock code to prevent anyone changing the set of applications that are enabled. To do this, you can use the ykman CLI `ykman config set-lock-code`. The lock code is 16 bytes presented as 32 hex characters. For more information, see ykman CLI and YubiKey Manager GUI Guide for that command.

8.2 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.

Yubico Authenticator is one of the tools most commonly used to configure YubiKeys. For a complete breakdown of Yubico Authenticator functionality by platform and connection type for each YubiKey model, see the Yubico Authenticator Functionality table.

The Yubico Authenticator, which replaces the old YubiKey Manager GUI, 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.

8.3 YubiKey Manager GUI / ykman CLI

ykman is a CLI tool for configuring all aspects of 5 Series YubiKeys and for determining the model of YubiKey and the firmware version it is running. It is a cross-platform tool that runs on Windows, macOS, and Linux.

Note: The YubiKey Manager (GUI) is available, but is not robust. For a graphical interface, we recommend the Yubico Authenticator.

Be aware, some of the more advanced options are only available through the latest version of the ykman, as documented in the ykman CLI and YubiKey Manager GUI Guide. Download and install the most recent version of ykman.

8.3.1 GUI: YubiKey Manager

The YubiKey Manager has been superseded by the Yubico Authenticator. However, the YubiKey Manager is still available; it 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.

8.3.2 CLI: ykman

Using ykman, you can do everything that YubiKey Manager can and more. ykman can also do more than the Yubico Authenticator. 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 ykman CLI and YubiKey Manager GUI Guide.

8.4 YubiKey Minidriver

The YubiKey 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 YubiKey Minidriver User Guide.

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

8.5 YubiKey Verification Site and FIDO Application Demo Site

The Verify your YubiKey page on the Yubico website enables users to:

- Validate the authenticity of the YubiKey
- Identify the model
- Read the firmware version on YubiKeys with firmware 5.4.0 and later.

For more detailed instructions, see How to Confirm Your Yubico Device is Genuine and/or Where to find YubiKey firmware details.

The Yubico WebAuthn Developer Tool offers users the ability to:

- Demo the capabilities of the YubiKey FIDO application
- Inspect the FIDO2 Attestation Certificate.

To set a FIDO2 PIN without using ykman, refer to Understanding YubiKey PINS.

8.6 Troubleshooting

For any issues with a YubiKey 5 series, refer to the Yubico Support 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 Submit a request with Yubico Support.

CHAPTER

NINE

YUBIKEY 5 CSPN SERIES SPECIFICS

Important: The YubiKey 5 CSPN Series has been discontinued.

9.1 CSPN Specific Marking

All of the keys in the YubiKey 5 CSPN Series have the acronym **CSPN** inscribed on the backs of the YubiKeys, underneath the Data Matrix code.



YubiKey Series 5 CSPN-Specific Markings

9.2 YubiKey 5 CSPN Series Firmware Version

Table 1: **Features Available per Firmware Version**

Feature	Firmware Version 5.4.2
Serial Number	Yes
OTP	Yes
OATH	Yes
OATH Credential Storage	
OpenPGP version	
PIV/Smart Card	Yes

continues on next page

Table 1 – continued from previous page

Feature	Firmware Version 5.4.2
FIDO U2F	Yes
FIDO2/WebAuthn	Yes
FIDO2 Credential Storage	
FIDO2 PIN Mgmt	
Enterprise Attestation	
Blob Storage	
AlwaysUV	
YubiHSM Auth	
SCP03	Yes
SCP11	

9.3 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.

9.3.1 Listing the Applications on the YubiKey 5

To obtain a list of all applications on the YubiKey 5, use ykman to execute the `ykman info` command (ykman CLI and YubiKey Manager GUI Guide).

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.

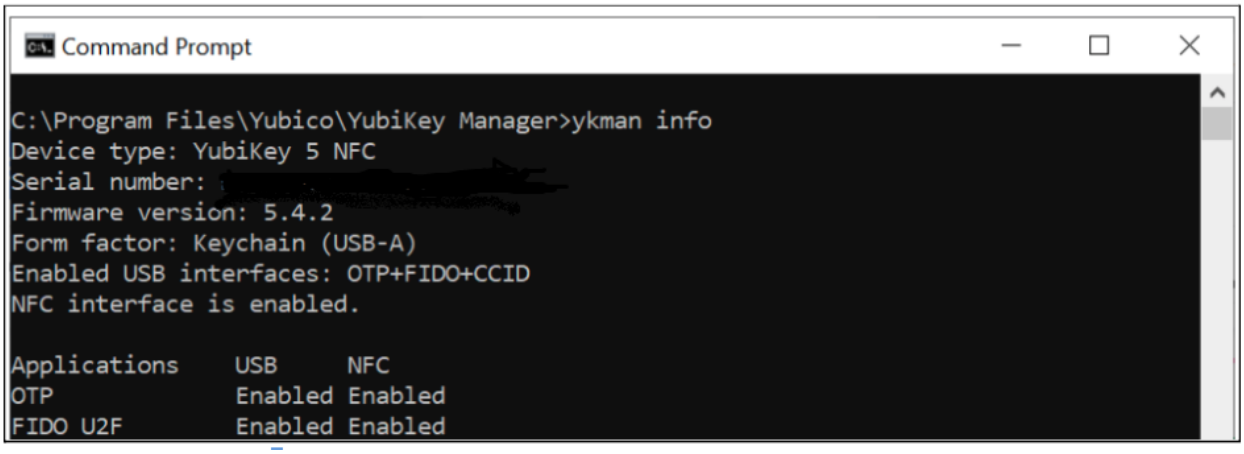


Fig. 1: **Figure 1** - Example of listing the applications on a YubiKey 5

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

9.3.2 Password Strength

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

9.3.3 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.

9.4 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.

9.4.1 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 ykman <https://developers.yubico.com/yubikey-manager/Releases/>`_. Note that you need the most recent version of the CLI, not the GUI. For more information about this tool, see ykman CLI and YubiKey Manager GUI Guide.

To protect 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.

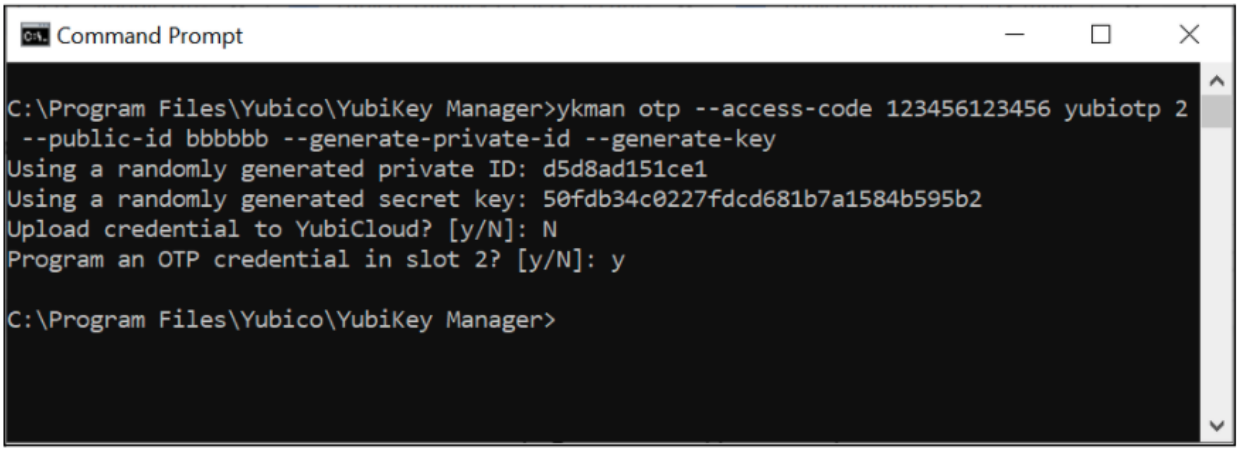


Fig. 2: **Figure 2** - Example of configuring protected Yubico OTP credentials

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 no longer possible to delete the Yubico OTP credentials without providing the correct access code. The screenshot below shows another example of how to use the ykman CLI 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.

9.4.2 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.

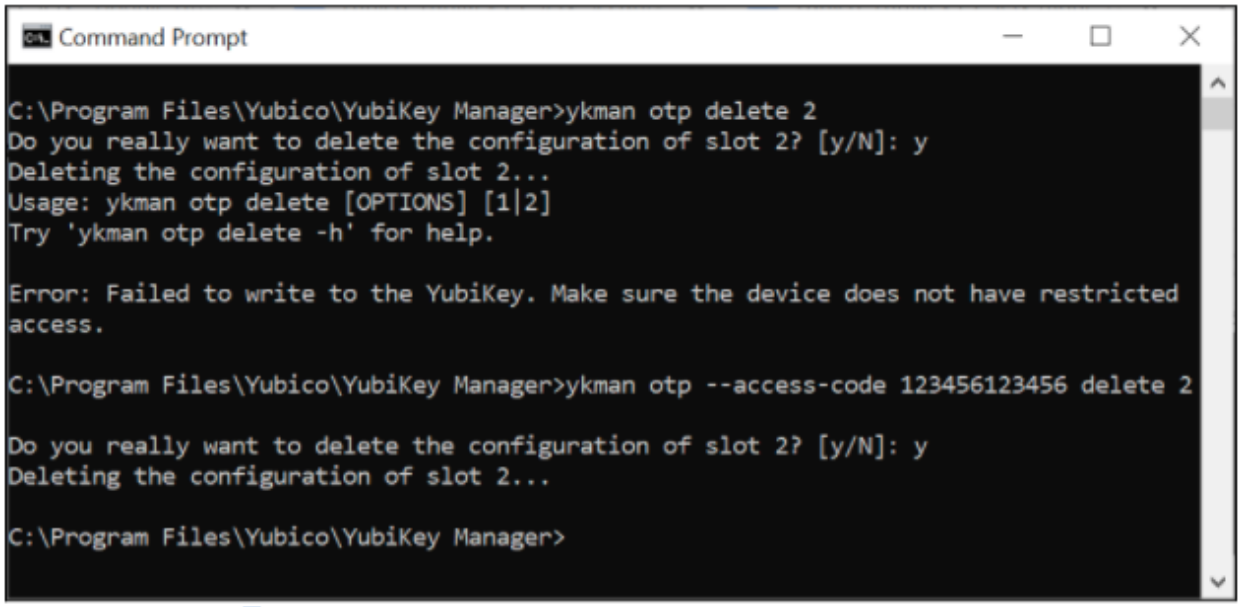


Fig. 3: **Figure 3** - Example of deleting protected Yubico OTP credentials

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 ykman. Install the most recent version of the ykman CLI, because the YubiKey Manager GUI includes an older version of the ykman CLI.

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 ykman CLI interaction for creating protected Challenge-Response credentials requiring touch.

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]`

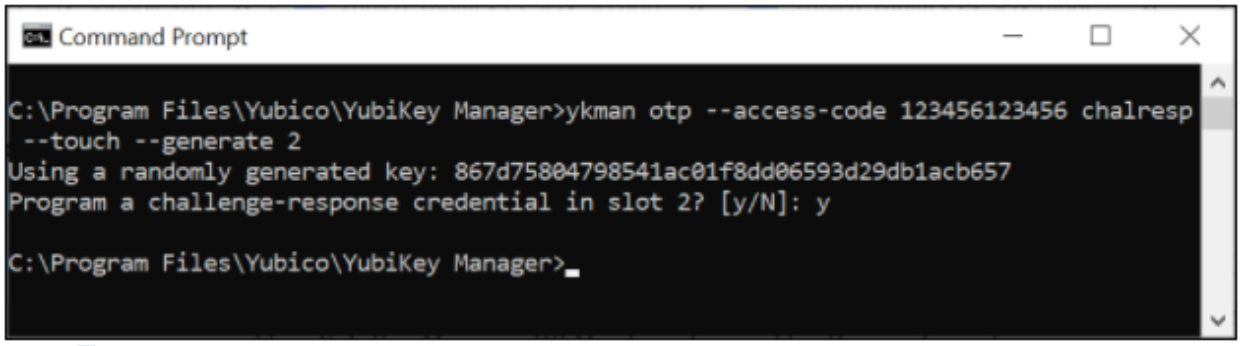


Fig. 4: **Figure 4** - Example of configuring protected Challenge-Response credentials with touch sensor

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 shows 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.

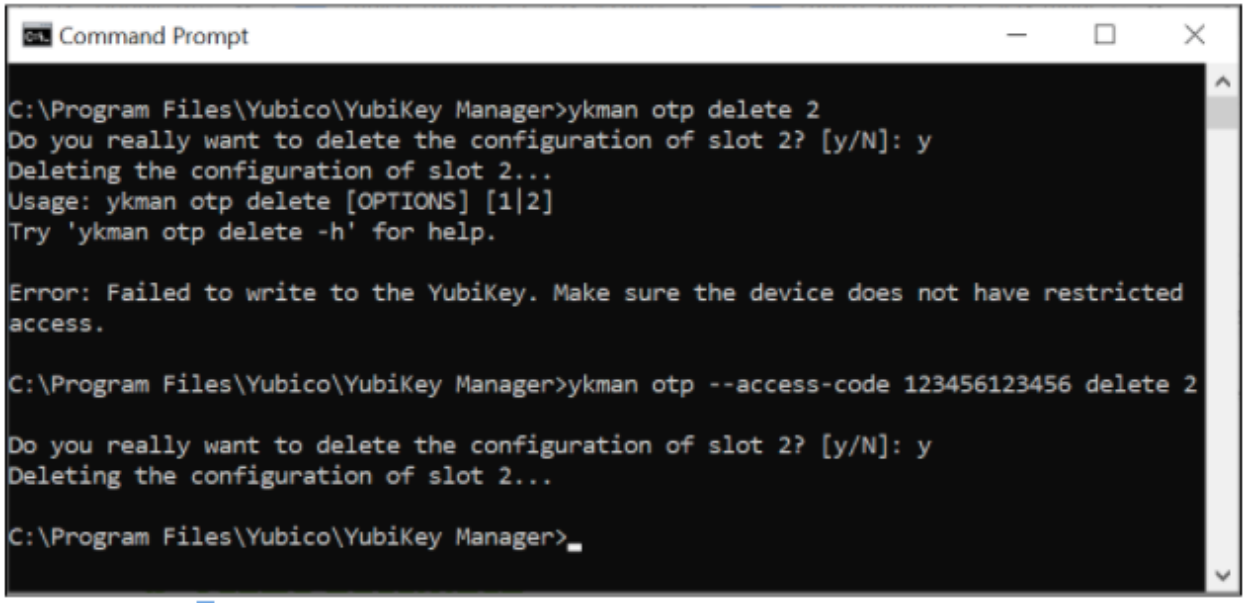


Fig. 5: **Figure 5** - Example of deleting protected Challenge-Response credentials

9.4.3 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 Programming a static password into your YubiKey.

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 ykman. Install the most recent version of the ykman CLI, because the YubiKey Manager GUI includes an older version of the ykman CLI.

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.

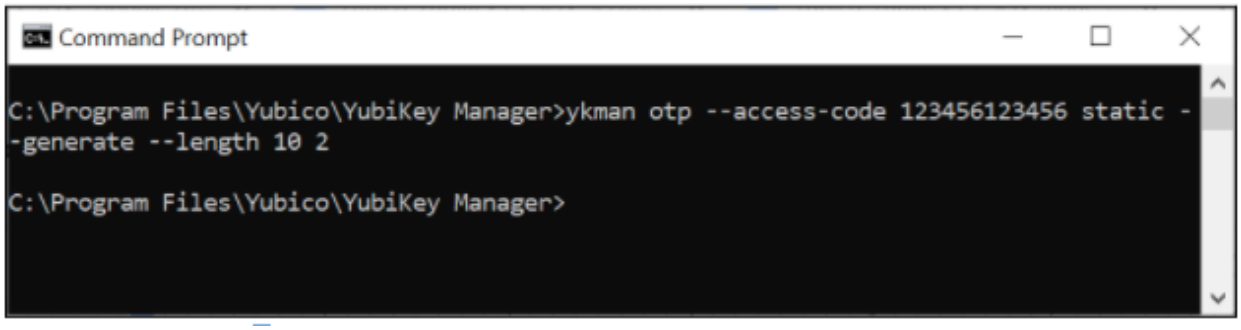


Fig. 6: **Figure 6** - Example of configuring a protected static password

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 shows 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.

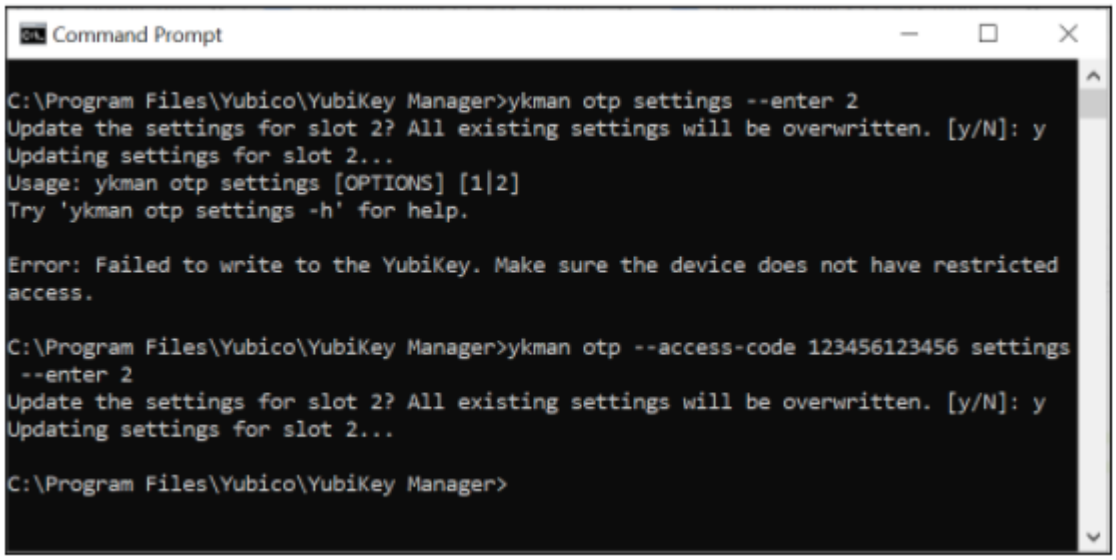


Fig. 7: **Figure 7** - Example of changing a protected static password

9.4.4 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 ykman. Install the most recent version of the ykman CLI, because the YubiKey Manager GUI includes an older version of the ykman CLI.

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.

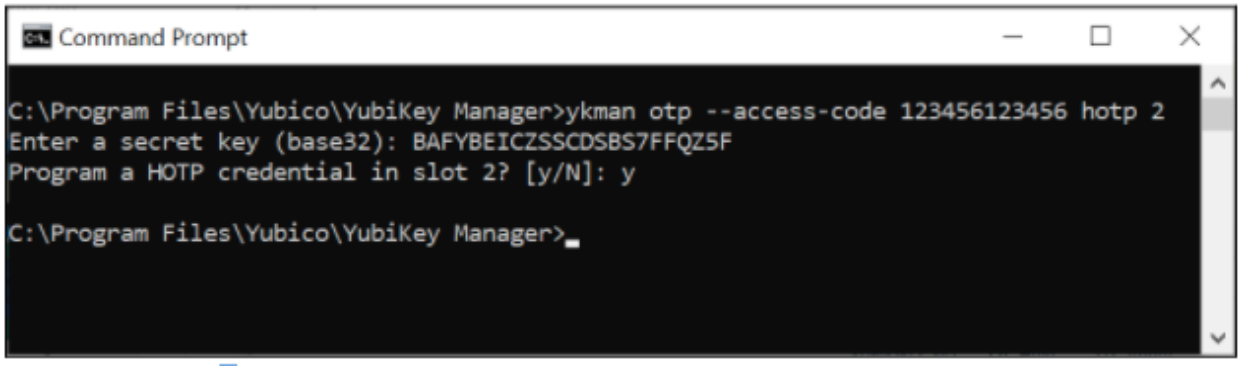


Fig. 8: **Figure 8** - Example of configuring protected OATH-HOTP credentials

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 CLI 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.

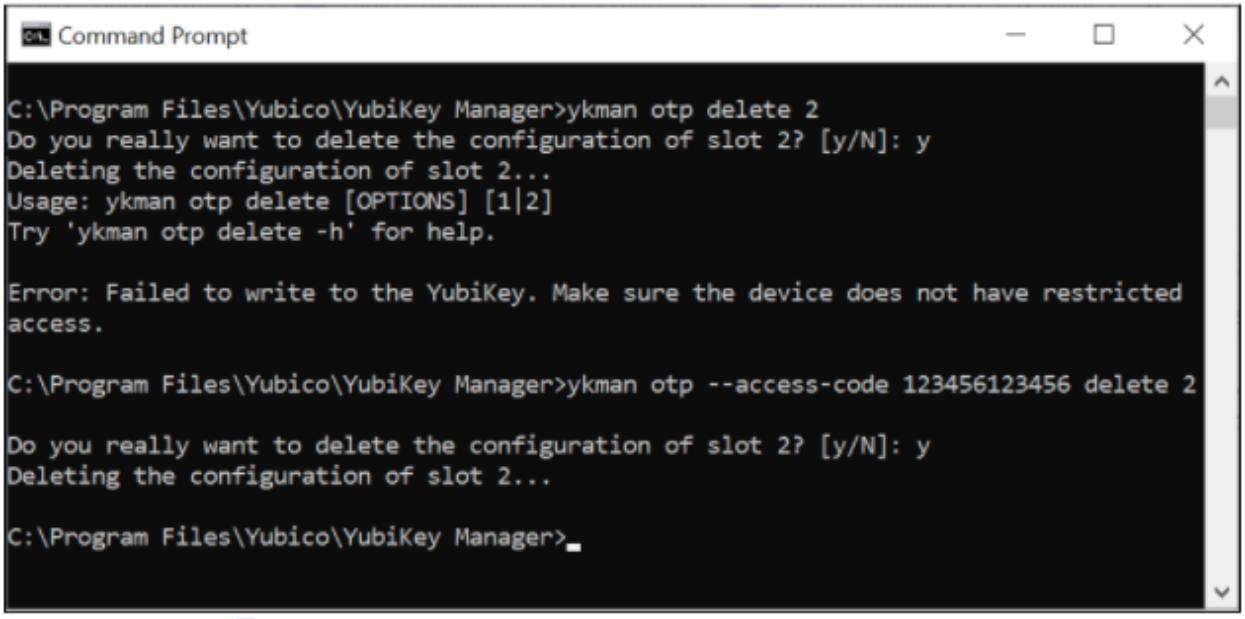


Fig. 9: **Figure 9** - Example of deleting protected OATH-HOTP credentials

9.5 OATH

9.5.1 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.

9.5.2 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.

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

9.5.3 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.

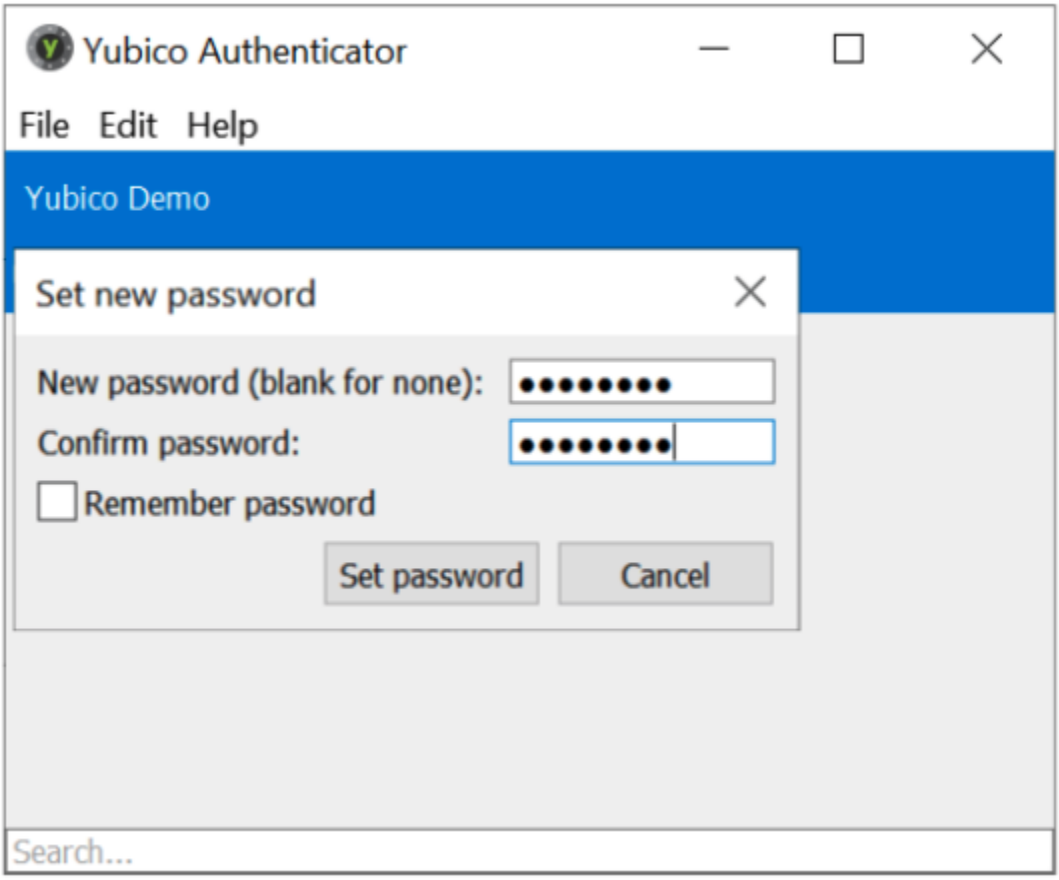


Fig. 10: **Figure 10** - Example of protecting the OATH-HOTP/TOTP credentials with a password

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

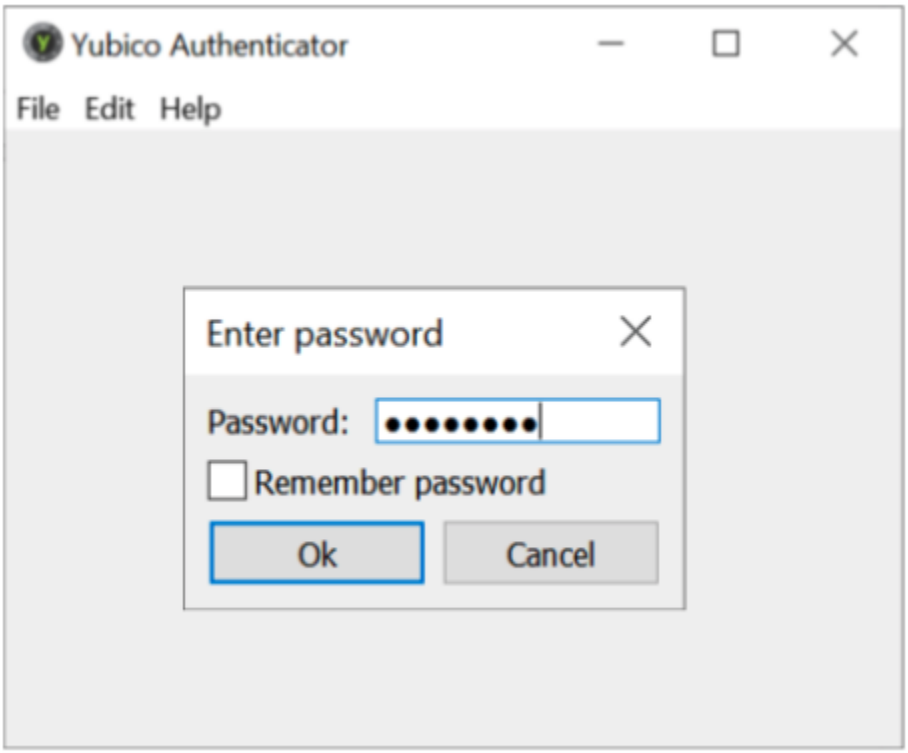


Fig. 11: **Figure 11** - Example of unlocking the OATH-HOTP/TOTP credentials

9.6 FIDO U2F

9.6.1 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 developer website, U2F.

9.6.2 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.

9.6.3 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.

9.7 FIDO2

9.7.1 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.

9.7.2 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*)

9.7.3 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 ykman or Yubico Authenticator 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.

9.7.4 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.

9.7.5 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 ykman or Yubico Authenticator.
- 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

Use Yubico Authenticator or ykman 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 GUI:

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

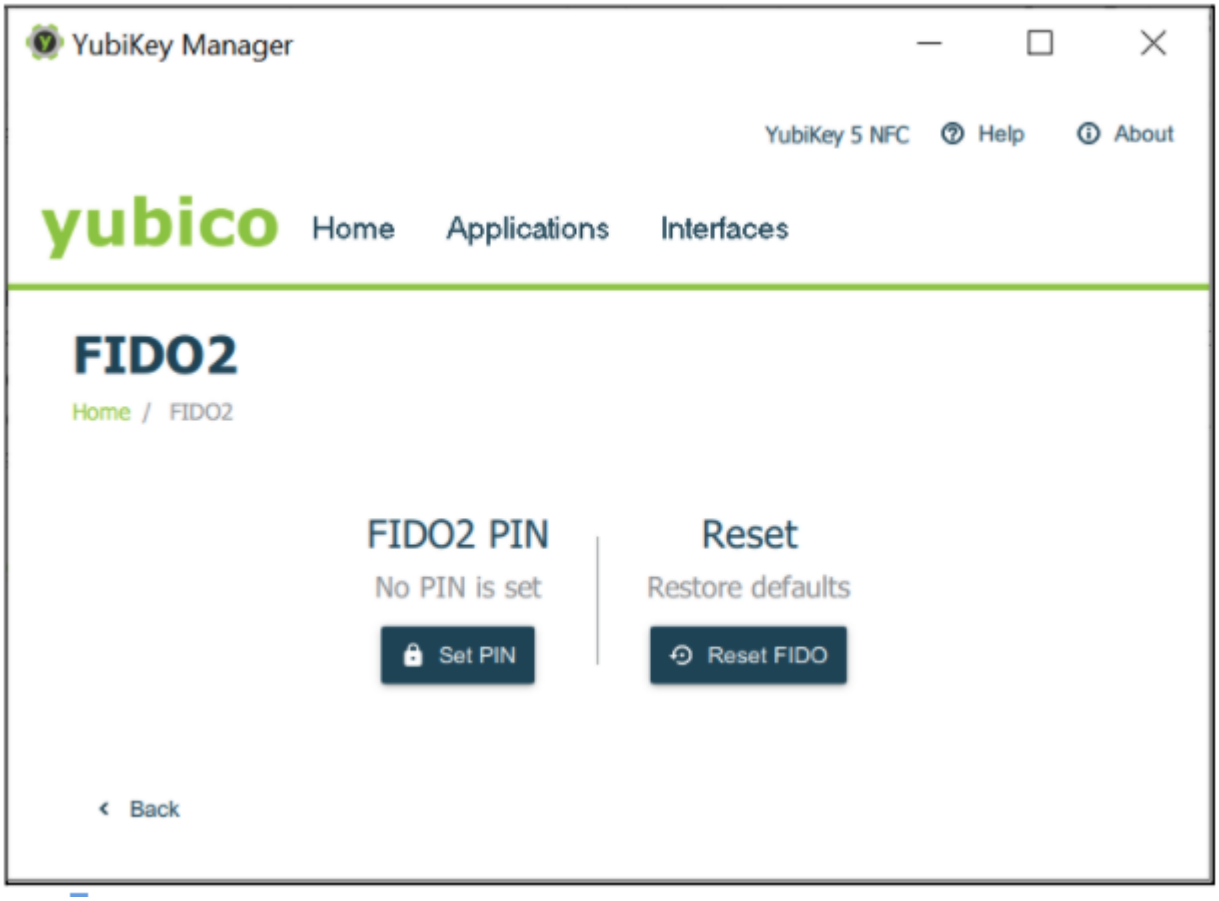


Fig. 12: **Figure 12** - Configuring the FIDO2 PIN with YubiKey Manager

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

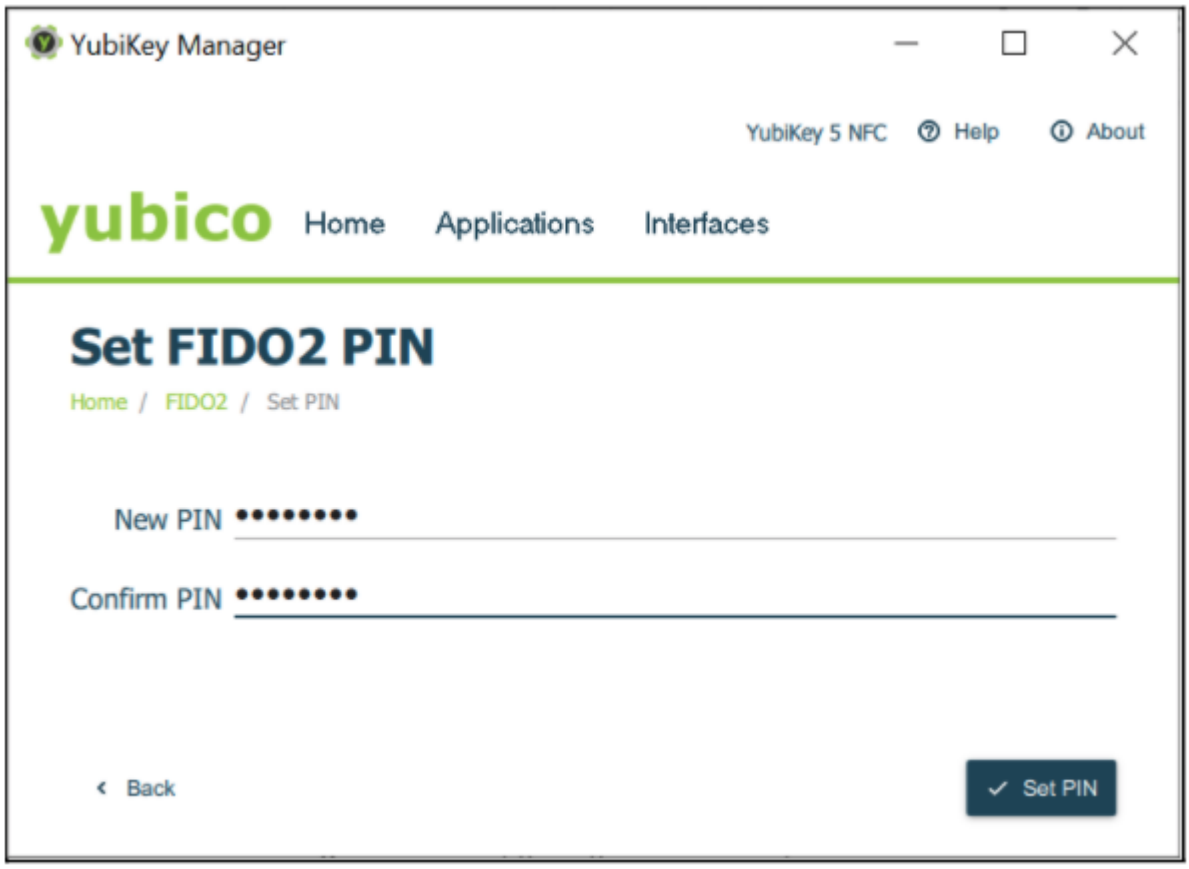


Fig. 13: **Figure 13** - Configuring the FIDO2 PIN with YubiKey Manager

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.

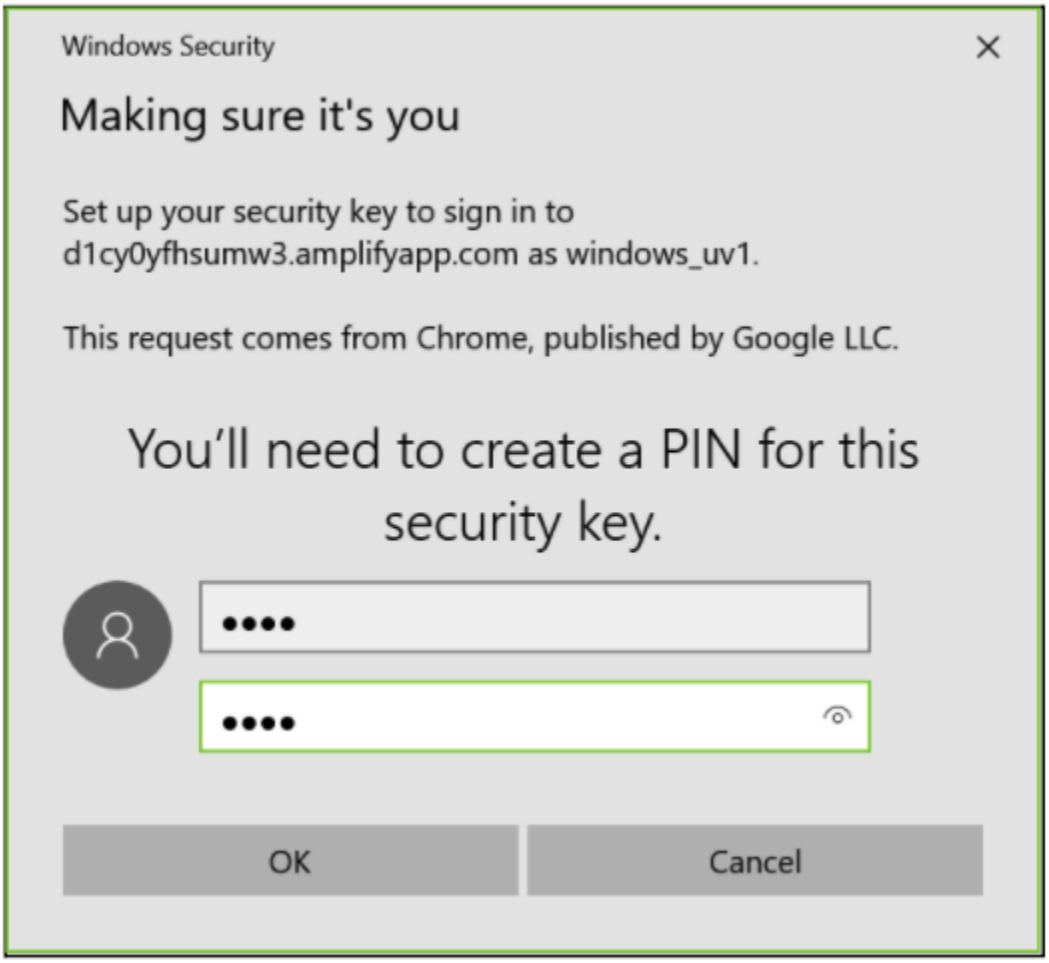


Fig. 14: **Figure 14** - Configuring the PIN code for FIDO2 with Windows 10

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 GUI, ykman, or Yubico Authenticator 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.

9.8 PIV

9.8.1 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.

9.8.2 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.

9.8.3 Technical Configuration

PIN Configuration of PIV

Use the YubiKey Manager GUI, ykman, or Yubico Authenticator to set the PIN, PUK, and management key on the YubiKey. In this scenario, a YubiKey 5 with default settings is assumed.

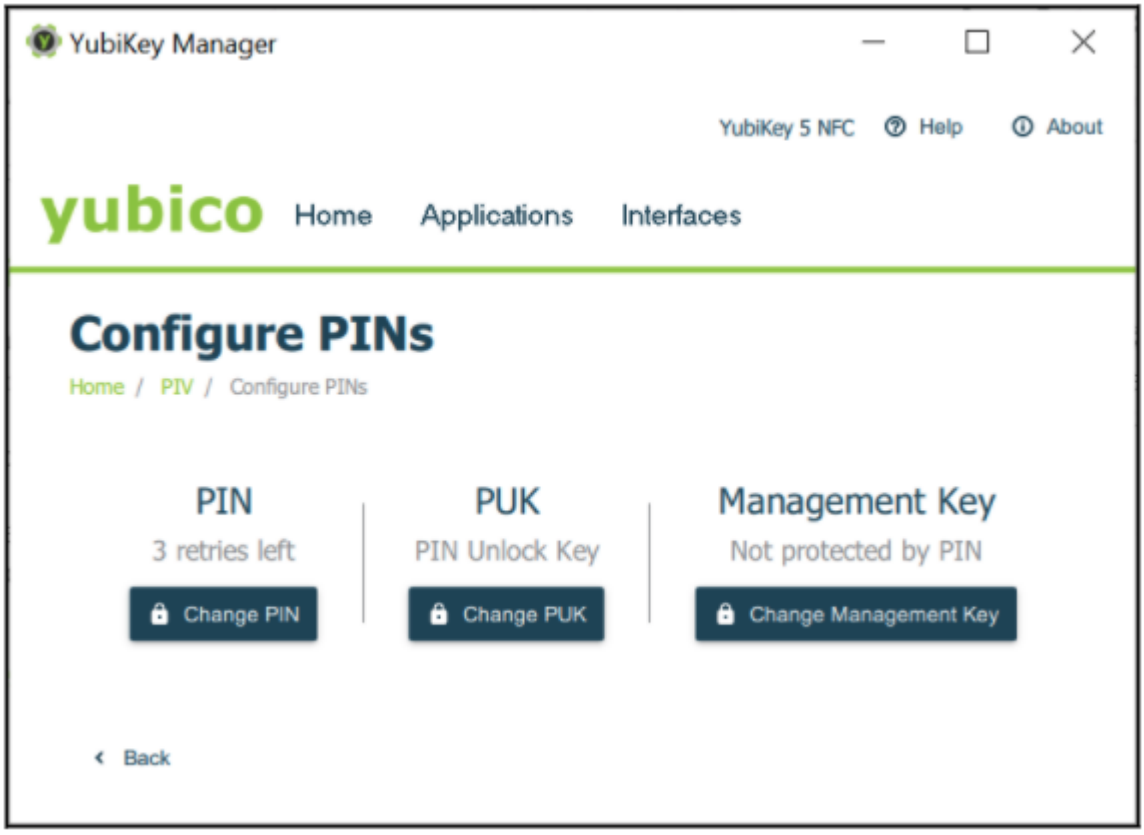


Fig. 15: **Figure 15** - Configuring the PIN, PUK, and management key for PIV

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 GUI, 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 ykman when generating or importing the key-pair on the YubiKey 5. Use the command:

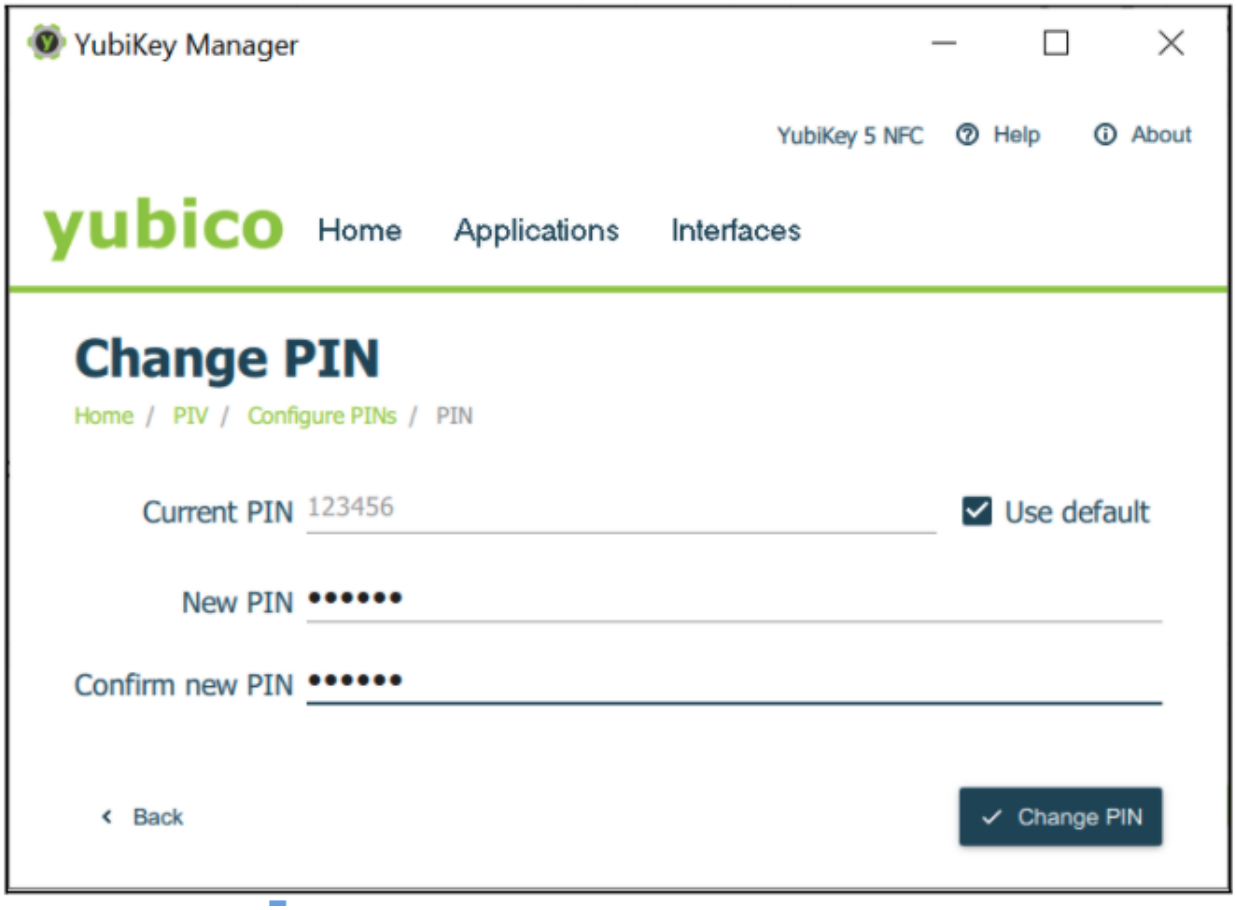


Fig. 16: **Figure 16** - Changing the PIN code for PIV

```
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 GUI, is shown in the following figure.

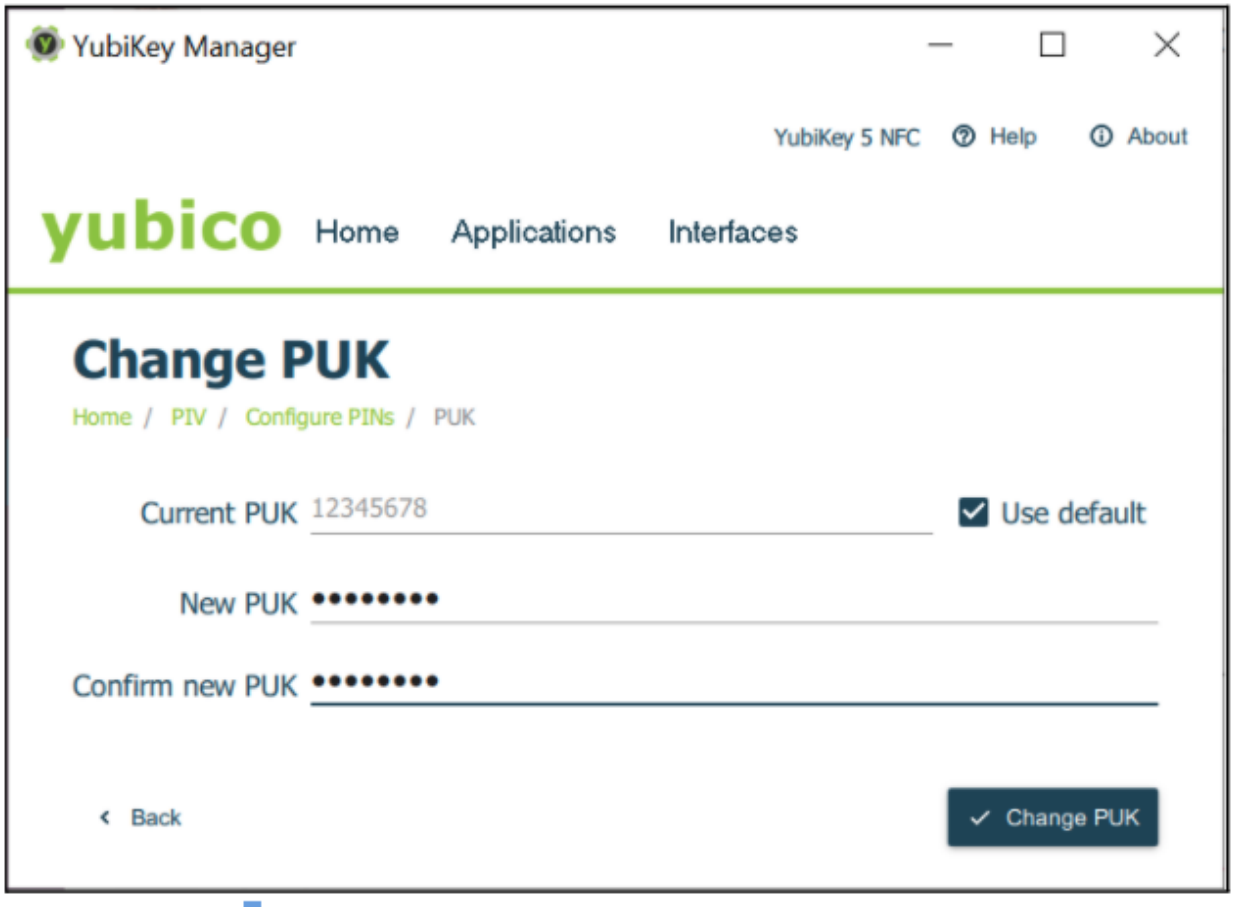


Fig. 17: **Figure 17** - Changing the PUK code for PIV

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 GUI, is shown in the following figure.

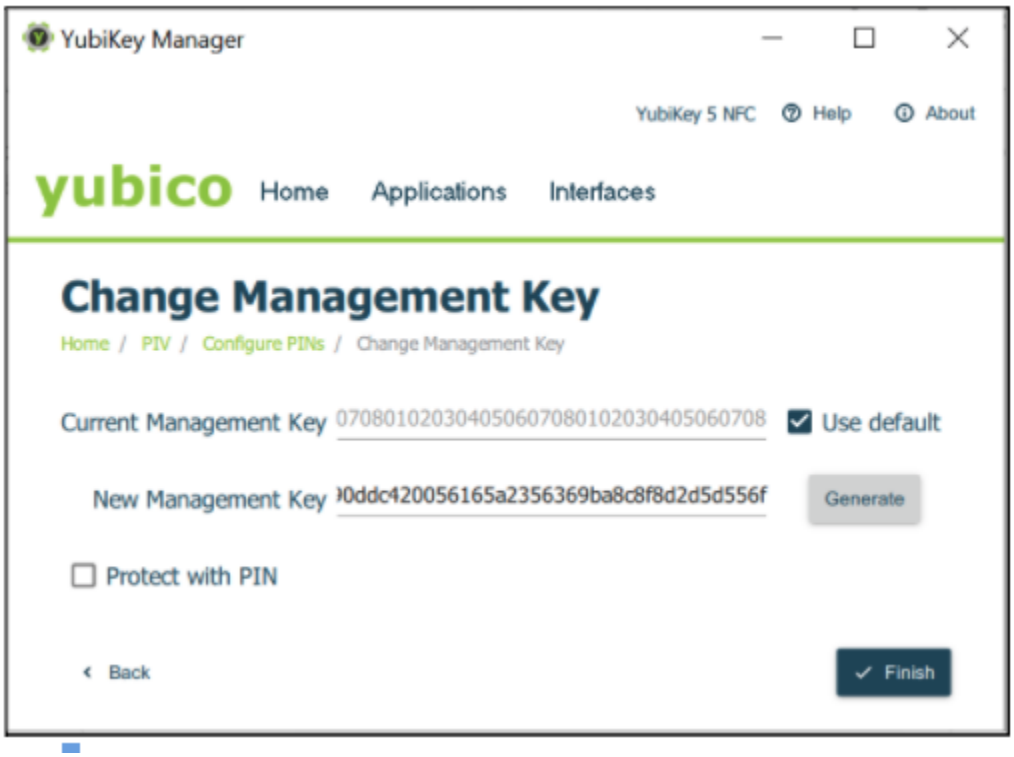


Fig. 18: **Figure 18** - Changing the management key for PIV

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.

9.9 References

Table 2: **CSPN Reference Documents**

Code	Document title	Reference (Link)
[RD1]	Certification de sécurité de premier niveau des technologies de l'information	https://cyber.gouv.fr/

continues on next page

Table 2 – continued from previous page

Code	Document title	Reference (Link)
[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/passkeys/
[RD4]	NIST Special Publication 800-73 (PIV)	https://csrc.nist.gov/pubs/sp/800/73/4/upd1/final
[RD5]	RFC 4226, An HMAC-Based One-Time Password Algorithm	https://datatracker.ietf.org/doc/html/rfc4226
[RD6]	T/Key: Second-Factor Authentication From Secure Hash Chains	https://arxiv.org/pdf/1708.08424
[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/

CHAPTER

TEN

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

CHAPTER

ELEVEN

COPYRIGHT

© 2021-2026 Yubico AB. All rights reserved.

11.1 Trademarks

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

11.2 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.

11.3 Contact Information

Yubico AB
Gävlegatan 22
113 30 Stockholm
Sweden

11.4 Getting Help

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

Contact

- Submit a request
- Contact our sales team
- Set up your YubiKey

11.5 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/s/requests and provide as much detail as you can.

11.6 Document Updated

2026-11-02 23:16:49 UTC