yubico

Yubico PIV Tool Command Line Guide

Command Line Reference

YubiKey 4, YubiKey 4 Nano, YubiKey NEO, YubiKey NEO-n

yubico

Contact Information

Yubico Inc
420 Florence Street, Suite 200
Palo Alto, CA 94301
USA
yubi.co/contact

Document Release Date

June 6, 2016

Contents

Introduction

Yubico changes the game for strong authentication, providing superior security with unmatched ease-of-use. Our core invention, the YubiKey, is a small USB and NFC device supporting multiple authentication and cryptographic protocols. With a simple touch, it protects access to computers, networks, and online services for the world's largest organizations.

Our innovative keys offer strong authentication via Yubico one-time passwords (OTP), FIDO Universal 2nd Factor (U2F), and smart card (PIV, OpenPGP, OATH) — all with a simple tap or touch of a button. YubiKeys protect access for everyone from individual home users to the world's largest organizations.

PIV and YubiKeys

The YubiKey 4, YubiKey 4 Nano, YubiKey NEO, and YubiKey NEO-n support the Personal Identity and Verification Card (PIV) interface specified in the National Institute of Standards and Technology (NIST). This enables you to perform RSA or ECC sign and decrypt operations using a private key stored on the YubiKey. Your YubiKey acts as a smart card in this case, through common interfaces like PKCS#11.

The structure of the YubiKey as a PIV card follows the specifications defined above. PIV can also be used for document signing, encryption, and physical access. There are four PIV slots on the YubiKey. Each slot is reserved for a specific purpose as defined by the NIST specifications:

- 9a -- PIV Authentication
- 9c -- Digital Signature
- 9d -- Key Management
- 9e -- Card Authentication

For more information about the PIV specifications see the PIV standards on the NIST website.

Yubico PIV Tool Command Line

We provide the Yubico PIV Tool for interacting with the Privilege and Identification Card (PIV) application on a YubiKey. Using the Yubico PIV Tool, you can generate keys on the device, import keys and certificates, create certificate requests, and other operations. A shared library and a command line tool is included.

The following topics are covered in this document:

- Downloading the Yubico PIV Tool
- Getting Command Help
- Access Control Matrix
- Command Line Options

- Parameters for the Action Option
- Troubleshooting

Getting Help

For more information, and to get help with your YubiKeys, see:

- Support home page
- Documentation and FAQs
- Start a Support ticket

Using YubiKey PIV for Smart Card Login Documentation Set

- YubiKey PIV Quick Start Guide
- YubiKey PIV Deployment Guide
- YubiKey PIV Manager User's Guide
- Yubico PIV Tool Command Line Guide (this document)
- CSIS Enrollment Station Guide

Each of the documents is available for download on the Yubico website.

Downloading the Yubico PIV Tool

Before you can use the Yubico PIV Tool Command Line to build scripts, be sure to download the file that corresponds to your operating system, and then extract the compressed files.

NOTE: The `x` in the following procedures represents the most recent release of the Yubico PIV Tool.

To download and get started with the Yubico PIV Tool Command Line

1. Under **PIV Tools** and **Yubico PIV Tool (command line)**, download the latest version of the Yubico PIV Tool compressed file from the Yubico website for the operating system you are using.

2. To start using the Yubico PIV Tool, do one of the following:

   a. For Windows:

      i. Extract the files from the `yubico-piv-tool-x.x.x-win64.zip` file (for 64-bit Microsoft Windows) or the `yubico-piv-tool-x.x.x-win32.zip` file (for 32-bit Microsoft Windows).

         where `x.x.x` is the version for the file you downloaded.

      ii. Move the extracted files to a directory of your choice, and open **Command Prompt**.

      iii. Move to the `bin` directory. For example, if you moved the files to the root of the `C` drive, type:

         cd c:\yubico-piv-tool-*x.x.x*-win64\bin
         where `x.x.x` is the version for the file you downloaded.

      iv. Press **Enter**, and then begin using the Yubico PIV Tool Command Line.

   b. For Mac OS X:

      i. Extract the files from the `yubico-piv-tool-x.x.x-mac.zip` file.

         where `x.x.x` is the version for the file you downloaded.

      ii. Move the extracted files to a directory of your choice, and open a **Terminal** window.

      iii. Move to the **bin** directory. For example, if you moved the files to the **Applications** folder, type the following in the Terminal window:

```
cd /Applications/yubico-piv-tool-x.x.x-mac/bin
```

         where `x.x.x` is the version for the file you downloaded.

      iv. Press **Return**, and then begin using the Yubico PIV Tool Command Line.

c. For Linux:

   i. Install the package from one of these distributions:

- https://packages.qa.debian.org/y/yubico-piv-tool.html
- https://admin.fedoraproject.org/pkgdb/package/rpms/yubico-piv-tool/
- https://aur.archlinux.org/packages/yubico-piv-tool/

   ii. Launch and complete the installation process for your Linux distro.

Getting Command Help

Use one of the help switches— `-h`, `--help`, or `--full-help`—to view the help page for the command actions and options.

NOTE: The `--verbose` action does not change the results for the `-h`, `--help`, and `--full-help` actions.

-h and --help switches

If you are not sure how to use a command, type:

```
yubico-piv-tool -h
```

or type:

```
yubico-piv-tool --help
```

For each action, you will see usage information and a list of options you can use with the action.

--full-help

The `--full-help` switch enables you to print full help, including hidden options.

```
yubico-piv-tool --full-help
```

Access Control Matrix

The table below contains a list of possible operations and their requirements for authenticating with the management key (MGM), PIN, and PUK. The requirements vary, depending on the action or option you need to use when creating your scripts.

Operation	Actions Associated with each Operation	Requires			Notes
		MGM	PIN	PUK	
Generate key pair	generate	x			
Management key	set-mgm-key	x			
Change retry counters	`pin-retries`, (requires `--puk-retries` parameter)	x	x		Resetting the counters also resets the PIN and PUK to their default values
Import key	import-key	x			Import private key
Import certificate	import-certificate	x			
Set cardholder identifier	set-CHUID set-CCC	x			
Reset card	reset				Can only be used if *both* the PIN and the PUK are blocked
Verify PIN	verify-pin		x		
Change PIN	change-pin		x		Requires a new PIN
Change PUK	change-puk			x	Requires a new PUK
Reset PIN	unblock-pin			x	Requires that you type a new PIN

yubico

Command Line Options

This chapter documents the options for the Yubico PIV Tool. The parameters for the `-a` or `--action` options are described in the next chapter, Parameters for the Action Option.

Option name	Description and example

`-a`
`--action=ENUM`
	See the next chapter, Parameters for the Action Option, for the list of parameters that the `-a` or `--action` option can take. The remaining options in this table are additional switches or options that you can use in the command line.

`-A, --algorithm=ENUM`	Specifies the encryption algorithm to use
	Values for the algorithm option:
	`RSA1024`, `RSA2048`, `ECCP256`, `ECCP384`
	The default value is `RSA2048`.
	
	Example (to generate a new `ECCP256` key on device in slot `9a`):
	
	`yubico-piv-tool -s 9a -A ECCP256 -a generate`
	
	NOTE: In this example, the default management key is used to authenticate. If the management key was changed from the default, this command line fails unless you add the `--key` option with the new management key.

`-f, --format=ENUM`	Specifies the format of data for write/read object
	Values for the format option:
	`hex`, `base64`, `binary`
	The default value is `hex`.
	
	Example (to read an object identified by `0x5fc102` with format `binary`):
	`yubico-piv-tool -a read-object --id="0x5fc102" --format=binary`

`--full-help`	Print help, including hidden options, and exit
	
	Example (to view `--full-help` for the Yubico PIV Tool):
	
	`yubico-piv-tool --full-help`

`-h, --help`	Print help and exit
	Examples (both examples provide the same results):
	`yubico-piv-tool -h`
	or
	`yubico-piv-tool --help`

Option name	**Description and example**
`-H, --hash=ENUM`	Specifies the hash to use for signatures Values for the hash option: SHA1, SHA256, SHA384, SHA512 The default value is `SHA256`. Example (to request a certificate in slot `9a` specifying `SHA512` to use for the signature - this example assumes that the certificate was already generated in slot `9a`): `yubico-piv-tool -s 9a -P 123456 -a request-certificate --hash=SHA512 -S '/CN=bar/OU=test/O=example.com/'`
`-i, --input=STRING`	Specifies the file name to use as input For `stdin`, the value is –. The default value is –. Example (to import a certificate from a file named `test.pfx` with password `test`, into slot `9c`) `yubico-piv-tool -s 9c -i test.pfx -p test -a import-cert` **NOTE**: In this example, the default management key is used to authenticate. If the management key was changed from the default, this command line fails unless you add the `--key` option with the new management key.
`--id=INT`	Required parameter for write-object and read-object actions Example (to read object indicated by `0x5fc102`): `yubico-piv-tool -a read-object --id="0x5fc102"`
`-k, --key[=STRING]`	Authentication key to use (default='010203040506070801020304050607080102030405060708') Example (change the management key): `yubico-piv-tool -k $currentMGM -a set-mgm-key -n $newMGM`
`-K, --key-format=ENUM`	Format of the key being read or written Values for -K or --key-format: PEM, PKCS12, GZIP, DER The default value for this option is `PEM`. Example (to import a certificate from a file named `test.pfx` with format `PKCS12` and password `test`, into slot `9c`): `yubico-piv-tool -s 9c -i test.pfx -K PKCS12 -p test -a import-cert` **NOTE**: In this example, the default management key is used to authenticate. If the management key was changed from the default, this command line fails unless you add the `--key` option with the new management key.

Option name	Description and example

`-n, --new-key=STRING`	New authentication key to use
	This is a required parameter for `set-mgm-key`
	Example (to generate a new management key when the management key is the default):
	`yubico-piv-tool -a set-mgm-key -n $newkey`
	Example (to generate a new management key when the management key is not the default):
	`yubico-piv-tool -k $currentMGM -a set-mgm-key -n $newMGM`

`-N, --new-pin=STRING`	Specifies new PIN or PUK code
	Example (to change the PIN from `123456`, the default, to `$pin`):
	`yubico-piv-tool -a change-pin -P 123456 -N $pin`

`-o, --output=STRING`	Specifies filename to use as output For `stdout`, the value is `–`. The default value is `–`. Example (to generate the key for slot `9a` and save the key to a file named `public-pem`): `yubico-piv-tool -s 9a -a generate -o public.pem` **NOTE**: In this example, the default management key is used to authenticate. If the management key was changed from the default, this command line fails unless you add the `--key` option with the new management key.
`-p, --password=STRING`	Password for decryption of private key file Example (to import a key and import a certificate from a `PKCS12` formatted file named `test.pfx` with password `test`, into slot `9c`): `yubico-piv-tool -s 9c -i test.pfx -K PKCS12 -p test -a import-key -a import-cert` NOTE: In this example, the default management key is used to authenticate. If the management key was changed from the default, this command line fails unless you add the `--key` option with the new management key.
`--pin-policy`	Optional parameter for `generate` and `import-key` actions The values are `never`, `once`, and `always`. `yubico-piv-tool -s 9a -A ECCP256 -a generate --pin-policy=once` **NOTE**: In this example, the default management key is used to authenticate. If the management key was changed from the default, this command line fails unless you add the `--key` option with the new management key.
`-P, --pin=STRING`	PIN or PUK code for verification

Option name	Description and example
	Example (authenticate with `123456` and then change the PIN to `$pin`): `yubico-piv-tool -a change-pin -P 123456 -N $pin`
`--puk-retries`	A required parameter for the `pin-retries` action, the `--puk-retries` parameter indicates the number of allowable PUK attempts before the PUK is blocked. **NOTE**: The `--puk-retries` is a required parameter of the `pin-retries` action. The `--puk-retries` parameter is not usable without the `pin-retries` action. Example (to reset the number of allowable PIN attempts and PUK attempts to `5` before blocking the PIN and the PUK): `yubico-piv-tool -a verify -P 123456 -a pin-retries --pin-retries=5 --puk-retries=5` **NOTE**: In this example, the default management key is used to authenticate. If the management key was changed from the default, this command line fails unless you add the `--key` option with the new management key. See also pin-retries
`-r, --reader=STRING`	Use only a matching reader The default value for this option is `Yubikey`. Example (to generate a certificate on slot `9a` and generate a valid list of readers): `yubico-piv-tool -a generate -s 9a -r "01 00"` **NOTE**: In this example, the default management key is used to authenticate. If the management key was changed from the default, this command line fails unless you add the `--key` option with the new management key.
`--serial=INT`	Serial number of the self-signed certificate The default value is `1`. Example (to generate a self-signed certificate with public key from `stdin`, print the certificate on `stdout` for later import, and specify the serial number of the self-signed certificate as `8`): `yubico-piv-tool -s 9a -P 123456 -a selfsign-certificate -S '/CN=bar/OU=test/O=example.com/' --serial=8`
`-s, --slot=ENUM`	Specifies the key slot on which to operate Values for `-s`: `9a, 9c, 9d, 9e, 82, 83, 84, 85, 86, 87, 88, 89, 8a, 8b, 8c, 8d, 8e, 8f, 90, 91, 92, 93, 94, 95`

Option name	Description and example

	This a required switch for all actions that operate on a key or certificate.
	Example (to generate an `ECCP256` key in slot `9a`):
	`yubico-piv-tool -s 9a -A ECCP256 -a generate`

`-S, --subject=STRING`	Subject to use for the certificate request
	The subject must be written as:
	`'/CN=host.example.com/OU=test/O=example.com/'`
	Example (to specify the subject of the certificate request):
	`yubico-piv-tool -s 9a -S '/CN=bar/OU=test/O=example.com/' -P 123456 -a verify -a selfsign`

`--touch-policy`	Optional parameter for `generate`, `import-key`, and `set-mgm-key` actions The values are `always`, `cached`, `never`. Example (to specify the touch policy for the key pair being generated): `yubico-piv-tool -s 9a -A ECCP256 -a generate --touch-policy=never`
`--valid-days=INT`	Time (in days) until the self-signed certificate expires The default value for this option is `365`. Example (to specify the number of valid days, `30`, of the certificate request): `yubico-piv-tool -s 9a -S '/CN=bar/OU=test/O=example.com/' -P 123456 -a verify -a selfsign --valid-days=30`
`-v, --verbose[=INT]`	Print more information `INT` is an optional parameter. The default value for `INT` is `0`. Example (to get more information about the results of the command line shown, in this case, the `pin-retries` action): `yubico-piv-tool -a pin-retries --verbose=2`
`-V, --version`	Print the version of the Yubico PIV Tool and exit Example (to print the version of the Yubico PIV Tool that you are using): `yubico-piv-tool -V`

Parameters for the Action Option

The `yubico-piv-tool -a [OPTION]` command takes one or more of the actions described in the sections of this chapter.

In this Chapter

- Syntax
- Multiple Action Options
- Each action, starting with the attest action

Syntax

```
yubico-piv-tool -a [OPTION]
```

or

```
yubico-piv-tool --action [OPTION]
```

Where [`OPTION]` is one or more of the options documented in the next sections, starting with `attest`.

Multiple Action Options

You can use multiple action options at once. If a command line uses multiple actions, they will be executed in order. For example:

```
yubico-piv-tool --action=verify-pin --action=request-certificate
```

The next sections provides command line samples of the action (`-a` or `-action`) options.

attest

When used on a slot with a generated key, outputs a signed x509 certificate for that slot showing that the key was generated in hardware.

NOTE: This action is available in firmware 4.3.0 and later.

Parameters

`-s`, indicating the slot, is a required parameter for this action and for all actions that operate on a key or a certificate.

Example

Generate a key pair on slot `9a` and show that the key was generated in hardware:

```
yubico-piv-tool -s 9a -a generate -a attest
```

change-pin

Change the PIN for the inserted YubiKey.

IMPORTANT: Be sure you know the current PIN, which you must type before you can specify a new PIN.

Parameters

`-P`, indicating the current PUK, is an optional parameter.

`-N`, indicating the new PUK, is an optional parameter.

Examples

Change the PIN (you will be prompted to type the current PIN):

```
yubico-piv-tool -a change-pin
```

Change the PIN to `$pin` after specifying the current PIN, which is `123456` in this example:

```
yubico-piv-tool -a change-pin -P 123456 -N $pin
```

change-puk

Change the PUK for the inserted YubiKey.

Parameters

`-P`, indicating the current PUK, is an optional parameter.

`-N`, indicating the new PUK, is an optional parameter.

Examples

Change the PUK (you will be prompted to type the current PUK and then the new PUK):

```
yubico-piv-tool -a change-puk
```

Change the PUK to `$puk` after specifying the current PUK, which is `12345678` in this example:

```
yubico-piv-tool -a change-puk -P 12345678 -N $puk
```

delete-certificate

Delete a certificate in slot `9a`.

Parameters

`-s`, indicating the slot, is a required parameter for this action and for all actions that operate on a key or a certificate.

Example

```
yubico-piv-tool -a delete-certificate -s 9a
```

generate

Generate a new key pair in the specified slot.

Parameters

`-s`, indicating the slot, is a required parameter for `generate` and for all actions that operate on a key or a certificate.

`pin-policy`, `touch-policy`, and `-A` are optional parameters:

`pin-policy=ENUM` Set the pin policy for action `generate`. Values for `pin-policy` are: `never`, `once`, `always`.

`touch-policy=ENUM` Set the touch policy for action `generate`. Values for `touch-policy` are: `always`, `cached`, `never`.

`-A`, an optional parameter, indicates the encryption algorithm to use. See the Command Line Options chapter earlier in this document.

Example

Generate a new `ECC-P256` key on the device in slot `9a`, and print the public key on `stdout`:

```
yubico-piv-tool -s 9a -A ECCP256 -a generate
```

import-key

Import a key to the inserted YubiKey.

Parameters

`-s`, indicating the slot, is a required parameter for this action and for all actions that operate on a key or a certificate.

`pin-policy` and `touch-policy` are optional parameters:

`pin-policy=ENUM` Set the pin policy for action `generate`. Values for `pin-policy` are: `never`, `once`, `always`.

`touch-policy=ENUM` Set the touch policy for action `generate`. Values for `touch-policy` are: `always`, `cached`, `never`.

Example

```
yubico-piv-tool -s 9c -i test.pfx -K PKCS12 -p test -a set-chuid -a import-key
-a import-cert
```

import-certificate

Import a certificate to the inserted YubiKey.

Parameters

`-s`, indicating the slot, is a required parameter for this action and for all actions that operate on a key or a certificate.

Example

Import a certificate from stdin to slot `9a` on the inserted YubiKey:

```
yubico-piv-tool -s 9a -a import-certificate
```

NOTE: The default management key is used to authenticate. If the management key was changed from the default, this command fails unless you add the `--key` option with the new management key.

list-readers

Show a list of the available readers for the inserted YubiKey.

Parameters

None

Example

Show the available readers for the inserted YubiKey:

```
yubico-piv-tool -a list-readers
```

pin-retries

Reset the number of PIN attempts before blocking the PIN.

Parameters

`--puk-retries`, indicating the number of retries before the PUK is blocked, is a required parameter for the `pin-retries` action.

`-P`, indicating the PIN for the inserted YubiKey, is a required parameter for `pin-retries`. -P depends on the parameter `-a verify` to function properly.

Example

Reset the number of allowable PIN attempts to `5` before blocking the PIN:

```
yubico-piv-tool -a verify -P 123456 -a pin-retries --pin-retries=5 --puk-
retries=5
```

See also puk-retries parameter

read-cert

Read the certificate from a specified slot.

Parameters

`-s`, indicating the slot, is a required parameter for this action and for all actions that operate on a key or a certificate.

Example

Read the certificate from a slot:

```
yubico-piv-tool -a read-cert -s 9a
```

read-object

Read the object indicated by the `--id`.

```
yubico-piv-tool -a read-object --id="0x5fc102" --format=binary
```

Parameters

`-id` a three-byte id starting with `0x5f`, indicates the id for the object to be read and is a required parameter for `read-object`. The value of `id` must be in enclosed in quotations as shown in the example.

`-f, --format`, indicating the structure of the data, is an optional parameter for `read-object`. Values include `hex`, `base64`, `binary`. The default value is `hex`.

Example

Read the indicated object:

```
yubico-piv-tool -a read-object --id="0x5fc102" --format=hex
```

request-certificate

Generate a certificate request.

Parameters

`-s`, indicating the slot, is a required parameter for this action and for all actions that operate on a key or a certificate.

`--hash`, indicating the hash to use for signatures, is an optional parameter for this action.

`-S` indicating the subject to use for the certificate request, is a required parameter for this action.

Example

Generate a certificate request in slot `9a` and authenticate with PIN `123456`:

```
yubico-piv-tool -s 9a -P 123456 -a request-certificate -S
'/CN=bar/OU=test/O=example.com/'
```

reset

Reset the application when the PIN and the PUK have been blocked due to incorrectly typing the PIN or PUK too many times.

IMPORTANT: The `reset` action removes all keys and objects from the YubiKey, and sets the default value for the PIN and the PUK. The default value for the PIN is `123456`, and the default value for the PUK is `12345678`. The attestation key and certificate are *not* cleared by a reset of the device.

Example

To reset the application:

```
yubico-piv-tool -a reset
```

When you successfully reset the YubiKey, Yubico PIV Tool display this message:

```
Successfully reset the application.
```

NOTE: The `reset` action can only be used if both the PIN and the PUK are blocked.

selfsign-certificate

Set a self-signed certificate on the specified slot.

Parameters

`-s`, indicating the slot, is a required parameter for this action and for all actions that operate on a key or a certificate.

`-S` indicating the subject to use for the self-signed certificate, is a required parameter for this action.

`--valid-days`, indicating the time (in days) until the self-signed certificate expires, is an optional parameter for this action. The default value is `365`.

`--serial`, indicating the serial number of the self-signed certificate, is an optional parameter for this action. The default value is `1`.

Example

Generate a self-signed certificate with public key from `stdin`, and print the certificate on `stdout` for later import:

```
yubico-piv-tool -s 9a -P 123456 -a selfsign-certificate -S
'/CN=bar/OU=test/O=example.com/'
```

set-ccc

Set a random Cardholder Capability Container (CCC).

For the application to be usable on Mac OS X, be sure you set the object `ccc` to a unique value. The card contents are aggressively cached. This means that if the card contents change, you also need to change the `ccc`.

Parameters

None

Example

Set the card compatibility container:

```
yubico-piv-tool -a set-ccc
```

When you successfully set a new `CCC`, Yubico PIV Tool displays this message:

```
Successfully set new CCC.
```

set-chuid

Set a random Cardholder Unique Identifier (CHUID).

For the application to be usable on Microsoft Windows, be sure you set the object `CHUID` to a unique value. The card contents are aggressively cached. This means that if the card contents change, you also need to change the `CHUID`.

Parameters

None required

Example

Set a random `CHUID` for slot `9c`:

```
yubico-piv-tool -s 9c -a set-chuid
```

When you successfully set a new `CHUID`, Yubico PIV Tool displays this message:

```
Successfully set new CHUID.
```

set-mgm-key

Change the management key used for administrative authentication.

IMPORTANT: The management key must be exactly 48 characters.

Parameters

`-n` is a required parameter for `set-mgm-key`: Specifies new authentication key to use.

`touch-policy` is an optional parameter for `set-mgm-key`:

`touch-policy=ENUM` Set the touch policy for action `generate`. Values for `touch-policy` are: `always`, `cached`, `never`.

Example

Set the management key to 080760540302010807060504030201080706050403020109:

```
yubico-piv-tool -n 080760540302010807060504030201080706050403020109 -a set-mgm-
key
```

When you successfully set a new management key, Yubico PIV Tool displays this message:

```
Successfully set new management key.
```

status

Shows the `CCC` (if set), the `CHUID` (if set), and the number of PIN retries remaining.

Parameters

none

Example

To show the PIV information stored on the inserted YubiKey:

```
yubico-piv-tool -a status
```

test-decipher

Validates the key pair (the key exchange determines whether it gets the same values) by encrypting data to a public key and deciphering it with the PIV tool.

Parameters

`-s`, indicating the slot, is a required parameter for this action and for all actions that operate on a key or a certificate.

Example

Encrypts to the public key in the certificate in slot `9a`:

```
yubico-piv-tool -a test-decipher -s 9a
```

test-signature

Start a signature test. You will be prompted to paste the certificate to test against.

Parameters

`-s`, indicating the slot, is a required parameter for this action and for all actions that operate on a key or a certificate.

Example

Launch a signature test on the indicated certificate:

```
yubico-piv-tool -s 9a -a test-signature
```

write-object

Write to the object indicated by the `id`.

Parameters

`-id` a three-byte id starting with `0x5f`, indicates the `id` for the object to be written to and is a required parameter for `write-object`. The value of `id` must be in enclosed in quotations as shown in the example.

`-f, --format`, indicating the structure of the data, is an optional parameter for `write-object`. Values include `hex`, `base64`, `binary`. The default value is `hex`.

`-i`, an optional parameter, indicates the file that contains the data to write to the object.

NOTE: If you do not include the `-i` option in the command line with the `write-object` action, you are prompted to paste the data to write to the object.

Example

Write the indicated object:

```
yubico-piv-tool -a write-object --id="0x5fc102" -i datafile --format=binary
```

verify-pin

Validate the PIN for the inserted YubiKey.

Example

Verify the PIN on the inserted YubiKey:

```
yubico-piv-tool -a verify-pin
```

You will then be prompted to type the PIN. If you type an incorrect PIN, you will receive a message with the number of tries remaining before the PIN is blocked; for example:

```
Pin verification failed, 2 tries left before pin is blocked.
```

If the PIN is blocked, a message like this appears:

```
Pin code blocked, use unblock-pin action to unblock.
```

NOTE: Be sure the PIN is 6-8 characters in length. Attempting to verify with anything shorter or longer than 6-8 characters does not count against the `pin-retries` action.

See also

- reset action
- unblock-pin action

version

Display the version of the application that is installed on the inserted YubiKey.

Parameters

None

Example

Show the version of the application that is installed on the inserted YubiKey:

```
yubico-piv-tool -a version
```

unblock-pin

Reset the PIN once the PIN has become blocked. Be sure you know the PUK before executing this action, as you will be prompted to type the PUK before you can reset the PIN.

See the PIN Code Requirements in the Troubleshooting chapter for the list of requirements for creating a new PIN.

NOTE: The `unblock-pin action` only works once the PIN is blocked. This action does not affect the `pin-retries` action.

Example

Reset the PIN on the inserted YubiKey:

```
yubico-piv-tool -a unblock-pin
```

Once you successfully reset the PIN, the following message appears:

```
Successfully unblocked the pin code.
```

See also

- pin-retries action
- reset action

yubico

Using Attestation

Attestation is used to show that a specified asymmetric key has been generated on the YubiKey itself, rather than imported to the YubiKey. Typically, you would use attestation before creating a certificate. Attestation is available on YubiKey 4 devices with firmware version 4.3.0 or later.

In this Chapter

Understanding Attestation

Attestation works through the special key slot, `f9`. The `f9` key slot, which you can overwrite, is loaded during manufacturing with a key and certificate signed by Yubico. After you generate a key in a slot, you can attest that key using the `f9` slot.

NOTE: Overwriting the Yubico key is permanent (the Yubico Key cannot be recovered). The attestation key and certificate are not cleared by a reset of the device.

When attestation occurs, the Yubico PIV Tool creates an X.509 certificate for the key that is to be attested. The certificate is created only if the key has been generated on a YubiKey rather than imported to the YubiKey.

IMPORTANT: The attestation certificate should only be used for the purpose of verifying that the key was generated on a YubiKey. Do not use this certificate for any other purpose.

Example Attestation

For example, the following commands generate a key in slot `9a` and then attest that key:

```
yubico-piv-tool --action=generate --slot=9a

yubico-piv-tool --action=attest --slot=9a
```

The result is a PEM-encoded certificate, signed by the key in slot `f9`.

Features of the Generated Certificate

The attestation certificate has the following features:

- Serial number: random 16-byte integer
- Issuer: subject of the attesting certificate
- Dates: copied from the attesting certificate
- Subject: the string, `YubiKey PIV Attestation`, with the attested slot appended

- Signature:
  - If the attesting key is RSA, the signature is `SHA256-PKCS`
  - If the attesting key is EC, the signature is `ECDSA-SHA256`

Extensions in the Generated Certificate

The generated attestation certificate has the following extensions:

- `1.3.6.1.4.1.41482.3.3`: Firmware version, encoded as 3 bytes. For example, version 4.3.0 is 040300.
- `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 is the PIN policy and the second is the touch policy. For example, a PIN policy of `always` and a touch policy of `always` is 0302.
  - PIN policy: 01 - never, 02 - once per session, 03 – always
  - Touch policy: 01 - never, 02 - always, 03 - cached for 15s

Verifying Attestation

To verify an attestation, you must build the certificate chain by putting the attestation root certificate in a file, adding the key's attestation certificate to the same file, and then typing the attest action.

To verify an attestation

1. To build the certificate chain, put the attestation root certificate in a file (or if you trust several root certificates, put all of them in the same file).
   TIP: The Yubico root certificate can be found on the Yubico Developers website.
2. Add the key's attestation certificate to the same file. For example, type:

   ```
   yubico-piv-tool --action=read-certificate --slot=f9 >> certs.pem
   ```

3. To verify the attestation, type the following commands:

   ```
   yubico-piv-tool --action=attest --slot=f9 > attestation.pem
   openssl verify -CAfile certs.pem attestation.pem
   ```

   If the attestation is successful, the Yubico PIV Tool displays these or similar results:

   ```
   attestation.pem: OK
   ```

Troubleshooting

If a command is not working the way you expect, try running the command line with increased verbosity to obtain more information about the results of the command. If you are receiving messages when you type a command, this section lists potential causes and solutions. If you want to change the PIN code, be sure you type a new PIN code that follows the requirements. Finally, if you changed the management key from the default, try running your command with the `-k` option and specifying the new management key.

In this Chapter

--verbose and --verbose=2 switches

For more information about each action or parameter, add `--verbose` or `--verbose=2` to the command, for example:

```
yubico-piv-tool -a change-pin --verbose
```

NOTE: The `--verbose` action does not change the results for the `-h`, `--help`, and `--full-help` actions.

PIN Code Requirements

When you choose a PIN code, be sure that you understand the following requirements:

- The PIN is a password that you type when you are requesting certificates, logging into the domain using your YubiKey, and so on.
- The PIN must be at least 6 characters in length, and no greater than 8 characters.
  NOTE: For Yubico PIV Tool versions prior to 1.4.0, there is no enforcement on PIN code length.

Errors and Messages

If you are not getting expected results with an action or command, be sure to type the action again with
`--verbose=2`. This may provide you with enough information to correct your command.

This section lists common Yubico PIV Tool errors and messages, with solutions for each one.

`Failed to connect to a reader`

This may indicate there are problems authenticating your YubiKey. If the management key was changed from the default, then you must specify the management key with the `-k` option for each action that requires authentication with the YubiKey.

`Failed authentication with the application`

This error message occurs when authentication with the management key fails. If you previously reset the management key, be sure you provide the new management key with the `-k` switch in every command line where YubiKey authentication is required.

This error also occurs if the PIN is required and is typed incorrectly.

For example:

```
yubico-piv-tool -a change-pin -P 123456 -N $pin -k
010203040506070801020304050607080102031234597899
```

where 010203040506070801020304050607080102031234597899 is the new management key.

`The puk code is blocked, you will have to reinitialize the application.`

The `unblock-pin` action requires that you provide the PUK code. If the PUK code is blocked, you must reset the YubiKey using the reset action. See the reset action. To use the reset action, both the PIN and the PUK codes must be blocked.

`Reset failed, are pincodes blocked?`

If you receive this message after attempting the `reset` action, it indicates that the PIN code is not blocked while the PUK code is blocked or that the PIN code is blocked but the PUK code is not blocked. Both the PIN code and the PUK codes must be blocked before you can implement the `reset` action. Once the PIN code is blocked, try the `reset` action.