## NAME

ykpersonalize − personalize YubiKey OTP tokens

## SYNOPSIS

**ykpersonalize** [*−Nkey*] [*−1* | *−2*] [*−sfile*] [*−ifile*] [*−fformat*] [*−axxx*] [*−cxxx*] [*−ooption*] [*−y*] [*−v*] [*−d*] [*−h*] [*−n*] [*−t*] [*−u*] [*−x*] [*−z*] [*−m*] [*−S*] [*−V*] [*−Dxxx_*]

## DESCRIPTION

Set the AES key, user ID and other settings in a YubiKey. For the complete explanation of the meaning of all parameters, see the reference manual: YubiKey manual (https://www.yubico.com/wp−content/uploads/2015/03/YubiKeyManual_v3.4.pdf)

## OPTIONS

**−Nkey**
use the nth YubiKey found.

**−1**
change the first configuration. This is the default and is normally used for true OTP generation. In this configuration, the option flag **−oappend−cr** is set by default.

**−2**
change the second configuration. This is for YubiKey II only and is then normally used for static key generation. In this configuration, the option flags **−oappend−cr**, **−ostatic−ticket**, **−ostrong−pw1**, **−ostrong−pw2** and **−oman−update** are set by default.

**−z**
delete configuration in selected slot.

**−s***file*
save configuration to file instead of key (if file is −, send to stdout).

**−i***file*
read configuration from file (if file is −, read from stdin). Configuration import is only valid for the ycfg format.

**−f***format*
format to be used with **−s** and **−i**. Valid options are **ycfg** and **legacy**.

**−a**[*xxx*]
the AES secret key as a 32 (or 40 for OATH−HOTP/HMAC CHAL−RESP) char hex value (not modhex) (none to prompt for key on stdin). If **−a** is not used a random key will be generated.

**−c**[*xxx*]
a 12 char hex value (not modhex) to use as the access code for programming. NOTE: this does NOT SET the access code. That is done with **−oaccess**=. If no argument is provided the code is prompted for on stdin.

**−o***option*
change configuration option. Possible option arguments are:

**fixed**=*ffffffffffff*
The modhex *public identity* of the YubiKey, 0−32 characters long (encoding up to 16 bytes). It's possible to give the identity in hex as well, just prepend the value with 'h:'. The fixed part is emitted before the OTP when the button on the YubiKey is pressed. It can be used as an identifier for the user, for example.

**uid**[=*uuuuuu*]
The uid part of the generated OTP, also called *private identity*, in hex. Must be 12 characters long. The uid is 6 bytes of static data that is included (encrypted) in every OTP, and is used to validate that an OTP was in fact encrypted with the AES key shared between the YubiKey and the validation service. It cannot be used to identify the YubiKey as it is only readable to those that know the AES key. If no argument is provided the uid is prompted for on stdin.

**access**[=*ffffffffffff*]
New hex access code to set. Must be 12 characters long. If an access code is set, it will be required for subsequent reprogramming of the YubiKey. If no argument is provided code is prompted for on stdin.

**oath−imf**=*xxx*
Set OATH Initial Moving Factor. This is the initial counter value for the YubiKey. This should be a value between 0 and 1048560, evenly dividable by 16.

*ticket−flag*
Set/clear ticket flag, see the section *Ticket Flags*.

*configuration−flag*
Set/clear configuration flag, see the section *Configuration flags*.

**−y**
always commit without prompting.

**−d**
dry−run, run without writing a YubiKey.

**−v**
be more verbose.

**−h**
display help.

**−V**
display version.

**YubiKey Neo only**

**−n** URI
program NFC NDEF URI.

**−t** text
program NFC NDEF text.

**YubiKey 3 and 4 only**

**−m** mode
set device configuration for the YubiKey. It is parsed in the form *mode:cr_timeout:autoeject_timeout* where mode is:

0
OTP device only.

1
CCID device only.

2
OTP/CCID composite device.

3
U2F device only.

4
OTP/U2F composite device.

5
U2F/CCID composite device.

6
OTP/U2F/CCID composite device. Add 80 to set MODE_FLAG_EJECT, for example: 81

cr_timeout is the timeout in seconds for the YubiKey to wait on button press for challenge response (default is 15)

autoeject_timeout is the timeout in seconds before the card is automatically ejected in mode 81

Removing OTP mode also disables communication between ykpersonalize and the YubiKey. Further mode changes will have to be done with ykneomgr (for CCID mode) or u2f−host (for U2F mode).

**YubiKey 3 and above**

**−S***0605...*

set the scanmap to be used with the YubiKey. It must be 45 unique bytes as 90 characters. Leave argument empty to reset to the YubiKey's default. The scanmap must be sent in the order:

cbdefghijklnrtuvCBDEFGHIJKLNRTUV0123456789!\t\r

The default scanmap in the YubiKey is:

06050708090a0b0c0d0e0f1115171819868587888 98a8b8c8d8e8f9195979899271e1f202122232425269e2b28

An example for simplified U.S. Dvorak would be:

0c110b071c180d0a0619130f120e09378c918b879c988d8a8699938f928e89b7271e1f202122232425269e2b28

Or for a French azerty keyboard (digits are shifted):

06050708090a0b0c0d0e0f1115171819868587888 98a8b8c8d8e8f9195979899a79e9fa0a1a2a3a4a5a6382b28

Or for a French BÉPO keyboard (French Dvorak):

0b140c0938363707130512330f0d16188b948c89b8b6b787938592b38f8d9698a79e9fa0a1a2a3a4a5a69c2b28

And a Turkish example (has a dotless i instead of usual i):

06050708090a0b340d0e0f1115171819868587888 98a8b8c8d8e8f9195979899271e1f202122232425269e2b28

Note that you must remove any whitespace present in these examples before using the values.

**YubiKey 5 and above**

**−D***0403...*

Set the deviceinfo to use with this YubiKey.

**YubiKey 2.3 and above**

**−u**

Update existing configuration, rather than overwriting. Only possible if the slot is configured as updatable.

**−x**

Swap configuration slot 1 and 2 inside the YubiKey. Only possible if both slots are configured as updatable.

## TICKET FLAGS

**tab−first**

Send a tab character as the first character. This is usually used to move to the next input field.

**append−tab1**

Send a tab character between the fixed part and the one−time password part. This is useful if you have the fixed portion equal to the user name and two input fields that you navigate between using tab.

**append−tab2**

Send a tab character as the last character.

**append−delay1**

Add a half−second delay before sending the one−time password part. This option is only valid for firmware 1.x and 2.x.

**append−delay2**
Add a half−second delay after sending the one−time password part. This option is only valid for firmware 1.x and 2.x.

**append−cr**
Add a carriage return after sending the one−time password part.

**YubiKey 2.0 firmware and above**

**protect−cfg2**
When written to configuration 1, block later updates to configuration 2. When written to configuration 2, prevent configuration 1 from having the lock bit set.

**YubiKey 2.1 firmware and above**

**oath−hotp**
Set OATH−HOTP mode rather than YubiKey mode. In this mode, the token functions according to the OATH−HOTP standard.

**YubiKey 2.2 firmware and above**

**chal−resp**
Set challenge−response mode.

## CONFIGURATION FLAGS

**send−ref**
Send a reference string of all 16 modhex characters before the fixed part. When combined with **−ostrong−pw2** this sends a *!* before the rest of the string.

**pacing−10ms**
Add a 10ms delay between key presses.

**pacing−20ms**
Add a 20ms delay between key presses.

**static−ticket**
Output a fixed string rather than a one−time password. The password is still based on the AES key and should be hard to guess and impossible to remember.

**YubiKey 1.x firmware only**

**ticket−first**
Send the one−time password rather than the fixed part first.

**allow−hidtrig**
Allow trigger through HID/keyboard by pressing caps−, num or scroll−lock twice. Not recommended for security reasons.

**YubiKey 2.0 firmware and above**

**short−ticket**
Limit the length of the static string to max 16 digits. This flag only makes sense with the **−ostatic−ticket** option. When **−oshort−ticket** is used without **−ostatic−ticket** it will program the YubiKey in "scan−code mode", in this mode the key sends the contents of fixed, uid and key as raw keyboard scancodes. For example, by using the fixed string *h:8b080f0f122c9a12150f079e* in this mode it will send *Hello World!* on a qwerty keyboard. This mode sends raw scan codes, so output will differ between keyboard layouts.

**strong−pw1**
Upper−case the two first letters of the output string. This is for compatibility with legacy systems that enforce both uppercase and lowercase characters in a password and does not add any security.

**strong−pw2**
Replace the first eight characters of the modhex alphabet with the numbers 0 to 7. Like **−ostrong−pw1**, this is intended to support legacy systems.

**man−update**
Enable user−initiated update of the static password. Only makes sense with the **−ostatic−ticket** option. This is only valid for firmware 2.x.

**YubiKey 2.1 firmware and above**

**oath−hotp8**
Generate an 8−digit HOTP rather than a 6−digit one.

**oath−fixed−modhex1**
Send the first byte of the fixed part as modhex.

**oath−fixed−modhex2**
Send the first two bytes of the fixed part as modhex.

**oath−fixed−modhex**
Send the fixed part is as modhex.

**oath−id**=m:OOTTUUUUUUUU
Configure OATH token id with a provided value. See description of this option under the 2.2 section for details, but note that a YubiKey 2.1 key can't report its serial number and thus a token identifier value must be specified.

**YubiKey 2.2 firmware and above**

**chal−yubico**
Yubico OTP challenge−response mode.

**chal−hmac**
Generate HMAC−SHA1 challenge responses.

**hmac−lt64**
Calculate HMAC on less than 64 bytes input. Whatever is in the last byte of the challenge is used as end of input marker (backtracking from end of payload).

**chal−btn−trig**
The YubiKey will wait for the user to press the key (within 15 seconds) before answering the challenge.

**serial−btn−visible**
The YubiKey will emit its serial number if the button is pressed during power−up. This option is only valid for the 2.x firmware line.

**serial−usb−visible**
The YubiKey will indicate its serial number in the USB iSerial field. This option is not available in the 3.0 and 3.1 firmwares.

**serial−api−visible**
The YubiKey will allow its serial number to be read using an API call.

**oath−id**[=m:OOTTUUUUUUUU]
Configure OATH token id with a provided value, or if used without a value use the standard YubiKey token identifier.

The standard OATH token id for a Yubico YubiKey is (modhex) OO=ub, TT=he, (decimal) UUUUUUUU=serial number.

The reason for the decimal serial number is to make it easy for humans to correlate the serial number on the back of the YubiKey to an entry in a list of associated tokens for example. Other encodings can be accomplished using the appropriate oath−fixed−modhex options.

Note that the YubiKey must be programmed to allow reading its serial number, otherwise automatic token id creation is not possible.

See section "5.3.4 − OATH−HOTP Token Identifier" of the YubiKey manual http://yubico.com/files/YubiKey_manual−2.0.pdf for further details.

**YubiKey 2.3 firmware and above**

**use−numeric−keypad**
Send scancodes for numeric keypad keypresses when sending digits − helps with some keyboard layouts. This option is only valid for the 2.x firmware line.

**fast−trig**
Faster triggering when only configuration 1 is available. This option is always in effect on firmware versions 3.0 and above.

**allow−update**
Allow updating (or swapping) of certain parameters in a configuration at a later time.

**dormant**
Hides/unhides a configuration stored in a YubiKey.

**YubiKey 2.4/3.1 firmware and above**

**led−inv**
Inverts the behaviour of the led on the YubiKey.

**OATH−HOTP Mode**

When using OATH−HOTP mode, a HMAC key of 160 bits (20 bytes, 40 chars of hex) can be supplied with **−a**.

**Challenge−response Mode**

In **CHAL−RESP** mode, the token will NOT generate any keypresses when the button is pressed (although it is perfectly possible to have one slot with a keypress−generating configuration, and the other in challenge−response mode). Instead, a program capable of sending USB HID feature reports to the token must be used to send it a challenge, and read the response.

**Modhex**

Modhex is a way of writing hex digits where the “digits” are chosen for being in the same place on most keyboard layouts. To convert from hex to modhex, you can use:

```
tr "[0123456789abcdef]" "[cbdefghijklnrtuv]"
```

To convert the other way, use:

```
tr "[cbdefghijklnrtuv]" "[0123456789abcdef]"
```

**EXAMPLES**

Programming for YubiCloud:

```
ouid=‘dd if=/dev/urandom 2>/dev/null | tr −d '[:upper:]' | tr −cd '[:xdigit:]' | fold −w12 | head −1‘
ofixed=ff‘dd if=/dev/urandom 2>/dev/null | tr −d '[:upper:]' | tr −cd '[:xdigit:]' | fold −w10 | head −1‘
ykpersonalize −1 −ouid=h:$ouid −ofixed=h:$ofixed
```

This will program a key with a random 6 byte uid and a 12 character fixed string starting with vv. This is suitable for upload to YubiCloud at https://upload.yubico.com/

**BUGS**

Report ykpersonalize bugs in the issue tracker https://github.com/Yubico/yubikey−personalization/issues

**SEE ALSO**

The ykpersonalize home page https://developers.yubico.com/yubikey−personalization/

YubiKeys can be obtained from Yubico http://www.yubico.com/