OpenSSH + FIDO workshop

Joost van Dijk | Yubico

Open Source Summit Europe - August 2025

© 2025 Yubico

yubico

About this workshop

- Exercises:
  - https://github.com/YubicoLabs/fido-openssh-workshop

- Basic OpenSSH knowledge assumed
- Some demos require a GitHub or GitLab account

- If you need one: grab a FIDO Security Key!

© 2025 Yubico

yubico

Before we start…

- If you haven't already, download/install:
  - OpenSSH client (version 8.2+, ideally 8.9+)
    `$ ssh –V`
  - Python 3.9+
    `$ python3 –V`
  - Docker Desktop (or something similar)
    https://www.docker.com/products/docker-desktop/
  - An Ubuntu image (latest)
    `$ docker pull ubuntu:latest`
  - FIDO command-line tools
    https://developers.yubico.com/libfido2/
    `$ <pkg-mgr> install libfido2`
    `(fido2-tools on debian/ubuntu)`

© 2025 Yubico

yubico

Overview

- Intro - what is FIDO?
  - Tools: libfido2
- Hardware-backed SSH private keys
  - Exercise: sign in using hardware-backed keys
- Generating SSH signatures using security keys
  - Exercise: git commit signing
- Storing SSH certificates
  - Exercise: largeBlobs
- Device attestation and the FIDO metadata service
  - Exercise: generating attestations
- ssh-agent with security keys
  - Exercise: more secure agent-forwarding

© 2025 Yubico

yubico

Introduction

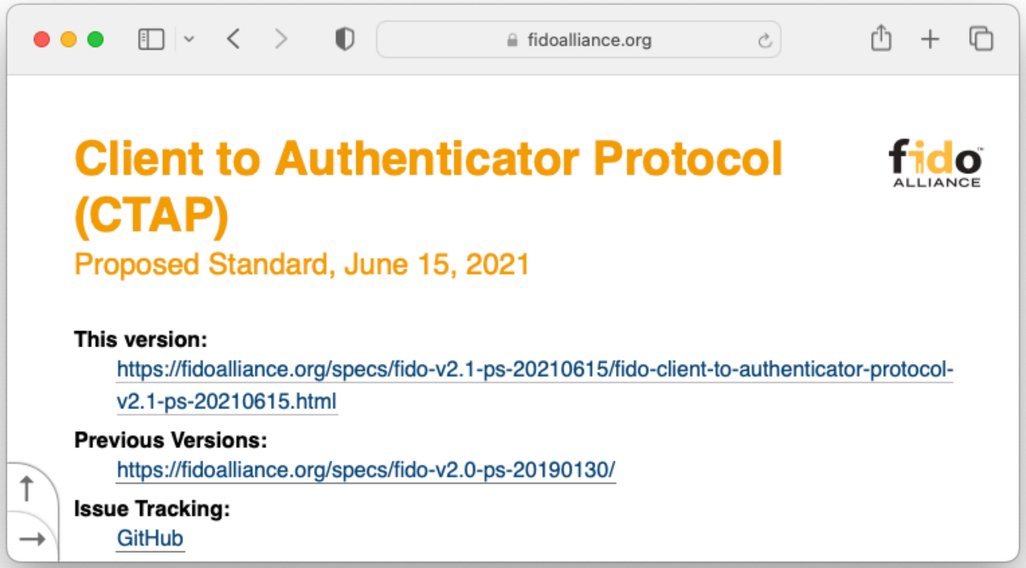
FIDO2 - concepts

© 2025 Yubico

yubico

FIDO2

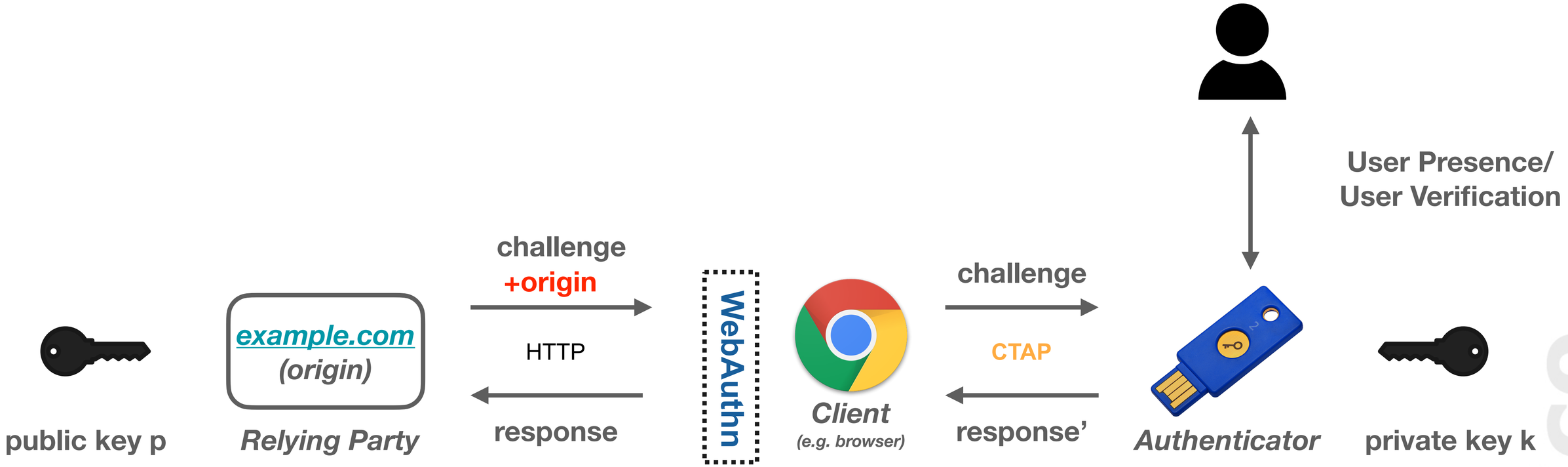
- Specifications:
  - CTAP - using a FIDO authenticator from a client (e.g. a browser)
  - Webauthn: API for using FIDO credentials in web applications



© 2025 Yubico

yubico

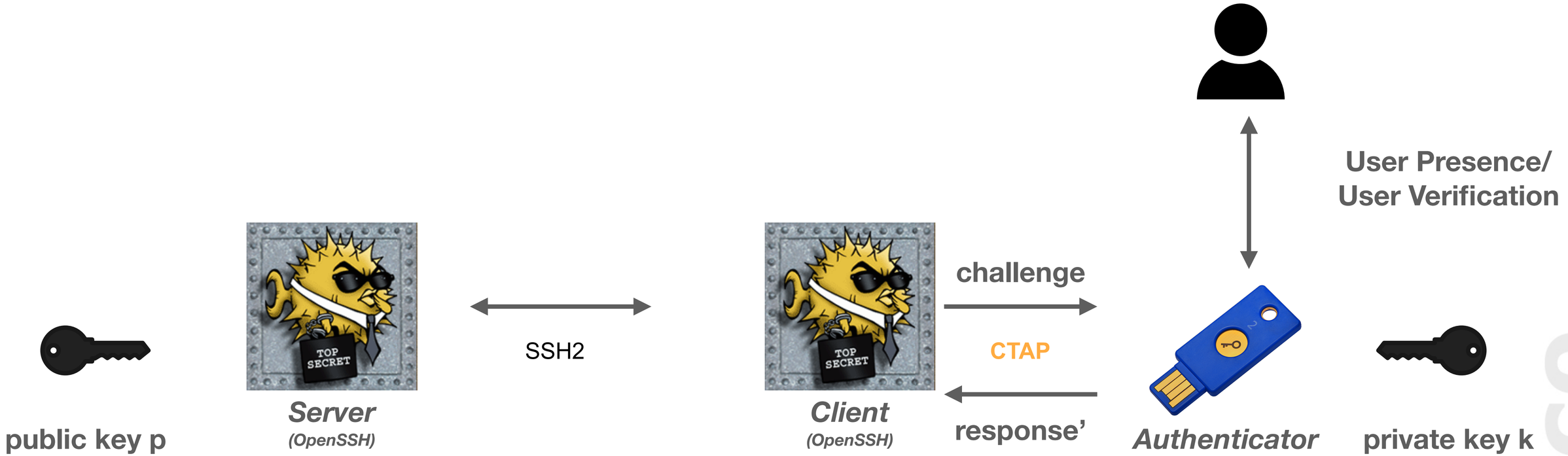
CTAP and Webauthn (simplified)



© 2025 Yubico

yubico

CTAP and SSH



© 2025 Yubico

Why OpenSSH + FIDO?

- SSH Private Key
  - Generated and stored on secure hardware (a FIDO **Security Key**)
  - Cannot be exported: a possession factor for MFA
- Before any private key operation
  - Option to require a button touch to prove **User Present**
  - Option to require PIN entry for **User Verification**
- Store associated SSH certificates on a portable device
- Cryptographically prove Security Key provenance (make and model)
- Simpler than using OpenPGP or PKCS#11

© 2025 Yubico

yubico

CTAP: operations (simplified)

uID	credId	pub
jdoe	0xa1b2	0x41…

(RP user store)

RP

Client

Authenticator

credId	uID	rpID	priv
0xa1b2	jdoe	eg.com	0xf1…

(authR credential mapping)

```
                                    makeCredential(rpId, uID, cData, …)
                                    (priv,pub) = genKeyPair()
                                    credId = genCredId(…)
                                    map {credId, uID, rpID} → priv
                                    signature = sign(…)
                                    { credId, pub, signature, … }
verify(resp,signature)
store[uID] = {credId, pub}

…

{credId, pub} = store[uID]
                                    getAssertion(credID, rpId, cData, …)
                                    priv = lookup {credID}
                                    signature = sign(priv, rpId, cData…)
                                    { signature, … }
verify(pub, resp, signature)
```

Authenticators

- ***cross-platform* Authenticator**
  also called *roaming* authenticator
  example: a USB security key

- ***Platform* Authenticator**
  Built into user's device
  example: a built-in fingerprint sensor

FaceID

TouchID

Windows Hello

- Roaming authenticators can use different *transports*: USB, NFC, BLE, hybrid

- Note: a single authenticator can store multiple FIDO credentials!

© 2025 Yubico

yubico

FIDO Security Keys: cross-platform authenticators

yubico

© 2025 Yubico

User Present vs User Verified

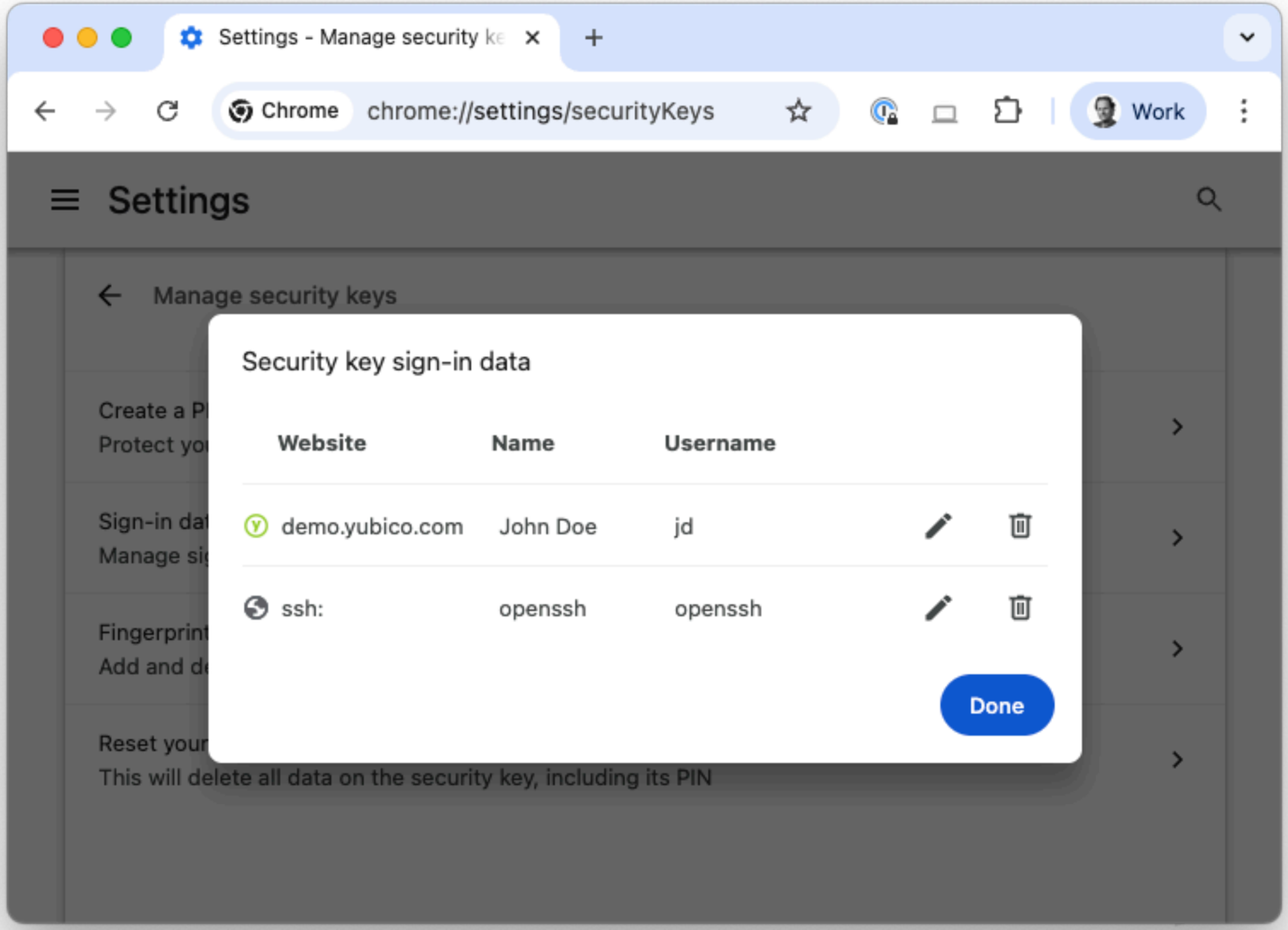
- User Present (UP): User is physically present
  - On a security key: a button touch
- User Verified (UV): User is verified, i.e. user authenticated to the Security Key
  - On a security key: entering a PIN
    (on biometric models: touching a built-in fingerprint scanner)
  - On a platform authenticator: typically a biometric
    (eg Touch ID/Face ID)

© 2025 Yubico

yubico

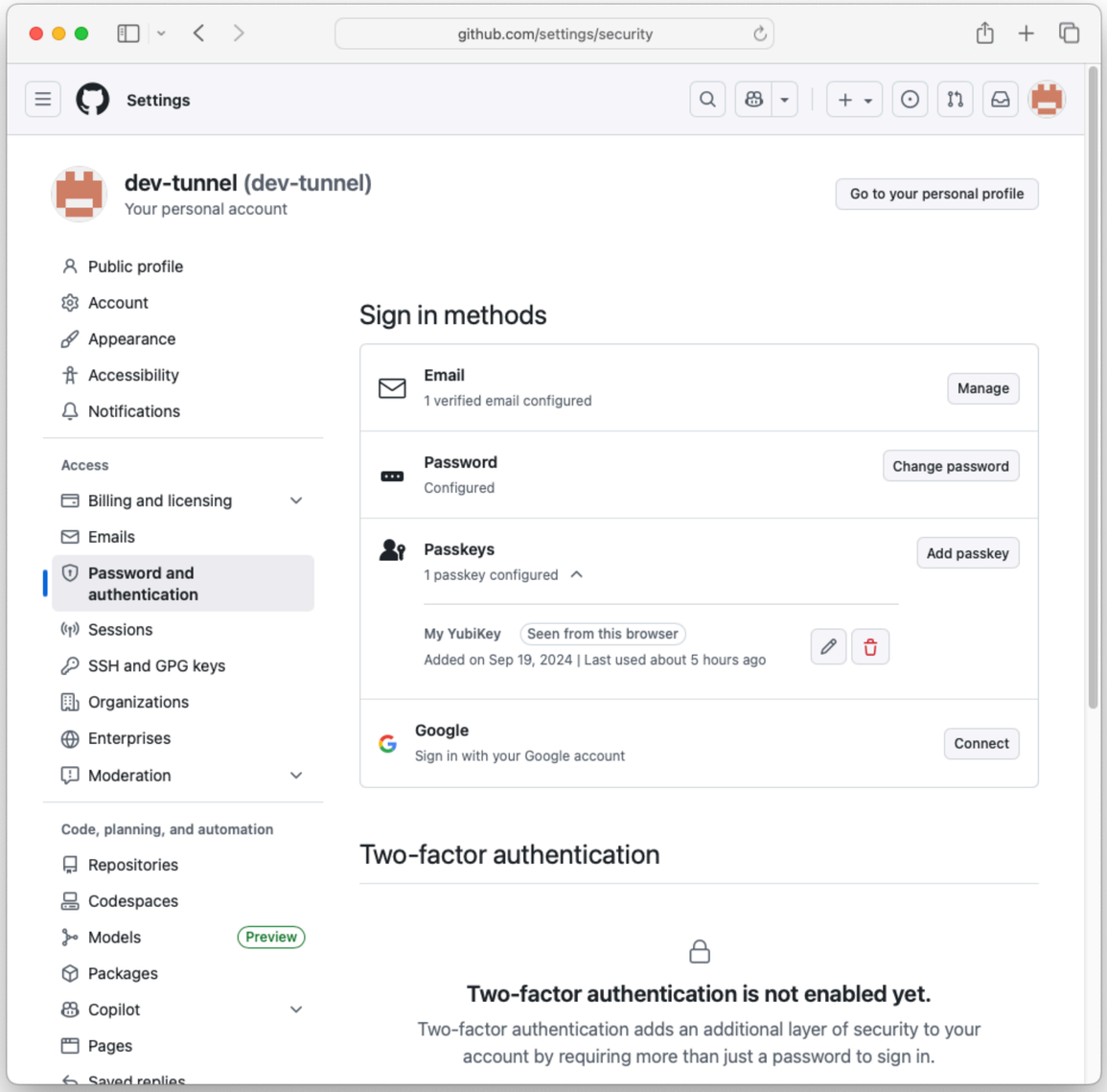
Resident vs non-resident Keys

- A Credential can be:
  - **Resident**: stored in a security key's internal memory
  - **Non-resident**: encrypted with a wrapping key and stored off-key
- Resident keys are called *Discoverable Credentials* since CTAP v2.1
- A security key can store a limited amount of resident keys



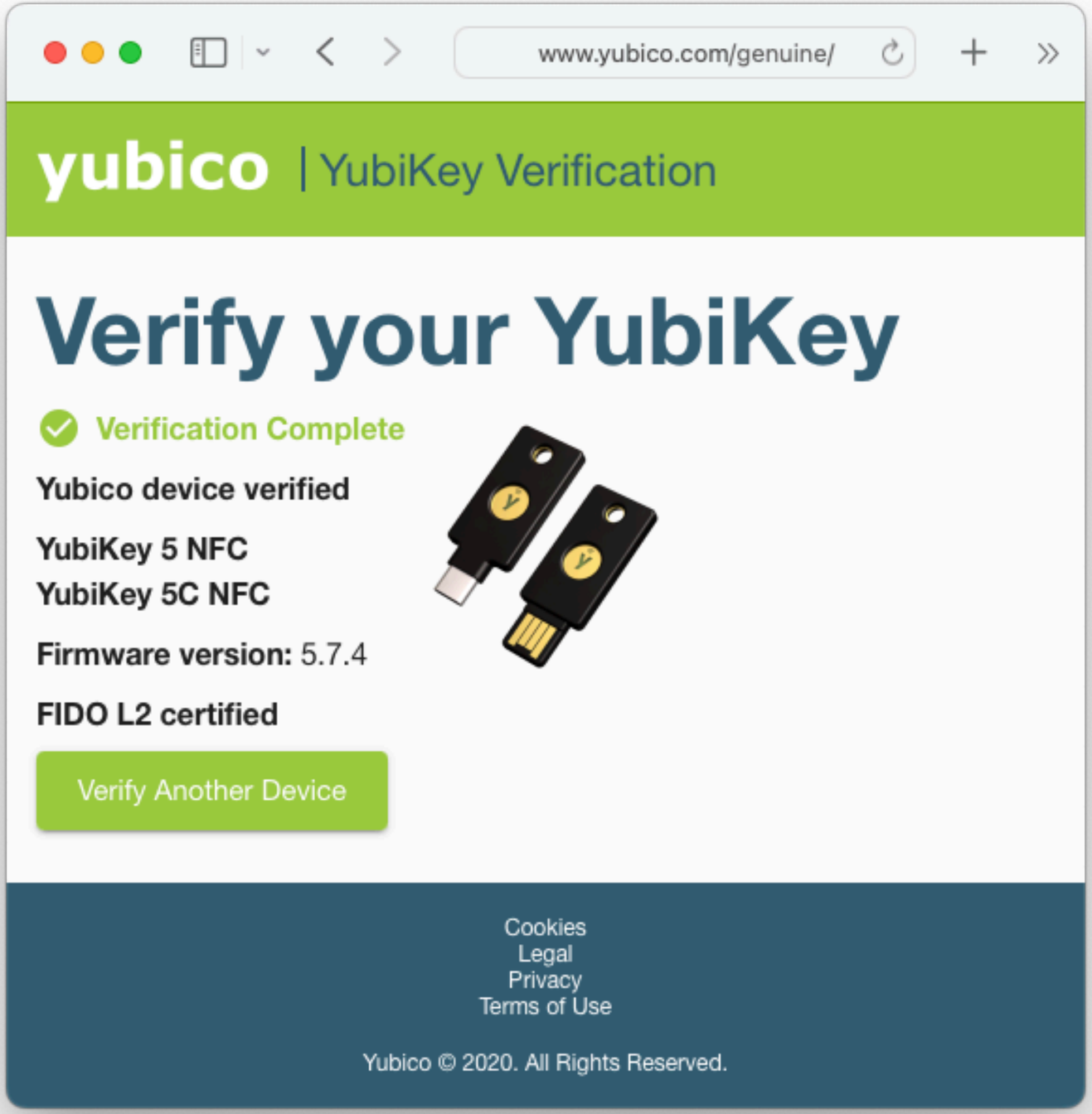
© 2025 Yubico

Passwordless sign-in with passkeys



© 2025 Yubico

Attestation



© 2025 Yubico

libfido2

- https://github.com/Yubico/libfido2
- Library and command-line tools
- Communicate with a FIDO device over USB or NFC
- Supports both FIDO U2F (CTAP 1) and FIDO2 (CTAP 2) protocols
- Works on Linux, macOS, Windows, OpenBSD, and FreeBSD
- Bindings for .NET, Go, Perl, Rust
- Also available:
  - https://github.com/Yubico/python-fido2

© 2025 Yubico

yubico

Get Info

```
$ fido2-token -L
ioreg://4296874505: vendor=0x1050, product=0x0407 (Yubico YubiKey OTP+FIDO+CCID)
$ fido2-token -I ioreg://4296874505
proto: 0x02
major: 0x05
minor: 0x07
build: 0x04
caps: 0x05 (wink, cbor, msg)
version strings: U2F_V2, FIDO_2_0, FIDO_2_1_PRE, FIDO_2_1
extension strings: credProtect, hmac-secret, largeBlobKey, credBlob, minPinLength
transport strings: nfc, usb
algorithms: es256 (public-key), eddsa (public-key), es384 (public-key)
aaguid: d7781e5de35346aaafe23ca49f13332a
options: rk, up, noplat, noalwaysUv, credMgmt, authnrCfg, clientPin, largeBlobs, pinUvAuthToken, setMinPINLength, makeCredUvNotRqd
fwversion: 0x50704
maxmsgsiz: 1536
maxcredcntlst: 8
maxcredlen: 128
maxcredblob: 32
maxlargeblob: 4096
maxrpids in minpinlen: 1
remaining rk(s): 98
minpinlen: 4
pin protocols: 2, 1
pin retries: 8
pin change required: false
uv retries: undefined
```

© 2025 Yubico

Resources

- Developer Program: https://dev.yubi.co/
- CTAP 2.1 spec: https://fidoalliance.org/specs/fido-v2.2-ps-20250228/
- FIDO dev forum: https://groups.google.com/a/fidoalliance.org/g/fido-dev
- libfido2: https://github.com/Yubico/libfido2
- FIDO Metadata:
  - https://fidoalliance.org/metadata/
  - https://opotonniee.github.io/fido-mds-explorer/
- SSH support for security keys:
  - https://github.com/openssh/openssh-portable/blob/master/PROTOCOL.u2f

© 2025 Yubico

yubico

Questions?

© 2025 Yubico

yubico