
U 2 can U2F

Hello!

Rob N ★

Email: robn@fastmail.com
Twitter: @robn
Github: @robn

https://robn.io/u2f-lca-2017

Two-factor authentication

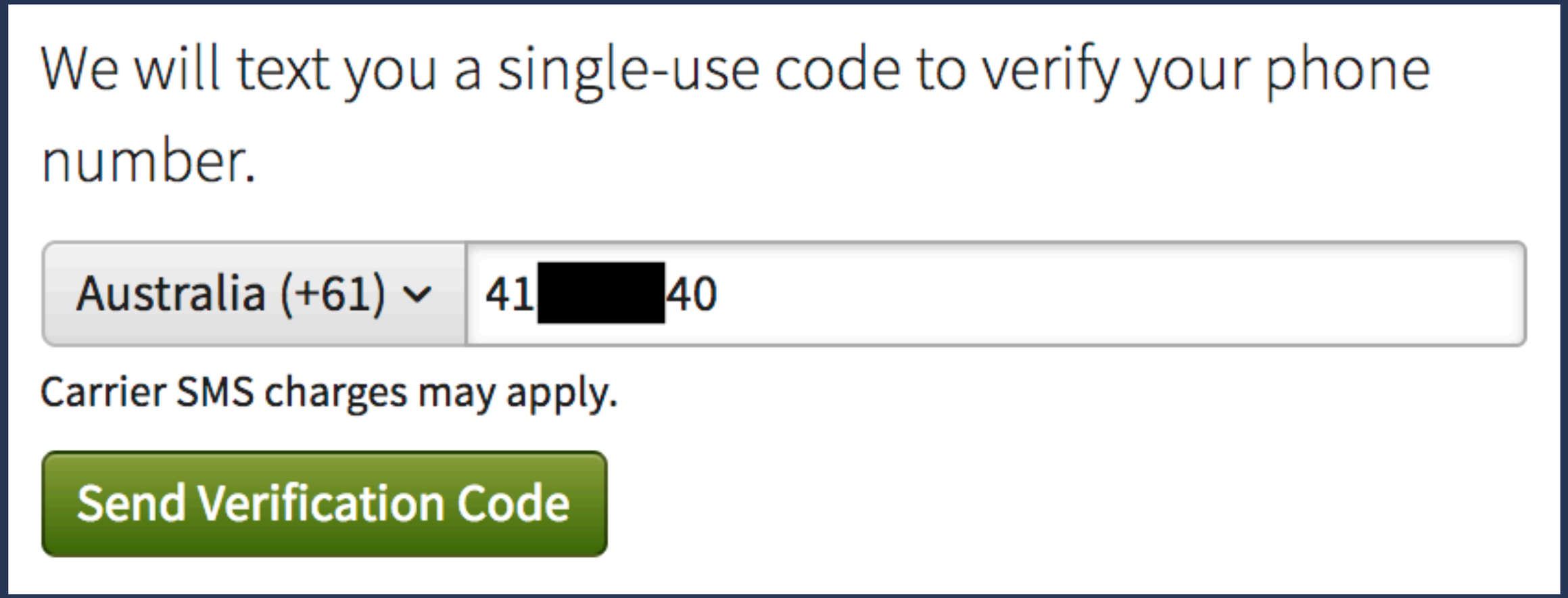
- Something you know
  - Password
- Something you have
  - SMS (phone)
  - TOTP (phone, standalone)
  - Proprietary/enterprise/corporate things (standalone)

Two-factor authentication

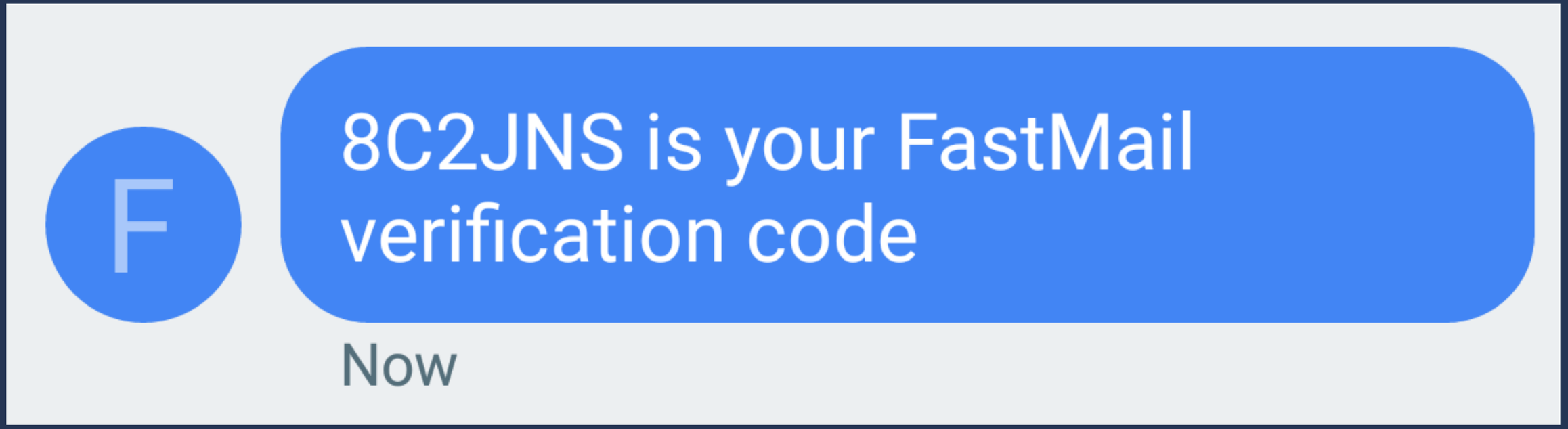
- Registration
  - Connect a second factor to an existing account
- Authentication
  - Verify that the user has the second factor


Not a
live demo

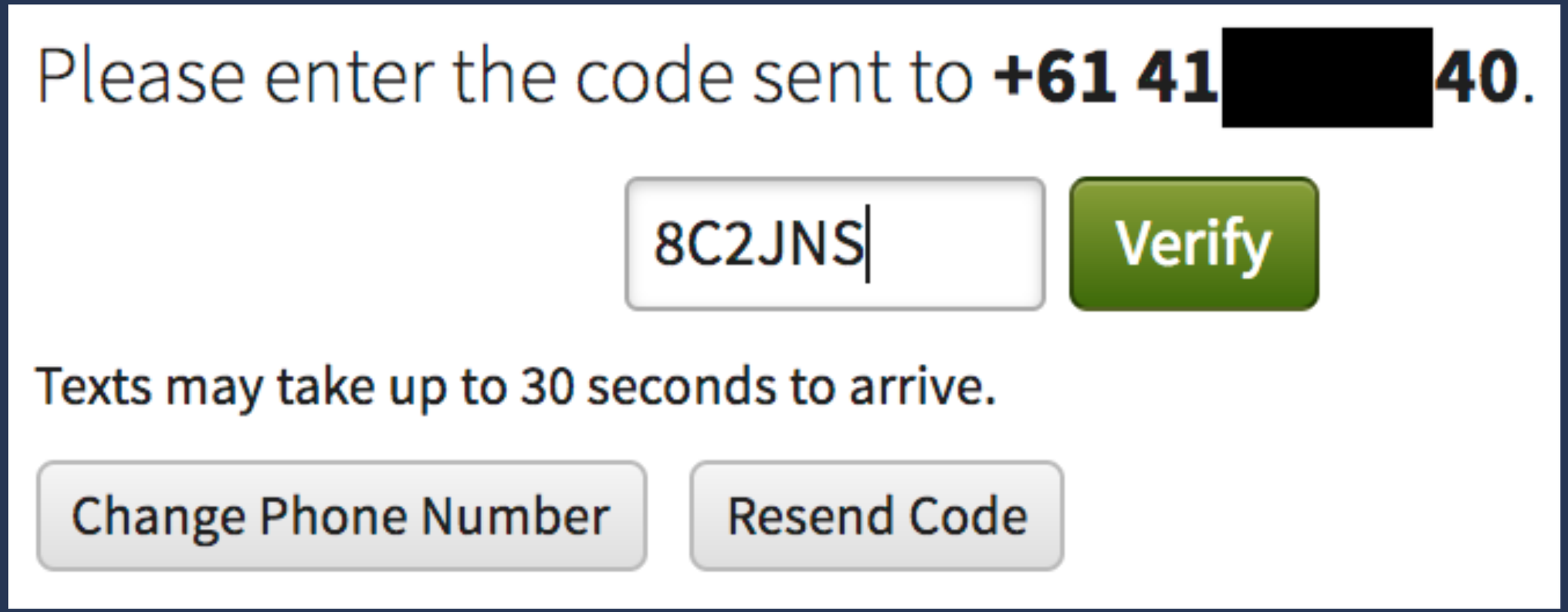
SMS registration



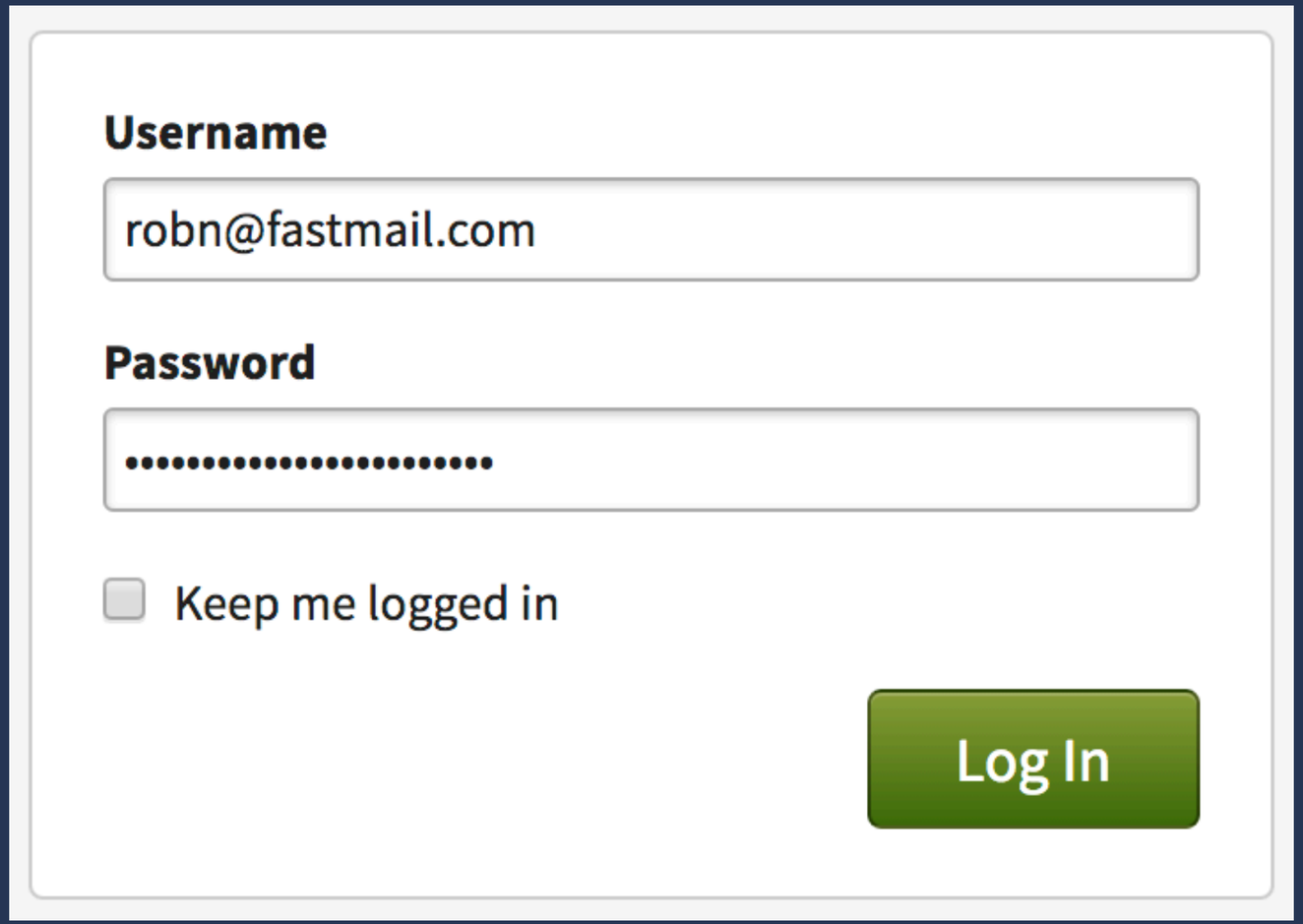
SMS registration



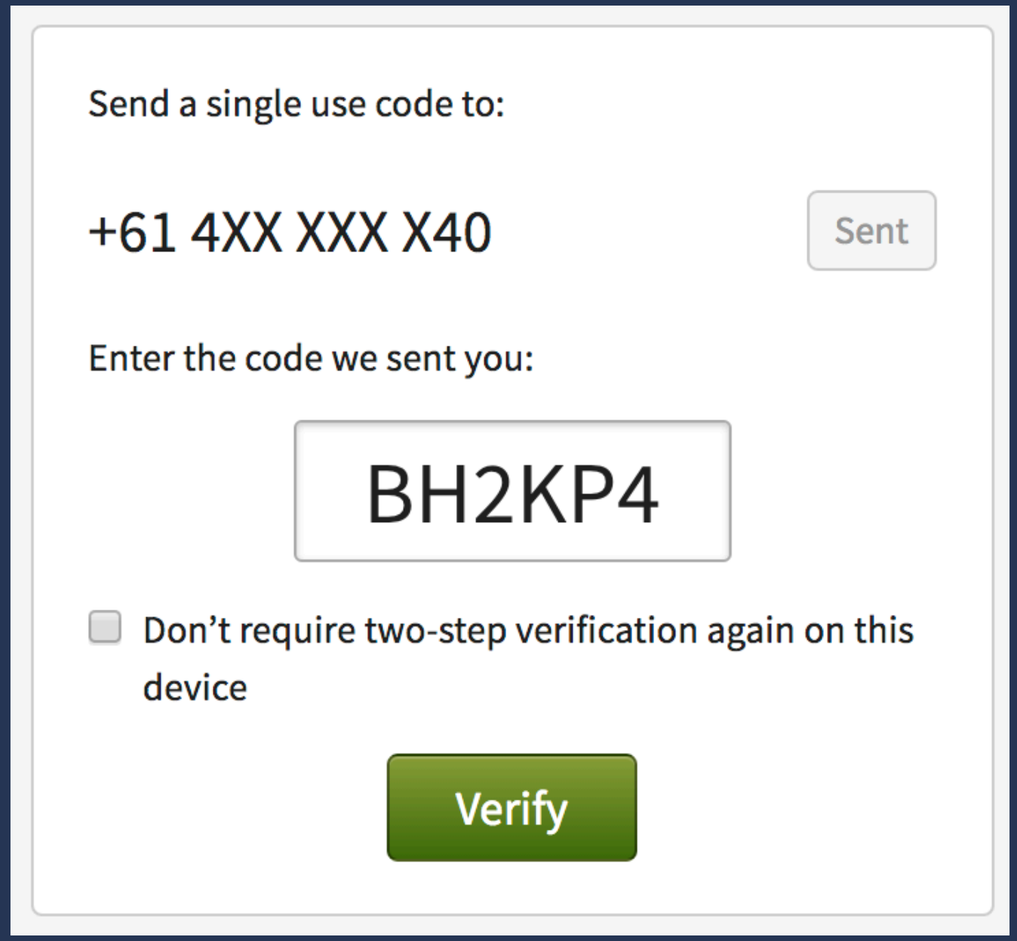
SMS registration



SMS authentication



SMS authentication

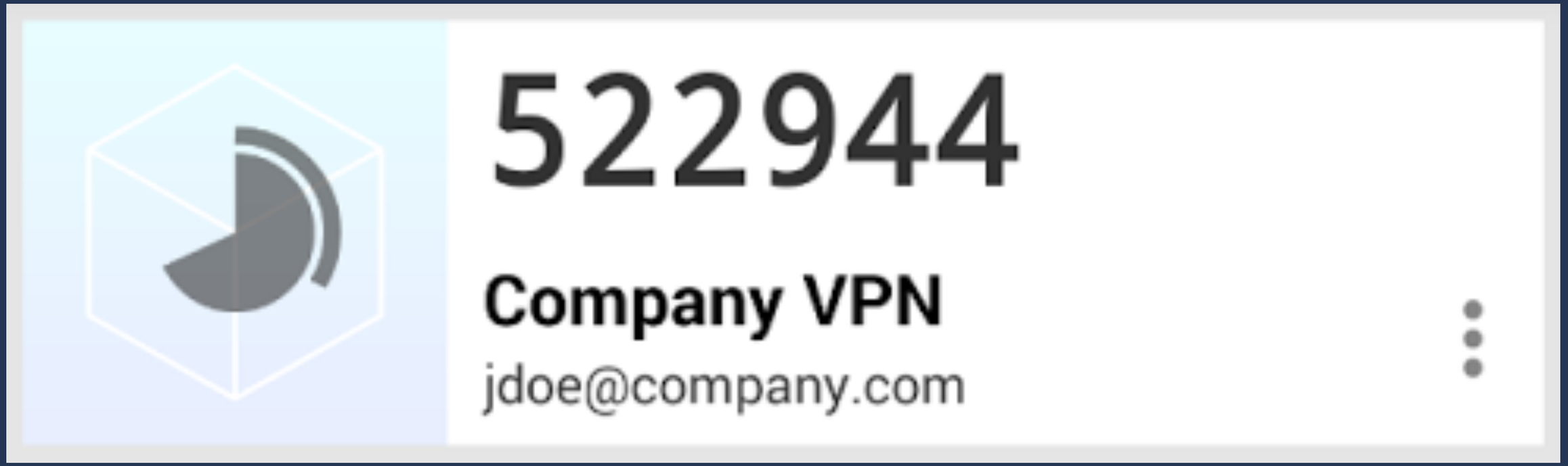


TOTP registration

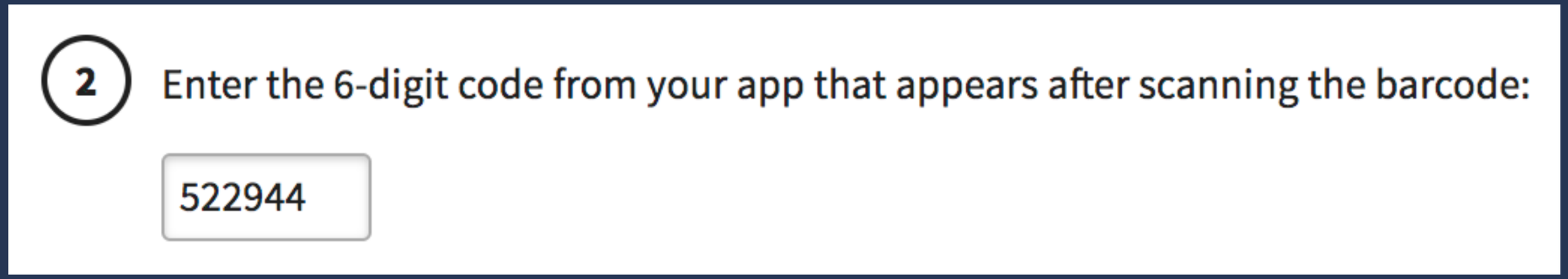
1 Scan this barcode in your authenticator app (Help):

Can’t use the barcode? Type in this secret key instead: `tvdn4prjecbn6xvaybs2qxlyoi`

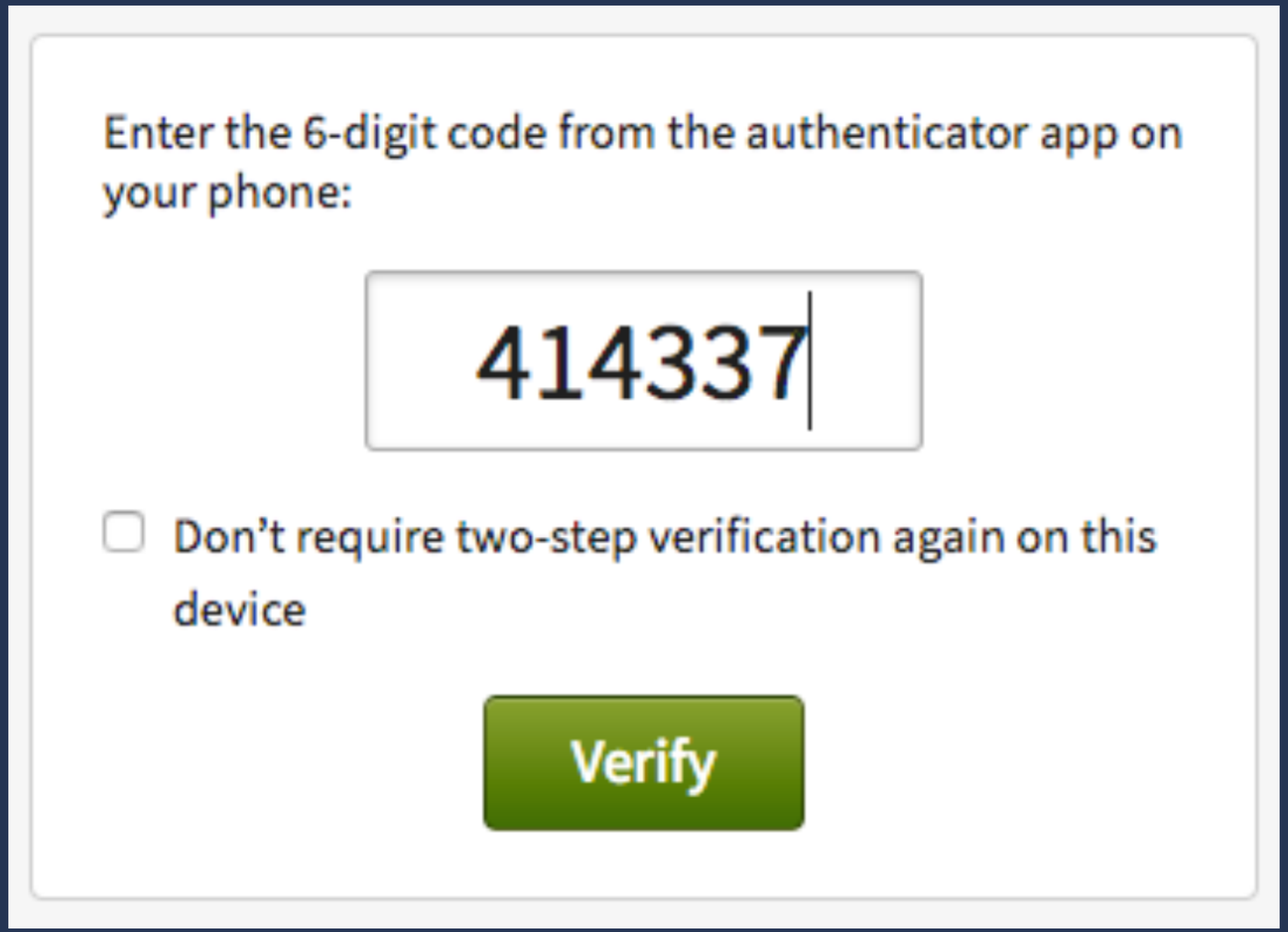
TOTP registration



TOTP registration

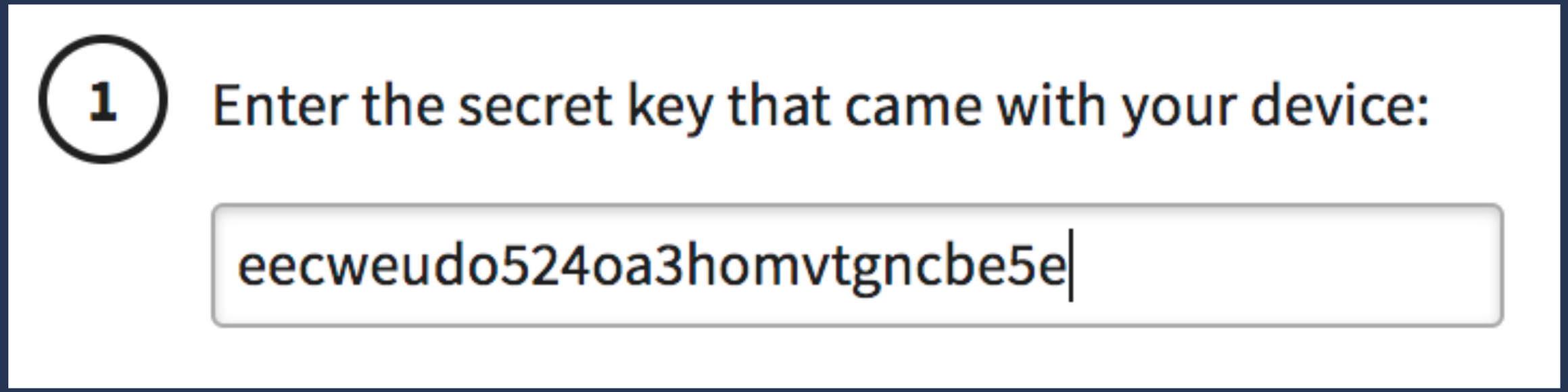


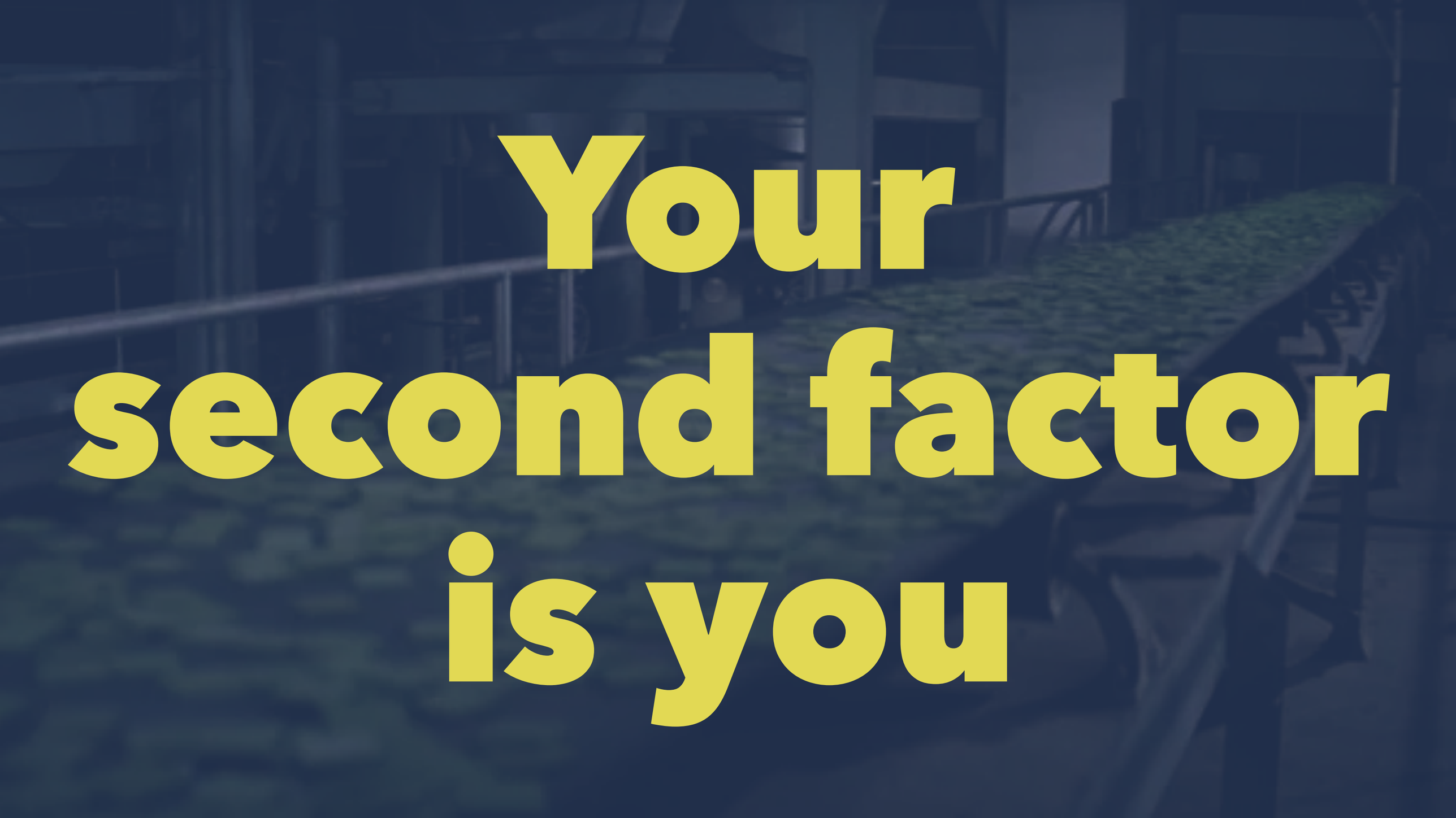
TOTP authentication

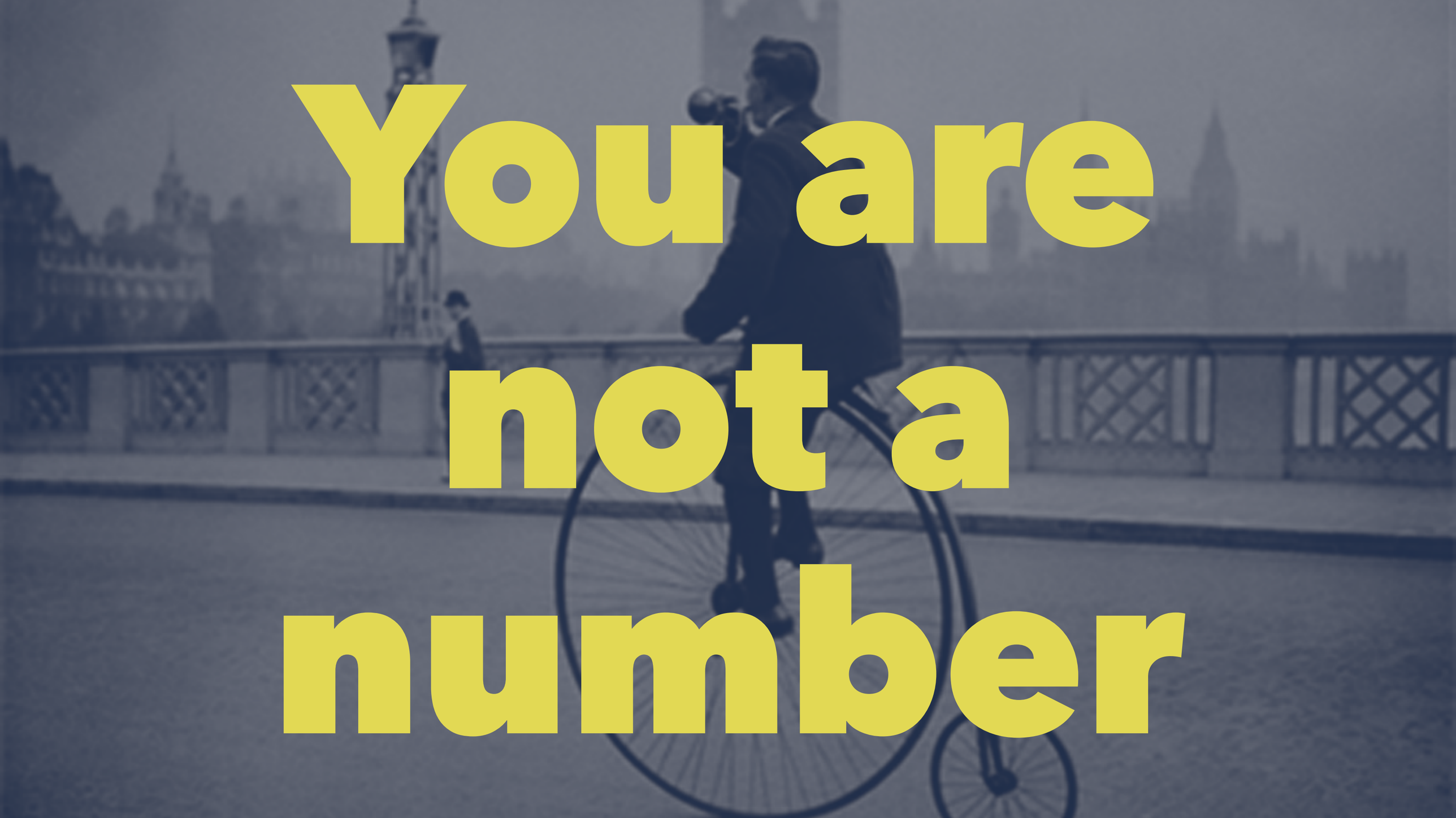


TOTP hardware token

TOTP hardware token




Your
second factor
is you


You are
not a
number


Phishing


Dignitaries
ENTERTAIN
Onions
CARAMELIZE
Disbelief
SUSPEND
Usability
Heliоscissor Scanner
Beeping Trapezoid
Previous Cr
Direct Orders
DISOBEY
EULOGIZE

Universal two-factor

Universal two-factor

- Open standard for a two-factor device
  - FIDO Alliance
  - Multiple hardware manufacturers
  - USB, NFC or Bluetooth
  - Multiple server and host implementations

Universal two-factor

- Active device
  - Participates directly in authentication process
  - Public key crypto

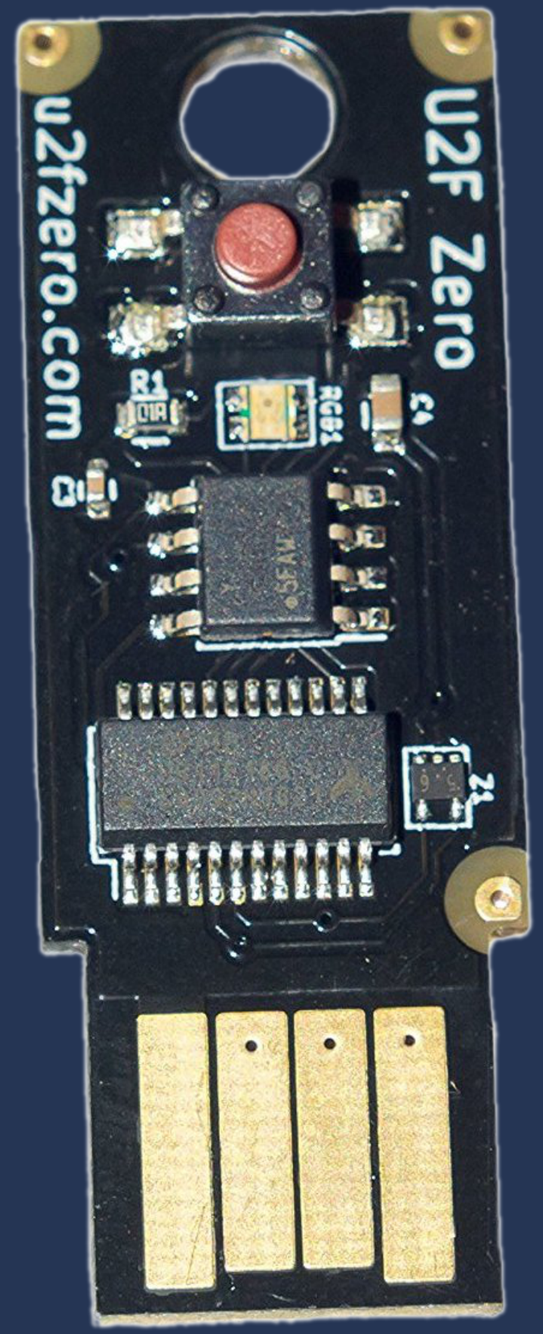
Universal two-factor

- One device, multiple sites
- Phishing protection
- Cloning protection
- Device properties


y
yubico
Keydo
FIDO U2F Security Key
fido U2F Ready
HYPERSECU
HyperFIDO.com
Nitrokey U2F
u2fzero.com
U2F Zero

U2F Zero

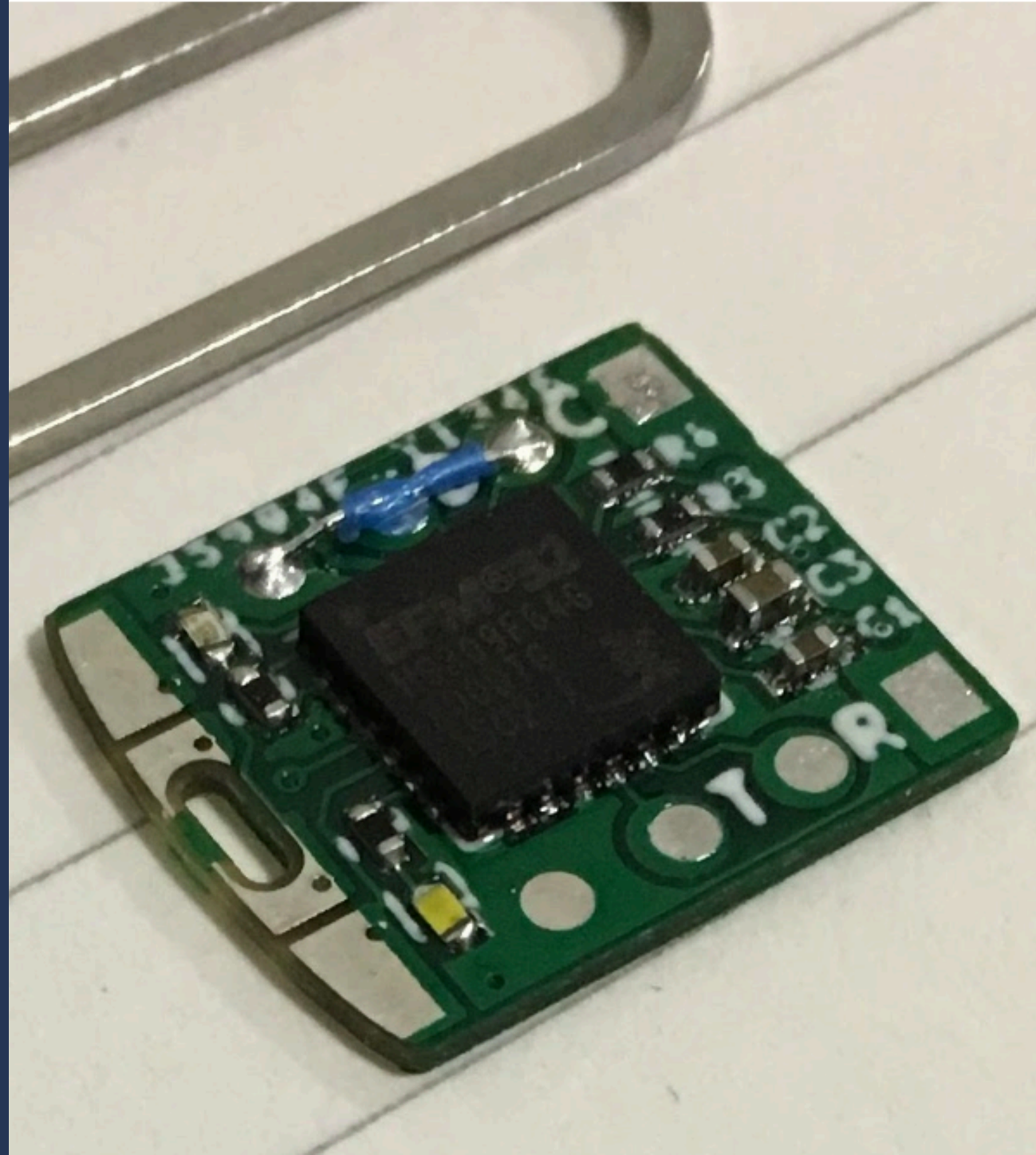
https://u2fzero.com/



Tomu

https://tomu.im/

- A tiny computer that fits inside your USB port
  - ARM CPU
  - Two buttons
  - Two LEDs
- Open Hardware Certified

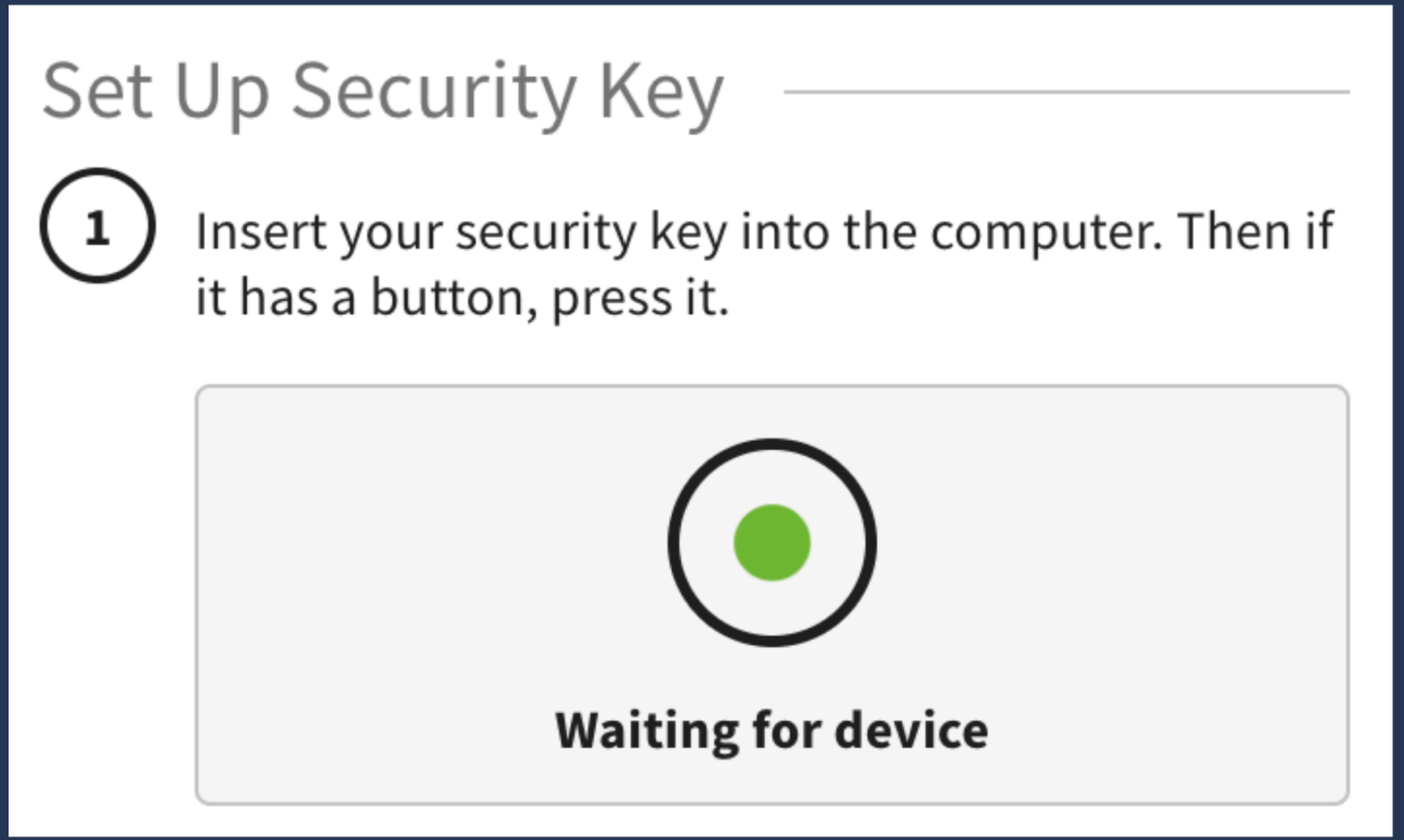


Tomu

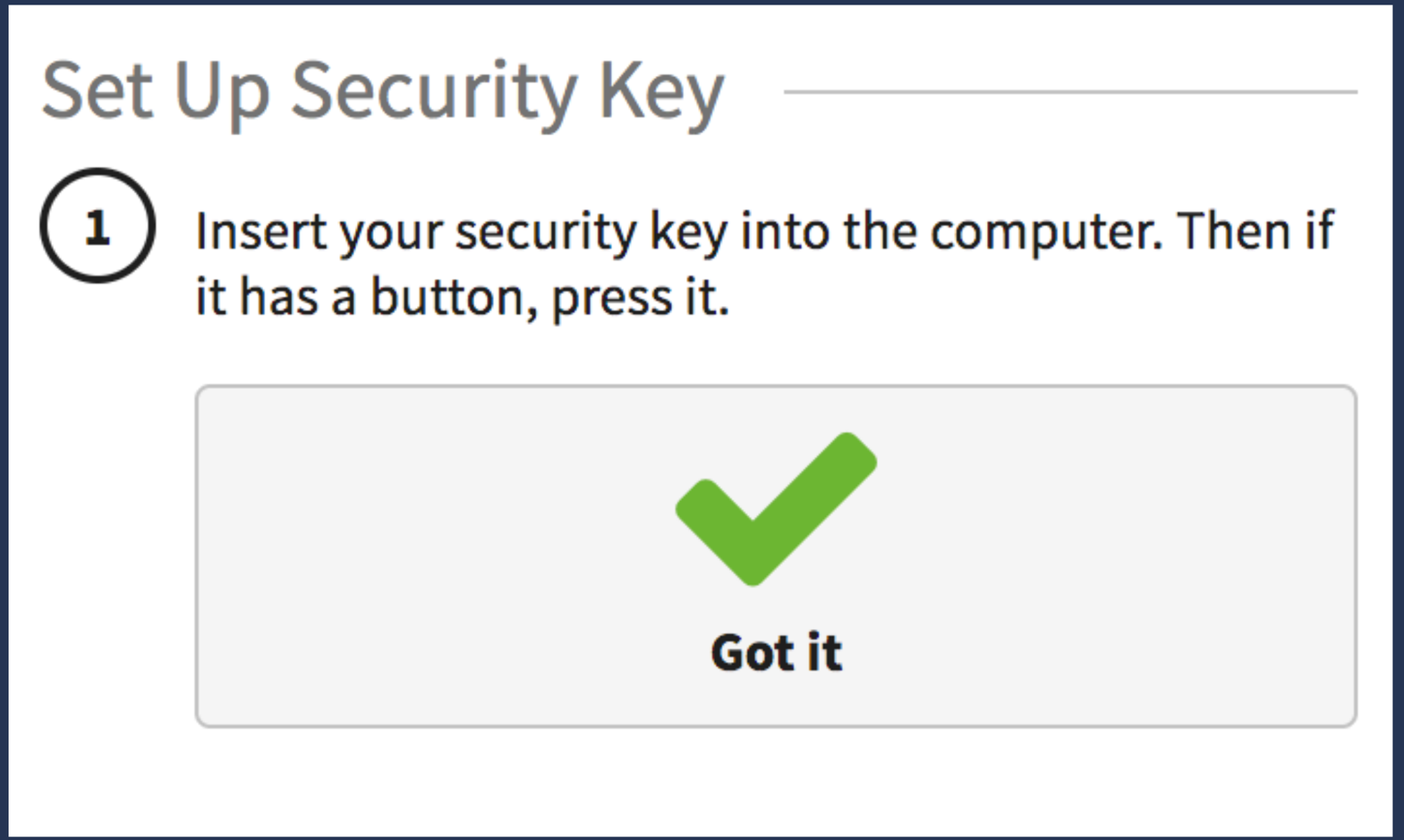
https://tomu.im/

- Designed by Tim Ansell
- Planned to be a U2F device
- Needs software
- You can help!

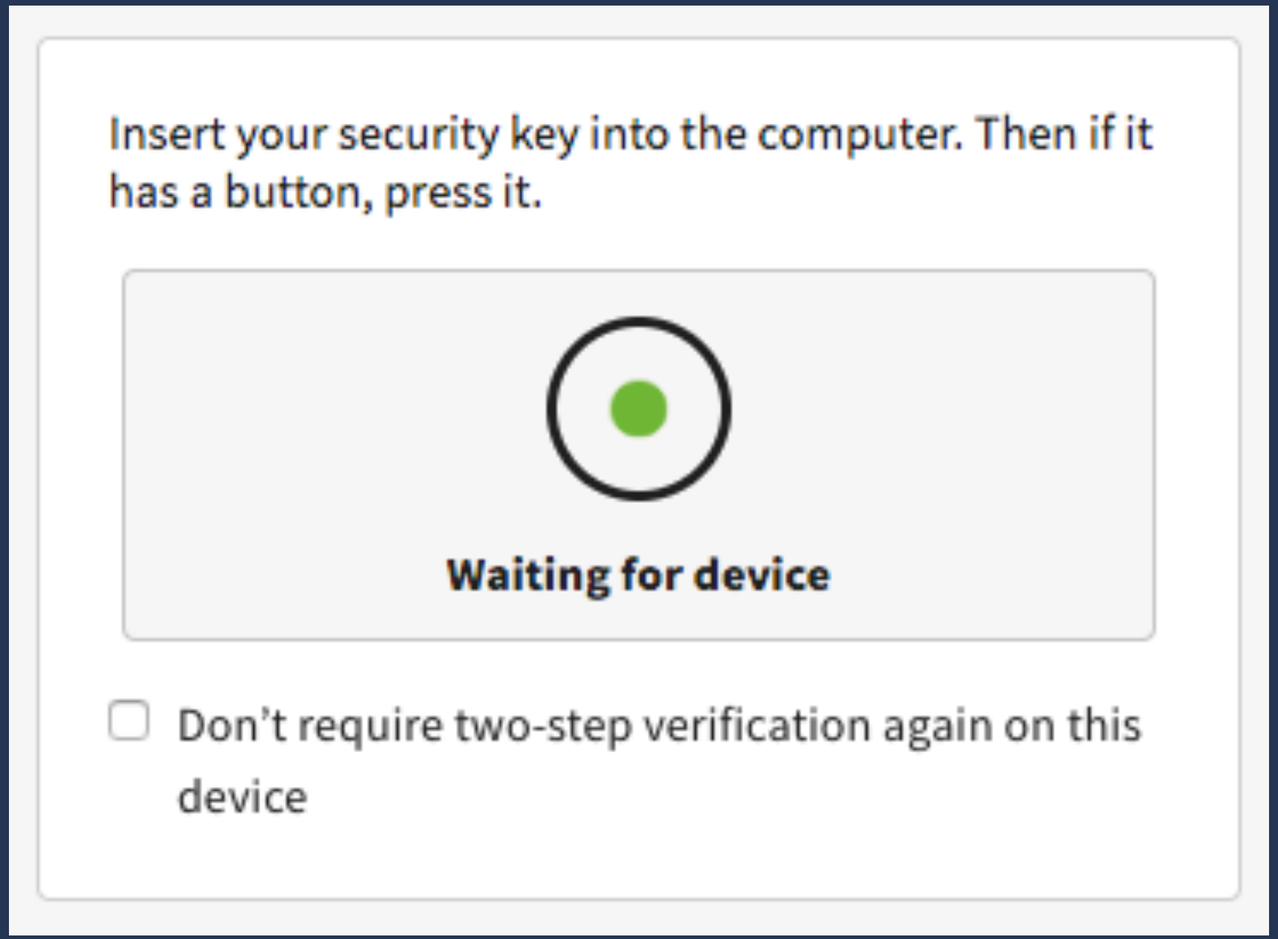
U2F registration



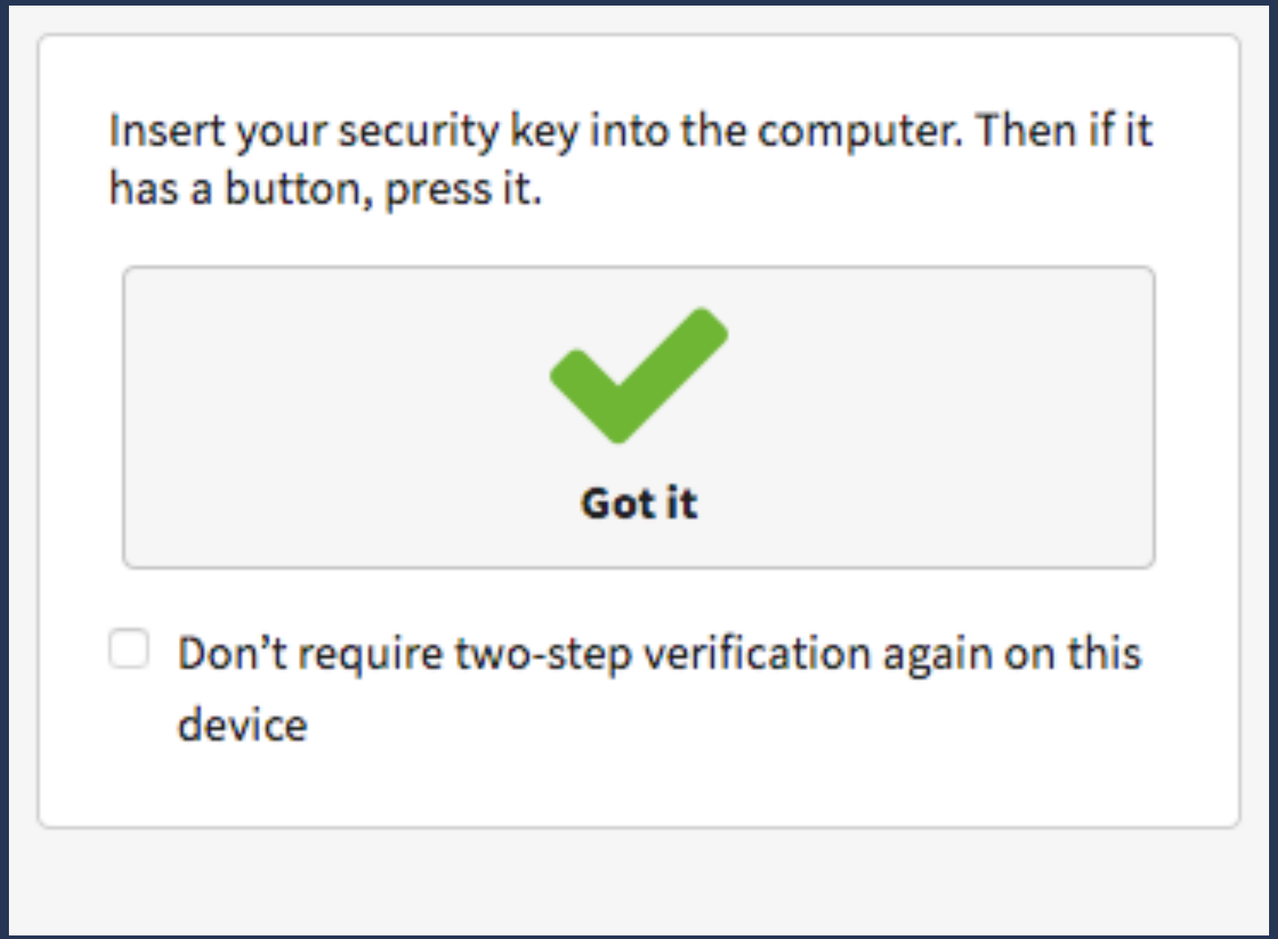
U2F registration

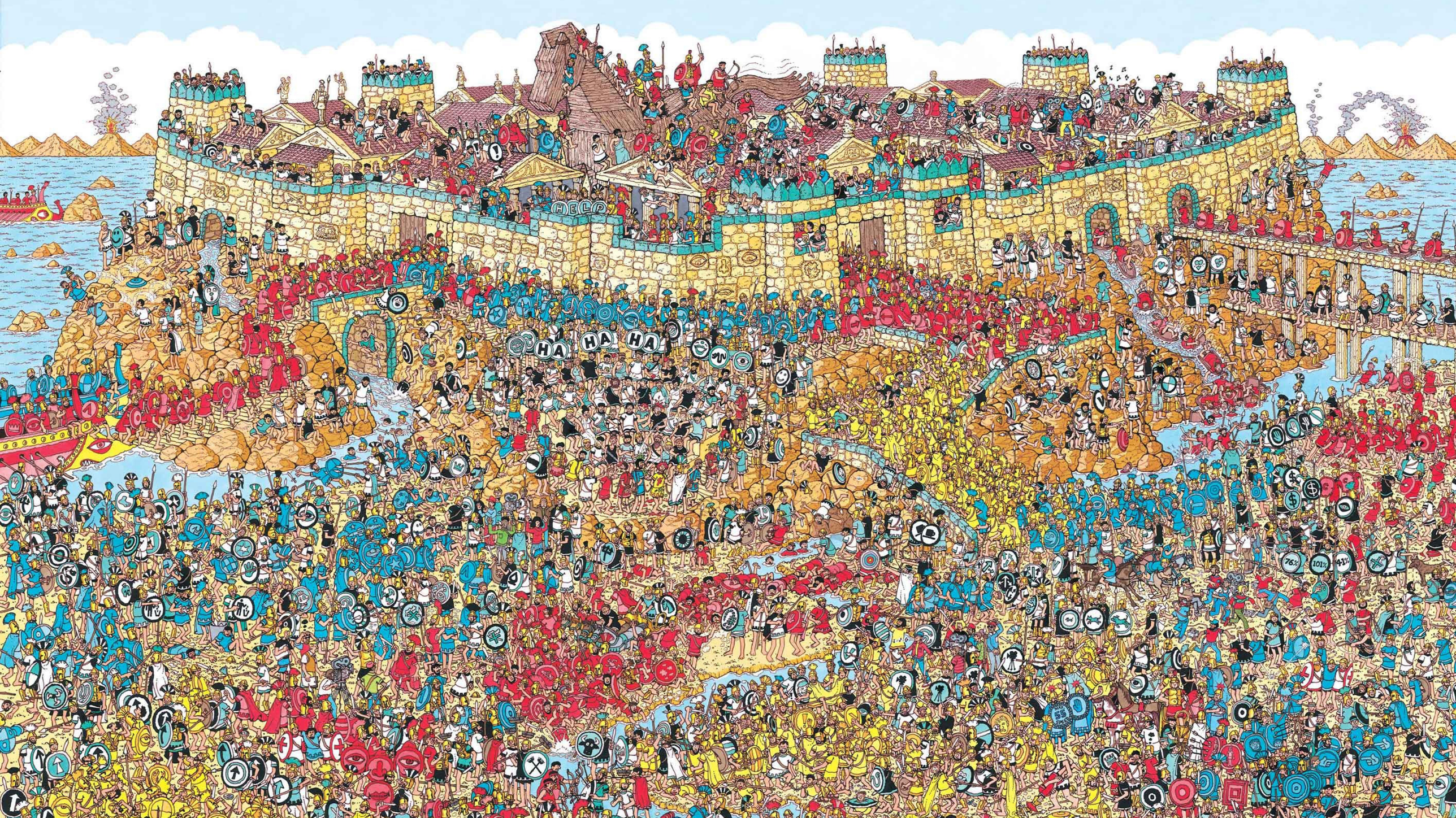


U2F authentication

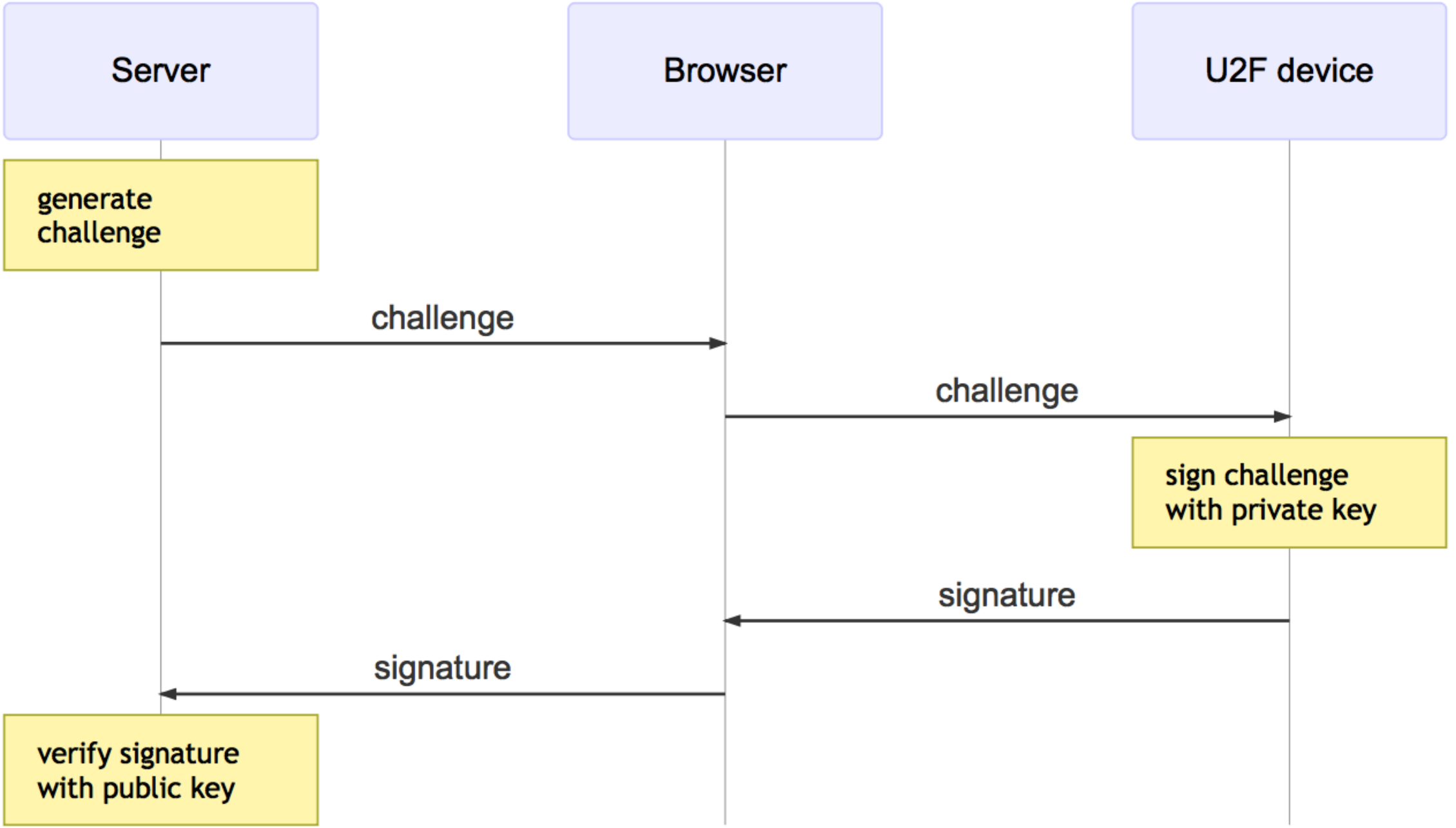


U2F authentication

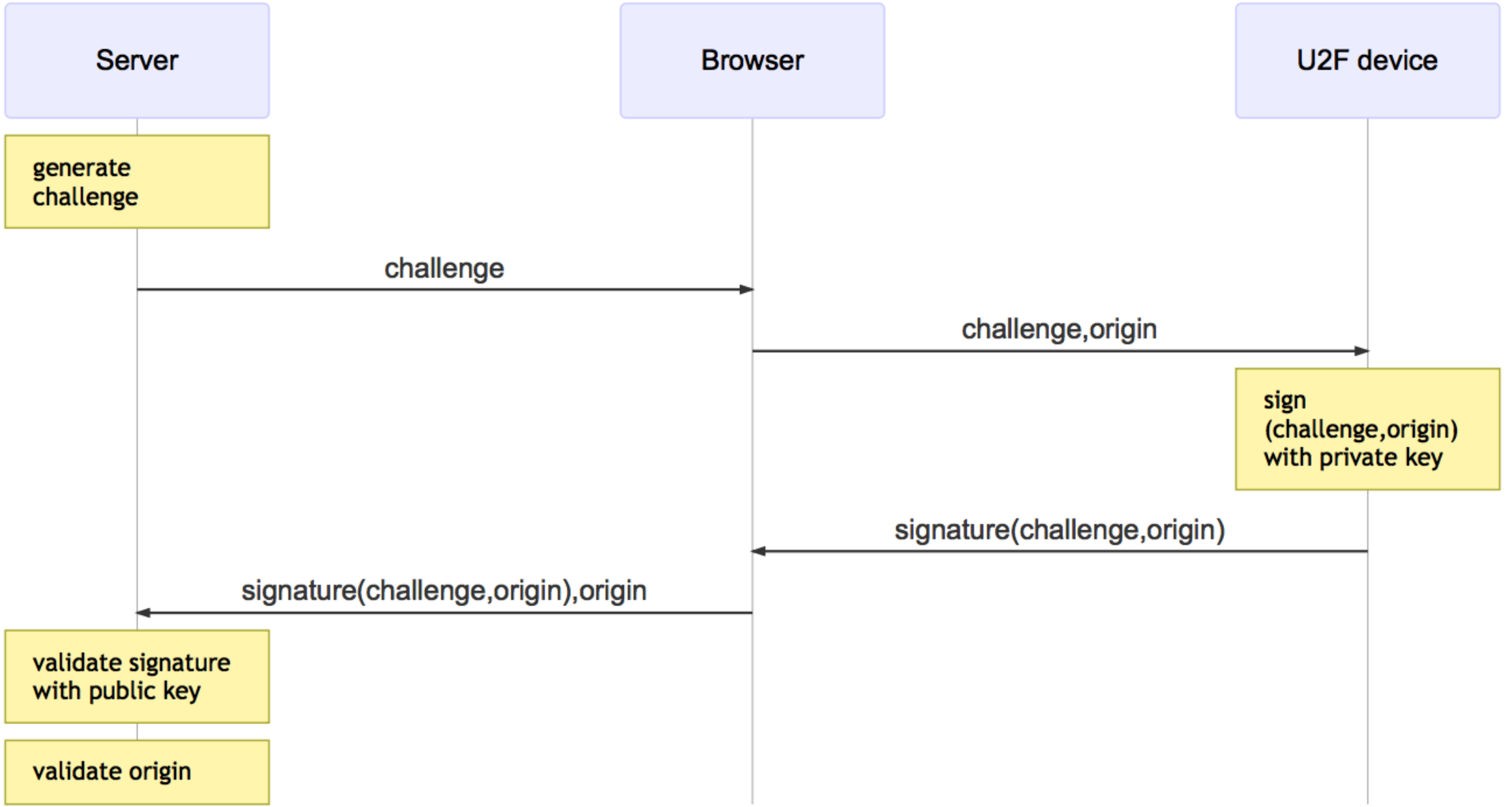



HELP
HA HA HA

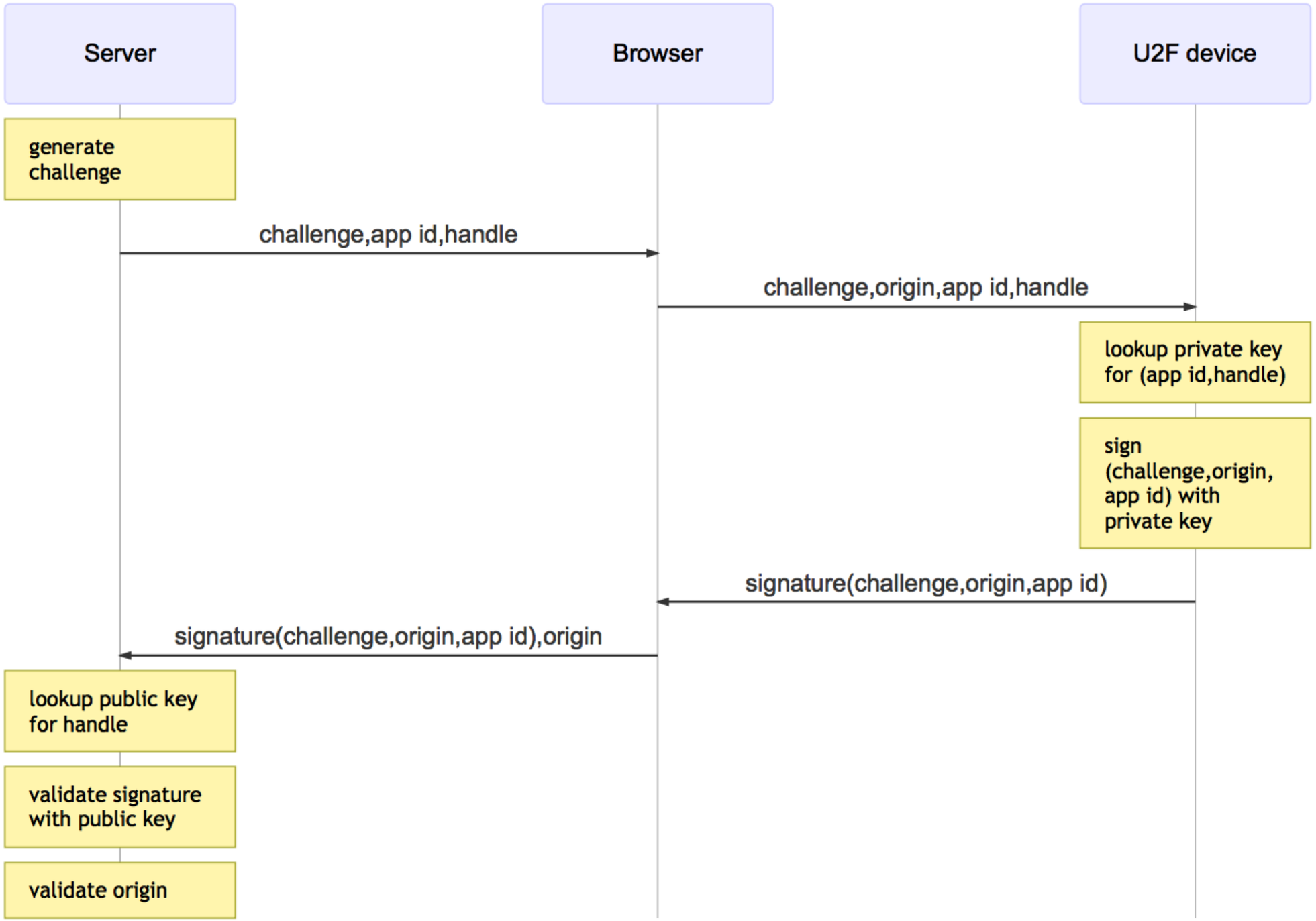

Authentication
flow


Server
Browser
U2F device
generate challenge
challenge
challenge
sign challenge with private key
signature
signature
verify signature with public key

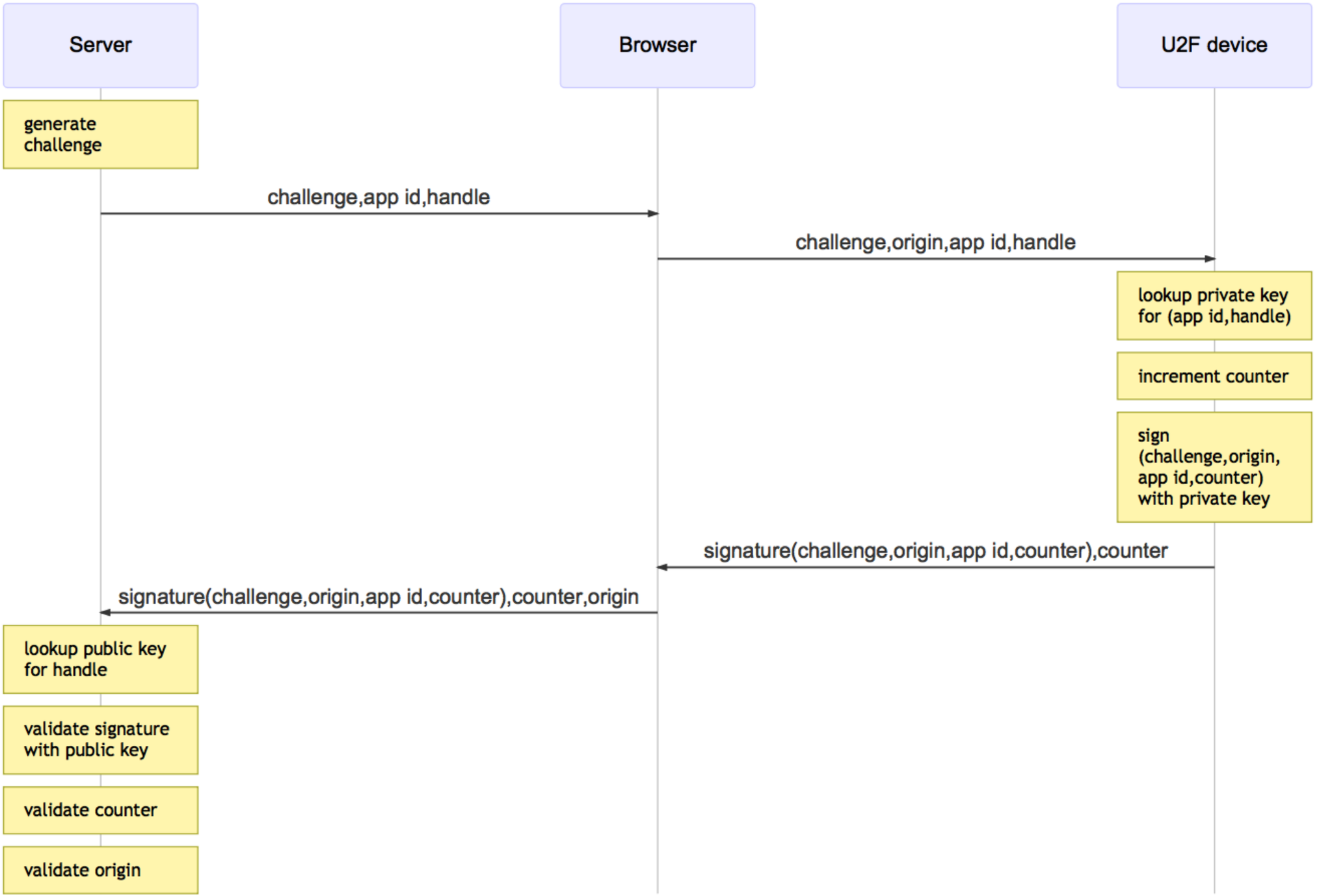

Phishing
protection


Server
Browser
U2F device
generate challenge
challenge
challenge,origin
sign (challenge,origin) with private key
signature(challenge,origin)
signature(challenge,origin),origin
validate signature with public key
validate origin


Application-specific keys


Server
Browser
U2F device
generate challenge
challenge,app id,handle
challenge,origin,app id,handle
lookup private key for (app id,handle)
sign (challenge,origin, app id) with private key
signature(challenge,origin,app id)
signature(challenge,origin,app id),origin
lookup public key for handle
validate signature with public key
validate origin


Cloning
protection


Server
Browser
U2F device
generate challenge
challenge,app id,handle
challenge,origin,app id,handle
lookup private key for (app id,handle)
increment counter
sign (challenge,origin, app id,counter) with private key
signature(challenge,origin,app id,counter),counter
signature(challenge,origin,app id,counter),counter,origin
lookup public key for handle
validate signature with public key
validate counter
validate origin


Registration
flow


Server
Browser
U2F device
generate challenge
challenge,app id
challenge,origin,app id
generate key pair and handle
store (app id,handle) -> private key
sign (challenge,origin, app id,handle, public key) with attestation private key
signature(challenge,origin, app id,handle, public key),handle,public key,attestation cert
signature(challenge,origin, app id,handle, public key),handle,public key,attestation cert,origin
validate signature with attestation public key
validate origin
store handle -> public key

Attestation certificate

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 776137165 (0x2e42e9cd)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=Yubico U2F Root CA Serial 457200631
        Validity
            Not Before: Aug  1 00:00:00 2014 GMT
            Not After : Sep  4 00:00:00 2050 GMT
        Subject: CN=Yubico U2F EE Serial 776137165
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
            EC Public Key:
                pub:
                    04:49:ba:3d:d4:9c:3b:a1:5b:d5:b8:75:8d:ef:db:
                    49:2e:2a:8c:3e:3f:70:02:c4:4d:5d:d4:83:3f:9f:
                    c0:ce:40:9d:91:37:4a:f0:51:7a:f2:00:6a:ba:39:
                    c2:fb:73:1b:36:71:a0:ce:5c:e9:da:c1:84:b5:61:
                    95:b9:70:cd:4c
                ASN1 OID: prime256v1
```

Implementation details

Serverland

- Perl, Ruby, Python, PHP, Java, C#, C, Go, Javascript...
- Provide three main functions
  - Generate a challenge
  - Verify a registration response
  - Verify an authentication response

Browserland

- Javascript
- Browser support
  - Chromium-based browsers (Chrome, Opera) via `u2f-api.js`
  - Firefox via extension (native Real Soon Now™)
  - ...
- Mobile
  - Android + Chrome + Google Authenticator + NFC

Browserland

```
u2f.register(location.origin, // appId
             [{ challenge: '...', version: 'U2F_V2' }],
             [],
             function (r) {
               if (r.errorCode) {
                 alert("something bad happened: "+r.errorCode);
                 return;
               }
               // send r.registrationData and r.clientData
               // to server via XHR, form POST, etc
             }
            );
```

Browserland

```
u2f.sign(location.origin, // appId
         [{ challenge: '...', version: 'U2F_V2' }],
         [{ keyHandle: '...', version: 'U2F_V2' }],
         function (r) {
           if (r.errorCode) {
             alert("something bad happened: "+r.errorCode);
             return;
           }
           // send r.keyHandle, r.signatureData and r.clientData
           // to server via XHR, form POST, etc
         }
        );
```


Browserless

libu2f-host

pam-u2f

Bonus prize round!

U 2 can U2F!

U 2 can U2F!

- U2F is an open standard
- Secure
- Easy to use
- Lots of hardware to choose from, or build your own
- Simple to implement


Questions?