FIDO, ID Proofing and Federation

Jerrod Chong, VP Solutions
FIDO U2F AKA Security Keys

Global open authentication standard co-created by Yubico & Google

One Authenticator  Any # of Services  No Shared Secrets
Google Security Key Login

1. Sign in to continue to Gmail
2. Welcome
3. 2-Step Verification: Insert your Security Key into your computer's USB port. If it has a button, tap it.

- **Secure**: Unphishable / UnMITMable
- **Simple**: Insert and press button
- **Scalable**: One device, many services
- **Privacy**: No Link-ability between services
1st Government to offer citizens opt-in U2F Secured Digital ID
Why are we solving this?

- Strong authentication not always tied to identity of user
- FIDO authentication mostly decoupled from ID Proofing
- ID Proofing required for higher assurance levels
- ID Proofing and strong authentication at odds with privacy
- Remote ID Proofing tied to Knowledge Based Verification (KBV)
- Reduce the reliance on weak recovery options
“Individuals and organizations utilize secure, efficient, easy-to-use and interoperable identity solutions to access online services in a manner that promotes confidence, privacy, choice, and innovation.”

-NIST Trusted Identity Group (TIG)
NIST SP 800-63-3

Digital Identity Guidelines 800-63 Revision 3

Highlighting 3 Policy Recommendations

- Decouple Identity assurance from Authenticator assurance
- Deprecate the use of SMS as Out-of-Band verifier
- Approves FIDO U2F for use at highest Authenticator assurance level (AAL3)
The Project

Yubico awarded US NIST grant collaborating with various Identity Providers

● Extend benefits of FIDO U2F to federated identity environments
● Integrate ID Proofing with FIDO U2F authentication
● Share attributes securely, conveniently and privacy-enhancing
ID verified FIDO Authenticators

- Successful Remote Proofing issues Pre-registered authenticator
- Pre-registration of authenticator ensures authenticity and integrity (first FIDO credential must be ID verified)

Mobile ID scanning, Driver’s license or state ID  ➔  U2F Authenticator sent to the address on ID  ➔  Secure access to any number of services
ID Proofing and Verification (IPV)
Remote ID Proofing Mobile App

Is the address on your driver license current?

Driver License
Using the camera on your phone, scan the front and back of your driver license in any order. A checkmark appears when the scan is complete for each side. Click OK to continue.

[Image of mobile app screens showing the process of remote ID proofing, including options for front and back of license, and prompts to confirm address on driver license.]

© 2017 Yubico
Token Issuance with Pre-Registration

Launch Logistics Flow

Yubico
Provision and send pre-registered U2F Token to captured address

Invalid code/not used name/address discarded automatically in 14 days

Core IPV user store

U2F Token Issuance Logistics Flow for Remote Proofing

Discard instruction

Dashboard fetch U2F binding data from core IPV and stores the U2F data in the IdP

Valid code

Login to dashboard with activation code to complete identity proofing and binding

+ Activation code

+ Activation code

Discard name/address from core IPV

Yubico
Provision and send pre-registered U2F Token to captured address

IdP

U2F binding data

© 2017 Yubico
Pre-Registration of Key Handle

U2F Device

Generate:
- \( k_{pub} \)
- \( k_{priv} \)
- handle \( h \)

Client

Check app id

- a; challenge, origin, channel id, etc.

- \( c \)

- \( k_{pub}, h, \) attestation cert, signature(\( a, c, k_{pub}, h \))

- \( s \)

Relying Party

- app id, challenge

- \( a \)

- \( c, k_{pub}, h, \) attestation cert, s

Release \( k_{pub} \) with handle \( h \) for user
Authentication Flow

USER

Request access to Service Provider (SP) Website

Redirect user to home institution Identity Provider (IdP)

Prompt user to login + Send attributes to SP

Deliver content to user

Purge user attribute per IdP-SP contract

Attribute Assertion

IdP data store
Authentication

U2F Device

- Lookup the $k_{priv}$ associated with $h$
- Sign with $k_{priv}$
- counter++

Client

- $h, a; \text{challenge, origin, channel id, etc.}$
- counter, signature(a,c, counter)
- $s$

Relying Party

- handle, app id, challenge
- $h, a$
- counter, c, s
- $s$
- Check $s$ using $k_{pub}$
- Verify origin, channel id & counter

- Lookup the $k_{pub}$ associated with $h$
Identity Ecosystem using Open Standards

- Extend FIDO to services connected via these federation protocols
  - U2F Shibboleth (SAML) and OpenID Connect plug-in
  - Open source reference implementation

- Build ID Proofing engine using OpenID Connect
  - Allows for multiple proofing solutions/providers
  - Part of the Identity toolkit
Lessons Learned

- Protecting PII is time and resource intensive
- Difficult to achieve highest identity assurance with Remote ID proofing
- High level of trust required in integrations with third-party vendors
- Compatibility challenges across diverse operating systems and devices
- Additional techniques needed to onboard special needs individuals
Questions?