"Key Migration Protocol"

Sounds Scary

Singapore Plenary - Oct, 2018

"Authenticator Migration Protocol"

Sounds less scary
...and comes with neat acronym: "AMP"!

Singapore Plenary - Oct, 2018

FIDO Use Cases / User Journeys

- 1st factor authentication (user has no password)
- 2nd factor authentication (user has a password, but it's not enough to sign in)
- Reauth (user has used pasword before, but FIDO is faster on this device)

Problem Space

User loses authenticator (or it breaks or gets stolen)

Problem Space

User loses authenticator (or it breaks or gets stolen)

- needs to set up new authenticator with every RP
- potentially undergoes account recovery (with many RPs)
- potentially gets locked out (of several RPs)

Ideal Solution

User loses authenticator (or it breaks or gets stolen)

- needs to set up new authenticator ~~with every RP~~
- potentially undergoes account recovery ~~(with many RPs)~~
- ~~potentially gets locked out (of several RPs)~~

Progress so far (that I'm aware of)

- AR Solution Catalog
  - AR through Credential Binding
    - Password
    - Static Knowledge-based Authentication (Static KBA)
    - Proof-of-Possession of Device & Phone Numbers
    - Multiple Authenticators
    - Trusted Person's Authenticator
  - AR through Attribute Binding
    - Copies of "Official" Document(s)
    - Email
    - SMS
    - Physical Mail
    - Passive Voice
    - Dynamic KBA
    - Activity and Behavior
  - AR through Assertion Binding
    - Passport or Advanced Government ID Cards
    - Identity Federation
    - Collaborative Recovery
    - Trusted Person's Authorization

FIDO Whitepaper

- Thorough overview of Account Recovery problem and solution space
- Related problem, but not exactly the same.
- Not yet public

University of Washington proposal (in collaboration with several FIDO members)

- Sketch of several possible solutions

Public mini-Workshop at the University of Washington

- Discussed acceptable solutions
- Outlined requirements for solution

FIDO Use Cases / User Journeys

- 1st factor authentication (user has no password)
- 2nd factor authentication (user has a password, but it's not enough to sign in)
- **Reauth (user has used pasword before, but FIDO is faster on this device)**

Possible User Experience(s)

Restoring Platform authenticator

1. [at some point] Platform performs an "authenticator backup"


Private Keys
can't leave
Authenticators

TLDR; we don't need to "have keys leave authenticators" to get the job done

Also "having keys not leave authenticators" is one way of protecting against certain attacks -- it's not the only way. It's an implementation detail.

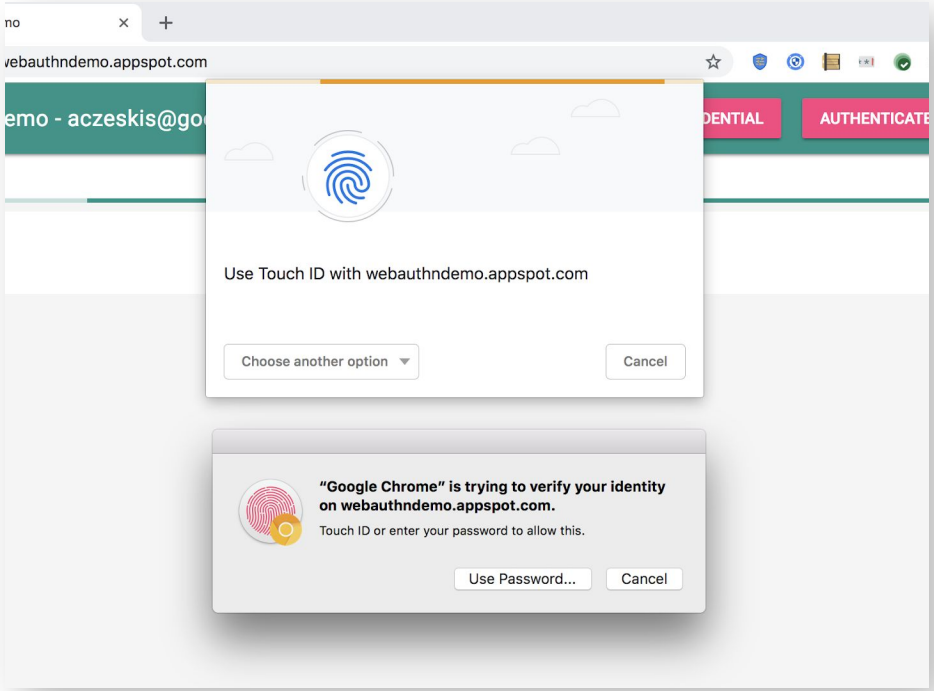
Possible User Experience(s)

Restoring Platform authenticator

1. [at some point] Platform performs an "authenticator backup" to the cloud
2. User loses access to platform (e.g., laptop gets stolen)
3. User gets new platform and authenticates to it
4. User visits all their favorite RPs, who learn that the user has a new device and that the platform has already verified the user

One way this might this work (using Chrome)

Chrome M70 provides a touch-id based authenticator on OS X



One way this could this work (using Chrome)

Chrome M70 (currently beta) provides a touch-id based authenticator on OS X

- Keys get stored on device

Attestation *could* be certificate chain of

- Root could be Google (sync)
- Intermediate could be unique for {user, RP}
- Leaf could be stored on device (in keychain) as before

When user gets new device

- Intermediate cert will be the same (pulled from cloud)
- Leaf will be different

RP will learn that the user is on a new device and that Google has verified them. RP can accept login or do more verifications.

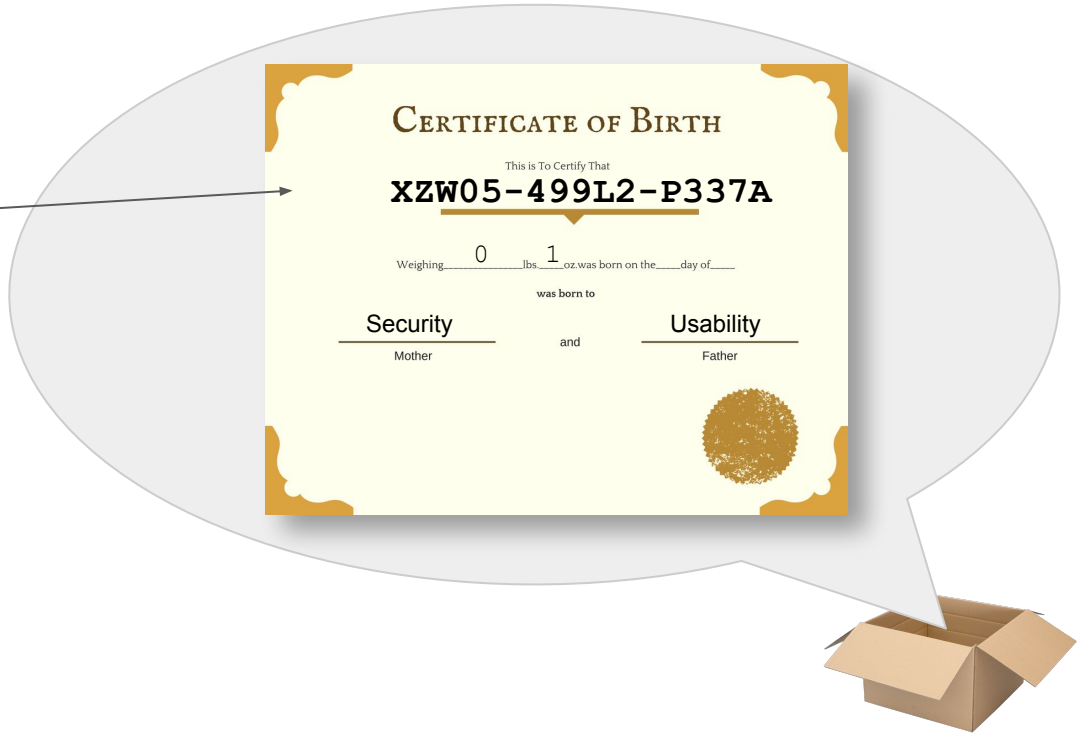
FIDO Use Cases / User Journeys

- **1st factor authentication (user has no password)**
- **2nd factor authentication (user has a password, but it's not enough to sign in)**
- Reauth (user has used pasword before, but FIDO is faster on this device)

Another way this could work

Your authenticator comes with a birth certificate:

Root Key = PIN + BirthKey



When losing an authenticator, user can bootstrap new authenticators and have all keys "back".

Many others approaches possible